퍼블릭 클라우드에 Threat Defense Virtual용 클러스터 구축

초판: 2022년 5월 27일

최종 변경: 2023년 12월 13일

퍼블릭 클라우드에서 Threat Defense Virtual용 클러스터 구축

클러스터링을 사용하면 여러 개의 Threat Defense Virtual를 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. 다음과 같은 퍼블릭 클라우드 플랫폼을 사용하여 퍼블릭 클라우드에서 Threat Defense Virtual 클러스터를 구축할 수 있습니다.

- AWS(Amazon Web Services)
- Microsoft Azure
- GCP(Google Cloud Platform)

현재는 라우팅 방화벽 모드만 지원됩니다.

참고 클러스터링을 사용할 경우 일부 기능이 지원되지 않습니다. 지원되지 않는 기능 및 클러스터링, 77 페이지의 내용을 참조하십시오.

퍼블릭 클라우드의 Threat Defense Virtual 클러스터링 정보

이 섹션에서는 클러스터링 아키텍처 및 이러한 아키텍처의 작동 방식에 대해 설명합니다.

클러스터를 네트워크에 맞게 활용하는 방법

클러스터는 하나의 디바이스로 작동하는 여러 개의 방화벽으로 구성됩니다. 클러스터로 작동하려면 방화벽에는 다음과 같은 인프라가 필요합니다.

- VXLAN 인터페이스를 사용하는 클러스터 내 통신을 위한 격리된 네트워크(클러스터 제어 링크라고 함). 레이어 3 물리적 네트워크를 통해 레이어 2 가상 네트워크 역할을 하는 VXLAN은 클러스터 제어 링크를 통해 Threat Defense Virtual에서 브로드캐스트/멀티캐스트 메시지를 전송하도록 합니다.
- Load Balancer(로드 밸런서) - 외부 로드 밸런싱의 경우 퍼블릭 클라우드에 따라 다음과 같은 옵션이 있습니다.

- AWS 게이트웨이 로드 밸런서

  AWS 게이트웨이 로드 밸런서는 트래픽을 분산하고 온디맨드 방식으로 가상 어플라이언스를 확장하는 로드 밸런서와 투명 네트워크 게이트웨이를 결합합니다. Threat Defense Virtual은 Geneve 인터페이스 단일 암 프록시를 사용하여 분산형 데이터 플레인(게이트웨이 로드 밸런서 엔드포인트)이 있는 게이트웨이 로드 밸런서 중앙 집중식 제어 평면을 지원합니다.

- Azure 게이트웨이 로드 밸런서

  Azure 서비스 체인에서 Threat Defense Virtual은 인터넷과 고객 서비스 간의 패킷을 인터셉트할 수 있는 투명 게이트웨이 역할을 합니다. Threat Defense Virtual는 페어링된 프록시에서 VXLAN 세그먼트를 활용하여 단일 NIC에서 외부 인터페이스 및 내부 인터페이스를 정의합니다.

- 기본 GCP 로드 밸런서, 내부 및 외부

- Cisco Cloud Services Router와 같은 내부 및 외부 라우터를 사용하는 ECMP(Equal-Cost Multi-Path Routing)

  ECMP 라우팅을 사용하면 라우팅 메트릭에서 가장 순위가 높은 여러 가지 "최상의 경로"를 통해 패킷을 전달할 수 있습니다. EtherChannel과 마찬가지로, 소스와 목적지 IP 주소 및/또는 소스와 목적지 포트의 해시를 사용하여 다음 홉 중 하나로 패킷을 보낼 수 있습니다. ECMP 라우팅을 위한 고정 경로를 사용할 경우, Threat Defense 오류가 발생하면 문제를 초래할 수 있습니다. 경로는 계속 사용할 수 있으며 오류가 발생한 Threat Defense에 대한 트래픽은 손실됩니다. 고정 경로를 사용할 경우 Object Tracking 같은 고정 경로 모니터링 기능을 사용할 수 있는지 확인하십시오. 동적 라우팅 프로토콜을 사용하여 경로를 추가 및 제거하는 것이 좋으며, 이 경우 동적 라우팅에 참여하도록 각 Threat Defense를 구성해야 합니다.

참고 레이어 2 스팬 EtherChannel은 로드 밸런싱에 지원되지 않습니다.

개별 인터페이스

클러스터 인터페이스를 개별 인터페이스로 구성할 수 있습니다.

개별 인터페이스는 정상적인 라우팅 인터페이스로, 각각 로컬 IP 주소가 있습니다. 인터페이스 구성은 제어 노드에서만 구성해야 하며 각 인터페이스는 DHCP를 사용합니다.

참고 레이어 2 Spanned EtherChannel은 지원되지 않습니다.

제어 및 데이터 노드 역할

클러스터의 멤버 중 하나는 제어 노드입니다. 여러 클러스터 노드가 동시에 온라인 상태가 되면 의 우선 순위 설정에 따라 제어 노드가 결정됩니다. 우선순위는 1에서 100까지 1이 가장 높은 우선순위

입니다. 다른 모든 멤버는 데이터 노드입니다. 클러스터를 처음 생성할 때 제어 노드가 될 노드를 지정하면 클러스터에 추가된 첫 번째 노드이기 때문에 제어 노드가 됩니다.

클러스터의 모든 노드에서는 동일한 구성을 공유합니다. 처음에 제어 노드로 지정하는 노드는 클러스터에 참가할 때 데이터 노드의 구성을 덮어쓰므로 클러스터를 구성하기 전에 제어 노드에서 초기 구성만 수행하면 됩니다.

일부 기능은 클러스터로 확장되지 않으며, 제어 노드에서 이러한 기능에 대한 모든 트래픽을 처리합니다.

클러스터 제어 링크

각 노드는 클러스터 제어 링크에 대한 하나의 인터페이스를 VTEP(VXLAN) 전용 인터페이스로 사용해야 합니다.

VXLAN 터널 엔드포인트

VXLAN 터널 엔드포인트(VTEP) 디바이스는 VXLAN 캡슐화 및 역캡슐화를 수행합니다. 각 VTEP에는 2개의 인터페이스 유형이 있습니다. VNI(VXLAN 네트워크 식별자) 인터페이스라고 하는 하나 이상의 가상 인터페이스에는 VTEP 소스 인터페이스라고 하는 일반 인터페이스는 VTEP 사이에서 VNI 인터페이스를 터널링합니다. VTEP 소스 인터페이스는 VTEP대 VTEP 통신을 위해 전송 IP 네트워크에 연결됩니다.

VTEP 소스 인터페이스

VTEP 소스 인터페이스는 VNI 인터페이스를 연결하려는 Threat Defense Virtual 일반 인터페이스입니다. 클러스터 제어 링크 역할을 하도록 하나의 VTEP 소스 인터페이스를 구성할 수 있습니다. 소스 인터페이스는 클러스터 제어 링크용으로만 예약되어 있습니다. 각 VTEP 소스 인터페이스는 동일한 서브넷에 IP 주소가 있습니다. 이 서브넷은 모든 다른 트래픽과 분리되어 있어야 하며, 클러스터 제어 링크 인터페이스만 포함해야 합니다.

VNI 인터페이스

VNI 인터페이스는 VLAN 인터페이스와 유사합니다. 이 인터페이스는 태그 지정을 사용하여 지정된 물리적 인터페이스에서 네트워크 트래픽을 분리하여 유지하는 가상 인터페이스입니다. 하나의 VNI 인터페이스만 구성할 수 있습니다. 각 VNI 인터페이스는 동일한 서브넷에 IP 주소가 있습니다.

피어 **VTEP**

단일 VTEP 피어를 허용하는 데이터 인터페이스용 일반 VXLAN과 달리 Threat Defense Virtual 클러스터링에서는 여러 피어를 구성할 수 있습니다.

클러스터 제어 링크 트래픽 개요

클러스터 제어 링크 트래픽에는 제어 및 데이터 트래픽이 모두 포함됩니다.

제어 트래픽에는 다음 사항이 해당됩니다.

- 제어 노드 선택.

- 구성 복제
- 상태 모니터링

데이터 트래픽에는 다음 사항이 해당됩니다.

- 상태 복제
- 연결 소유권 쿼리 및 데이터 패킷 전송

구성 복제

클러스터의 모든 노드에서는 단일 구성을 공유합니다. 제어 노드에서는 구성만 변경할 수 있으며(부트스트랩 구성 예외), 변경 사항은 클러스터의 모든 다른 노드에 자동으로 동기화됩니다.

관리 네트워크

관리 인터페이스를 사용하여 각 노드를 관리해야 합니다. 데이터 인터페이스에서의 관리는 클러스터링에서 지원되지 않습니다.

Threat Defense Virtual 클러스터링용 라이선스

각 threat defense virtual 클러스터 노드에는 동일한 성능 계층 라이선스가 필요합니다. 모든 멤버에 대해 동일한 수의 CPU 및 메모리를 사용하는 것이 좋습니다. 그렇지 않으면 성능이 가장 낮은 멤버와 일치하도록 모든 노드에서 제한됩니다. 처리량 레벨은 제어 노드에서 각 데이터 노드로 복제되어 일치합니다.

개별 노드가 아니라 전체 피처 클러스터에 라이선스를 할당합니다. 그러나 클러스터의 각 노드는 각 기능에 대한 별도 라이선스를 사용합니다. 클러스터링 기능 자체에는 라이선스가 필요하지 않습니다.

Management Center에 제어 노드를 추가하는 경우 클러스터에 사용하려는 기능 라이선스를 지정할 수 있습니다. **Devices(**디바이스**)** > **Device Management(**디바이스 관리**)** > **Cluster(**클러스터**)** > **License(**라이선스**)** 영역에서 클러스터에 대한 라이선스를 수정할 수 있습니다.

참고 Management Center가 라이선스 되기 전에 (평가 모드에서 실행 되기 전에) 클러스터를 추가하는 경우, Management Center를 라이선스하면 클러스터에 정책 변경을 구축할 때 트래픽 중단이 발생할 수 있습니다. 라이선스 모드를 변경하면 모든 데이터 유닛이 클러스터를 벗어났다가 다시 참가합니다.

Threat Defense Virtual 클러스터링의 요구 사항 및 사전 요건

모델 요구 사항

- FTDv5, FTDv10, FTDv20, FTDv30, FTDv50, FTDv100

참고 FTDv5 및 FTDv10은 AWS(Amazon Web Services) 게이트웨이 로드 밸런서(GWLB) 및 Azure GWLB를 지원하지 않습니다.

- 다음 퍼블릭 클라우드 서비스:
  - AWS(Amazon Web Services)
  - Microsoft Azure
  - GCP(Google Cloud Platform)
- 최대 16개 노드

Cisco Secure Firewall Threat Defense Virtual 시작 가이드의 Threat Defense Virtual에 대한 일반 요구 사항도 참조하십시오.

사용자 역할

- 관리자
- 액세스 관리자
- 네트워크 관리자

하드웨어 및 소프트웨어 요건

클러스터의 모든 유닛은 다음과 같아야 합니다.

- 동일한 성능 계층에 있어야 합니다. 모든 노드에 대해 동일한 수의 CPU 및 메모리를 사용하는 것이 좋습니다. 그렇지 않으면 성능이 가장 낮은 노드와 일치하도록 모든 노드에서 제한됩니다.
- Management Center 액세스는 관리 인터페이스에서 이루어져야 합니다. 데이터 인터페이스 관리는 지원되지 않습니다.
- 이미지 업그레이드 시간을 제외하고는 동일한 소프트웨어를 실행해야 합니다. 무중단 업그레이드가 지원됩니다.
- 클러스터의 모든 유닛은 동일한 가용성 영역에 구축되어야 합니다.
- 모든 유닛의 클러스터 제어 링크 인터페이스는 동일한 서브넷에 있어야 합니다.

MTU

클러스터 제어 링크에 연결된 포트에 올바른(더 높은) MTU가 구성되어 있는지 확인합니다. MTU가 일치하지 않으면 클러스터 형성이 실패합니다. 클러스터 제어 링크 MTU는 데이터 인터페이스보다 154바이트 더 커야 합니다. 클러스터 제어 링크 트래픽에는 데이터 패킷 전달이 포함되므로 클러스터 제어 링크는 데이터 패킷의 전체 크기와 클러스터 트래픽 오버헤드(100바이트) 및 VXLAN 오버헤드(54바이트)를 모두 수용해야 합니다.

GWLB를 사용하는 AWS의 경우 데이터 인터페이스는 Geneve 캡슐화를 사용합니다. 이 경우 전체 이더넷 데이터그램이 캡슐화되고 있으므로 새 패킷이 더 크고 더 대량의 MTU가 필요합니다. 소스 인터페이스 MTU를 네트워크 MTU + 306바이트로 설정해야 합니다. 따라서 표준 1500 MTU 네트워크 경로의 경우 소스 인터페이스 MTU는 1806이어야 하며 클러스터 제어 링크 MTU는 +154, 1960이어야 합니다.

GWLB를 사용하는 Azure의 경우 데이터 인터페이스는 VXLAN 캡슐화를 사용합니다. 이 경우 전체 이더넷 데이터그램이 캡슐화되고 있으므로 새 패킷이 더 크고 더 대량의 MTU가 필요합니다. 클러스터 제어 링크 MTU를 소스 인터페이스 MTU + 80바이트로 설정해야 합니다.

다음 표에는 클러스터 제어 링크 MTU 및 데이터 인터페이스 MTU의 기본값이 나와 있습니다.

표 ***1:*** 기본 ***MTU***

퍼블릭 클라우드	클러스터 제어 링크 **MTU**	데이터 인터페이스 **MTU**
GWLB를 사용하는 AWS	1960	1806
AWS	1654	1500
GWLB를 사용하는 Azure	1554	1454
Azure	1554	$1400
GCP	1554	$1400

표 ***2:*** 기본 ***MTU(***버전 ***7.4.x*** 이상***)***

퍼블릭 클라우드	클러스터 제어 링크 **MTU**	데이터 인터페이스 **MTU**
GWLB를 사용하는 AWS	1980	1826
AWS	1654	1500
GWLB를 사용하는 Azure	1454	1374
Azure	1454	1300
GCP	1554	$1400

Threat Defense Virtual 클러스터링에 대한 지침

고가용성

고가용성은 클러스터링에서 지원되지 않습니다.

IPv6

클러스터 제어 링크는 IPv4를 사용하는 경우에만 지원됩니다.

추가 지침

- 중요한 토폴로지 변경 사항(예: EtherChannel 인터페이스 추가 또는 제거, Threat Defense 또는 스위치의 인터페이스 활성화 또는 비활성화, 추가 스위치를 추가하여 VSS 또는 vPC 구성)이 발생할 경우 상태 검사 기능을 비활성화하고 비활성화된 인터페이스에 대해 인터페이스 모니터링도 비활성화해야 합니다. 토폴로지 변경이 완료되고 구성 변경 사항이 모든 유닛과 동기화되면 인터페이스 상태 검사 기능을 다시 활성화할 수 있습니다.
- 기존 클러스터에 노드를 추가하거나 노드를 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.
- 노드에서 클러스터링을 비활성화하기 전에 노드의 전원을 끄지 마십시오.
- 암호 해독된 TLS/SSL 연결의 경우, 암호 해독 상태가 동기화되지 않습니다. 연결 소유자 장애가 발생하는 경우, 암호 해독된 연결이 재설정됩니다. 새 노드에 대한 새 연결을 설정해야 합니다. 암호 해독되지 않은 연결(암호 해독 안 함 규칙과 일치하는 연결)은 영향을 받지 않으며, 올바르게 복제됩니다.
- 동적 확장은 지원되지 않습니다.
- Secure Firewall 버전 7.2 또는 7.3을 사용하는 경우에는 AWS에 클러스터를 구축할 때 스테이트풀 대상 페일오버가 지원되지 않습니다.
- 각 유지 보수 기간이 완료된 후 전역 구축을 수행합니다.
- Auto Scale 그룹(AWS)/인스턴스 그룹(GCP)/확장 세트(Azure)에서 한 번에 둘 이상의 디바이스를 제거하지 않아야 합니다. 또한 Auto Scale 그룹(AWS)/인스턴스 그룹(GCP)/확장 세트(Azure)에서 디바이스를 제거하기 전에 디바이스에서 **cluster disable** 명령을 실행하는 것이 좋습니다.
- 클러스터의 데이터 노드 및 제어 노드를 비활성화하려는 경우에는 제어 노드를 비활성화하기 전에 데이터 노드를 비활성화하는 것이 좋습니다. 클러스터에 다른 데이터 노드가 있는 동안 제어 노드가 비활성화되는 경우, 데이터 노드 중 하나를 제어 노드로 승격해야 합니다. 역할 변경으로 인해 클러스터가 중단될 수 있습니다.
- 이 가이드에서 제공하는 사용자 지정 Day 0 컨피그레이션 스크립트를 사용하여 요구 사항에 따라 IP 주소를 변경하고, 사용자 지정 인터페이스 이름을 제공하고, CCL-Link 인터페이스의 시퀀스를 변경할 수 있습니다.
- 클라우드 플랫폼에서 Threat Defense Virtual 클러스터를 구축한 후 간헐적 ping 실패와 같은 CCL 불안정 문제가 발생하는 경우 CCL 불안정을 유발하는 원인을 해결하는 것이 좋습니다. 또한 CCL 불안정 문제를 어느 정도 완화하기 위한 임시 해결 방법으로 보류 시간을 늘릴 수 있습니다. 보류 시간을 변경하는 방법에 대한 자세한 내용은 클러스터 상태 모니터 설정 편집을 참조하십시오.
- Management Center Virtual에 대한 보안 방화벽 규칙 또는 보안 그룹을 구성할 때는 소스 IP 어드레스 레인지에 Threat Defense Virtual의 사설 및 공용 IP 주소를 모두 포함해야 합니다. 또한 Threat Defense Virtual의 보안 그룹 또는 보안 방화벽 규칙에서 Management Center Virtual의 사설 및 공용 IP 주소를 지정해야 합니다. 이는 클러스터링 구축 중에 노드를 올바르게 등록하기 위해 중요합니다.

클러스터링 기본값

- cLACP 시스템 ID가 자동 생성되며 시스템 우선순위는 기본적으로 1입니다.
- 클러스터 상태 검사 기능은 기본적으로 활성화되어 있으며 3초간의 대기 시간이 있습니다. 인터페이스 상태 모니터링은 모든 인터페이스에서 기본적으로 활성화됩니다.
- 장애가 발생한 클러스터 제어 링크에 대한 클러스터 자동 다시 참가 기능은 5분마다 무제한으로 시도됩니다.
- 장애가 발생한 데이터 인터페이스에 대한 클러스터 자동 다시 참가 기능은 간격이 2로 늘어 5분마다 3번 시도됩니다.
- 5초 연결 복제 지연은 HTTP 트래픽에 대해 기본적으로 활성화되어 있습니다.

AWS에서 클러스터 구축

AWS에서 클러스터를 구축하려면 수동으로 구축하거나 CloudFormation 템플릿을 사용하여 스택을 구축할 수 있습니다. AWS 게이트웨이 로드 밸런서 또는 Cisco Cloud Services Router와 같은 기본이 아닌 로드 밸런서와 함께 클러스터를 사용할 수 있습니다.

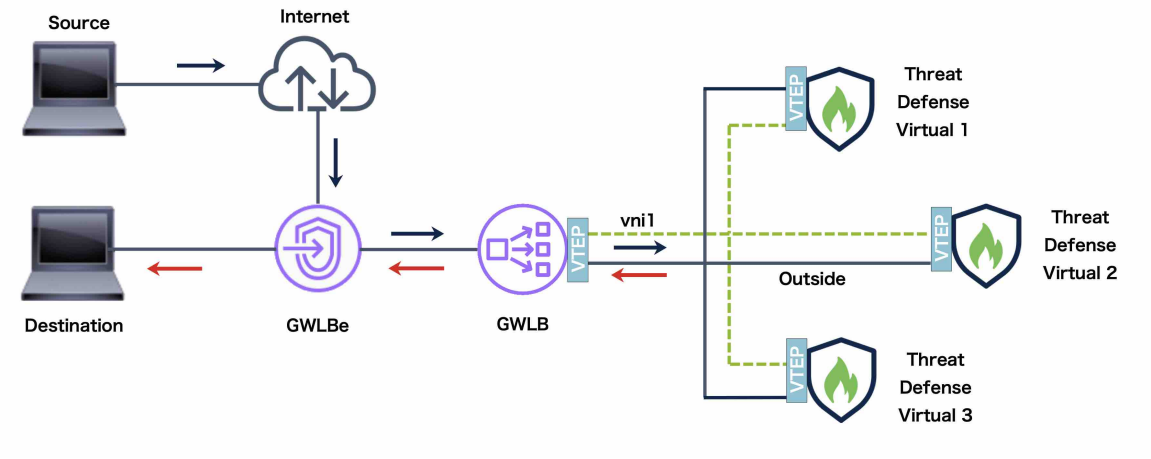
AWS 게이트웨이 로드 밸런서 및 **Geneve** 단일 암 프록시

참고 이 사용 사례는 Geneve 인터페이스에 대해 현재 지원되는 유일한 사용 사례입니다.

AWS 게이트웨이 로드 밸런서는 트래픽을 분산하고 온디맨드 방식으로 가상 어플라이언스를 확장하는 로드 밸런서와 투명 네트워크 게이트웨이를 결합합니다. Threat Defense Virtual은 분산형 데이터 플레인(게이트웨이 로드 밸런서 엔드포인트)이 있는 게이트웨이 로드 밸런서 중앙 집중식 제어 플레인을 지원합니다. 다음 그림에는 게이트웨이 로드 밸런서 엔드포인트에서 게이트웨이 로드 밸런서로 전달되는 트래픽이 나와 있습니다. 게이트웨이 로드 밸런서는 여러 Threat Defense Virtual 간에 트래픽을 밸런싱하며, 이를 삭제하거나 게이트웨이 로드 밸런서로 다시 전송하기 전에 트래픽을 검사합니다(U-turn 트래픽). 그런 다음 게이트웨이 로드 밸런서는 게이트웨이 로드 밸런서 엔드포인트 및 대상으로 트래픽을 다시 전송합니다.

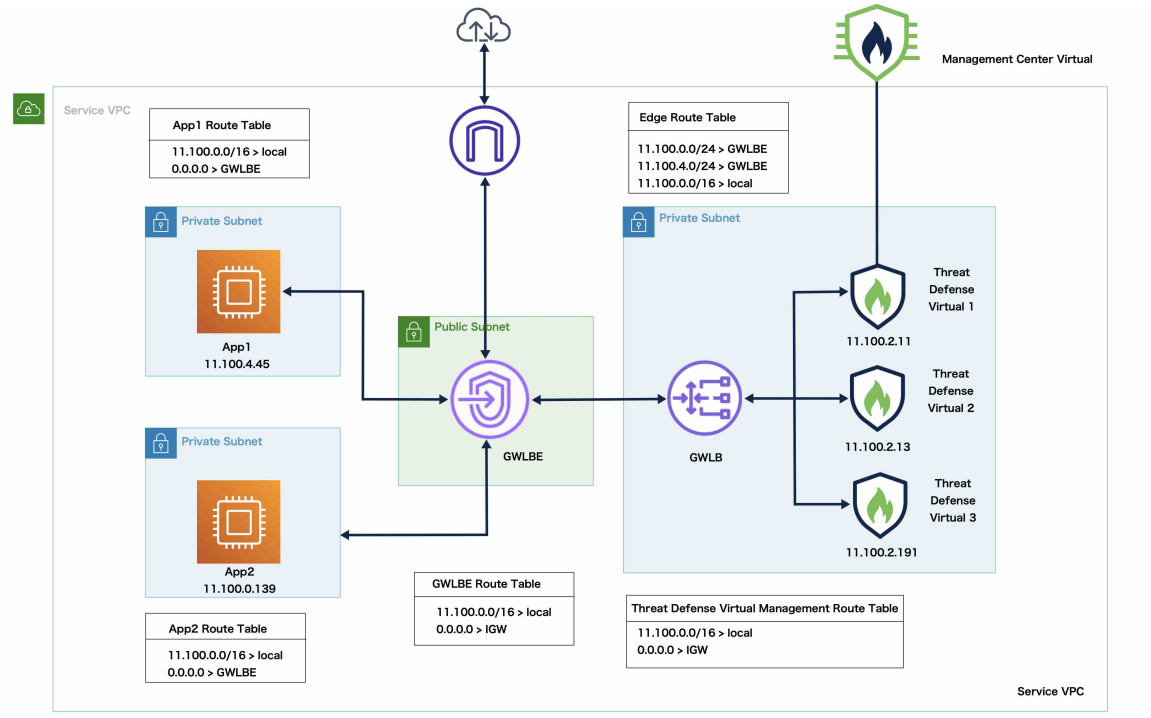
참고 TLS(Transport Layer Security) 서버 ID 검색은 AWS의 Geneve 단일 암 설정에서 지원되지 않습니다.

그림 ***1: Geneve*** 단일 암 프록시



샘플 토폴로지

아래 토폴로지에는 인바운드 및 아웃바운드 트래픽 플로우가 모두 나와 있습니다. GLLB에 연결된 클러스터에는 Threat Defense Virtual 인스턴스 3개가 있습니다. Management Center Virtual 인스턴스는 클러스터를 관리하는 데 사용됩니다.



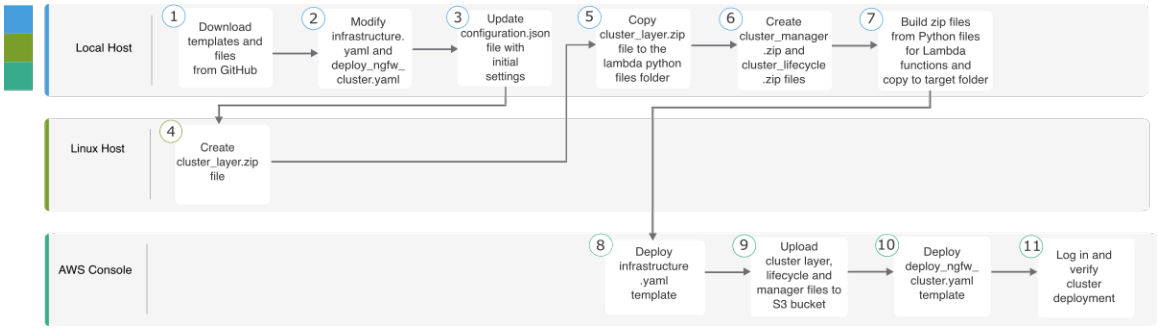
인터넷의 인바운드 트래픽은 GLLB 엔드포인트로 이동한 다음 이 엔드포인트에서 GWLB로 트래픽을 전송합니다. 그런 다음 트래픽은 Threat Defense Virtual 클러스터로 전달됩니다. 클러스터의 Threat Defense Virtual 인스턴스에서 검사된 트래픽은 애플리케이션 VM, App1/App2로 전달됩니다.

App1/App2의 아웃바운드 트래픽은 GWLB 엔드포인트로 전송된 다음 인터넷으로 전송됩니다.

AWP에서 **Threat Defense Virtual** 클러스터를 구축하기 위한 **End-to-End** 프로세스

템플릿 기반 구축

다음 순서도에는 AWS에서 Threat Defense Virtual 클러스터를 템플릿 기반으로 구축하는 워크플로우가 나와 있습니다.



	업무 환경	단계
1	로컬 호스트	GitHub에서 템플릿 및 파일을 다운로드합니다.
2	로컬 호스트	*infrastructure.yaml* 및 *deploy_ngfw_cluster.yaml* 템플릿을수정합니다.
3	로컬 호스트	초기 설정을 사용해 *Configuration.json* 파일을 업데이트합니다.
4	Linux 호스트	*cluster_layer.zip* 파일을 생성합니다.
5	로컬 호스트	*cluster_layer.zip* 파일을 Lambda python files 폴더에 복사합니다.
6	로컬 호스트	*cluster_manager.zip* 및 *cluster_lifecycle.zip* 파일을 생성합니다.
7	로컬 호스트	Lambda 함수용 Python 파일에서 zip 파일을 빌드하고 대상 폴더에 복사합니다.
8	AWS 콘솔	*infrastructure.yaml* 템플릿을 구축합니다.
9	AWS 콘솔	cluster_layer.zip, cluster_lifecycle.zip 및 cluster_manager.zip을 S3 버킷에 업로드합니다.
10	AWS 콘솔	*deploy_ngfw_cluster.yaml* 템플릿을 구축합니다.
11	AWS 콘솔	로그인하여 클러스터 구축을 확인합니다.

수동 구축

다음 순서도에는 AWS에서 Threat Defense Virtual 클러스터를 수동으로 구축하는 워크플로우가 나와 있습니다.

	업무 환경	단계
1	로컬 호스트	AWS에 대한 Day0 구성 생성
2	AWS 콘솔	Threat Defense Virtual 인스턴스를 구축합니다.
3	AWS 콘솔	인스턴스에 인터페이스를 연결합니다.
4	AWS 콘솔	노드가 클러스터에 조인되었는지 확인합니다.
5	AWS 콘솔	대상 그룹 및 GLLB를 생성합니다. TWLB에 대상 그룹을 연결합니다.
6	AWS 콘솔	데이터 인터페이스 IP를 사용하여 대상 그룹에 인스턴스를 등록합니다.
7	Management Center	제어 노드를 등록합니다.

템플릿

아래에 제공된 템플릿은 GitHub에서 사용할 수 있습니다. 매개변수 값은 템플릿에 제공된 매개변수 이름, 기본값, 허용된 값 및 설명을 통해 이해할 수 있습니다.

- infrastructure.yaml – 인프라 구축용 템플릿입니다.
- deploy_ngfw_cluster.yaml – 클러스터 구축용 템플릿입니다.

참고 클러스터 노드를 구축하기 전에 지원되는 AWS 인스턴스 유형 목록을 확인하십시오. 이 목록은 *deploy_ngfw_cluster.yaml* 템플릿의 InstanceType 매개변수에 허용되는 값 아래에서 찾을 수 있습니다.

CloudFormation 템플릿을 사용하여 AWS에서 스택 구축

사용자 지정된 CloudFormation 템플릿을 사용하여 AWS에 스택을 구축합니다.

시작하기 전에

- Python 3이 설치된 Linux 컴퓨터가 필요합니다.
- 클러스터가 management center에 자동 등록되도록 하려면 REST API를 사용할 수 있는 management center에 대한 관리 권한이 있는 사용자를 생성해야 합니다. Cisco Secure Firewall Management Center 관리 가이드를 참조하십시오.
- Configuration.JSON에서 지정한 정책의 이름과 일치하는 액세스 정책을 management center에 추가합니다.

프로시저

단계 **1** 템플릿을 준비합니다.

a) 로컬 폴더에 github 리포지토리를 복제합니다. https://github.com/CiscoDevNet/cisco-ftdv/tree/master/cluster/aws을 참조하십시오.

b) 필수 매개변수를 사용하여 **Infrastructure.yaml** 및 **deploy_ngfw_cluster.yaml**을 수정합니다.

c) 초기 설정으로 **cloud-clustering/ftdv-cluster/lambda-python-files/Configuration.json**을 수정합니다.

대표적인 예는 다음과 같습니다.

```
{
  "licenseCaps": ["BASE", "MALWARE", "THREAT"],
  "performanceTier": "FTDv50",
  "fmcIpforDeviceReg": "DONTRESOLVE",
  "RegistrationId": "cisco",
  "NatId": "cisco",
  "fmcAccessPolicyName": "AWS-ACL"
}
```

- fmcIpforDeviceReg 설정을 DONTRESOLVE로 유지합니다.
- fmcAccessPolicyName은 management center의 액세스 정책과 일치해야 합니다.

참고 FTDv5 및 FTDv10 계층은 지원되지 않습니다.

d) 람다 함수에 필수 Python 라이브러리를 제공하기 위해 **cluster_layer.zip**이라는 파일을 생성합니다.

Python 3.9가 설치된 Amazon Linux를 사용하여 **cluster_layer.zip** 파일을 생성하는 것이 좋습니다.

참고 Amazon Linux 환경이 필요한 경우 Amazon Linux 2023 AMI를 사용하여 EC2 인스턴스를 생성하거나 최신 버전의 Amazon Linux를 실행하는 AWS Cloudshell을 사용할 수 있습니다.

cluster-layer.zip 파일을 생성하려면 먼저 python 라이브러리 패키지 세부 정보로 구성된 **Requirements.txt** 파일을 생성한 다음 셸 스크립트를 실행해야 합니다.

1. Python 패키지 세부 정보를 지정하여 **requirements.txt** 파일을 생성합니다.

   다음은 **requirements.txt** 파일에서 제공하는 샘플 패키지 세부 정보입니다.

   ```
   $ cat requirements.txt
   pycryptodome
   paramiko
   requests
   scp
   jsonschema
   cffi
   zipp
   importlib-metadata
   ```

2. 다음 셸 스크립트를 실행하여 **cluster_layer.zip** 파일을 생성합니다.

   ```
   $ pip3 install --platform manylinux2014_x86_64
   --target=./python/lib/python3.9/site-packages
   --implementation cp --python-version 3.9 --only-binary=:all:
   --upgrade -r requirements.txt
   $ zip -r cluster_layer.zip ./python
   ```

   참고 설치 중에 종속성 충돌 오류(예: urllib3 또는 암호화)가 발생하는 경우, **requirements.txt** 파일에 권장 버전과 함께 충돌 패키지를 포함하는 것이 좋습니다. 그런 다음 설치를 다시 실행하여 충돌을 해결할 수 있습니다.

e) 결과 **cluster_layer.zip** 파일을 lambda python files 폴더에 복사합니다.

f) **cluster_manager.zip** 및 **cluster_lifecycle.zip** 파일을 생성합니다.

make.py 파일은 복제된 리포지토리에 있습니다. 이렇게하면 python 파일을 Zip 파일로 압축하고 대상 폴더에 복사합니다.

python3 make.py build

단계 **2** **Infrastructure.yaml**을 구축하고 클러스터 구축에 대한 출력 값을 기록합니다.

a) AWS 콘솔에서 **CloudFormation**으로 이동하여 **Create stack(**스택 생성**)**을 클릭합니다. **With new resources (standard)(**새 리소스 포함**(**표준**))**를 선택합니다.

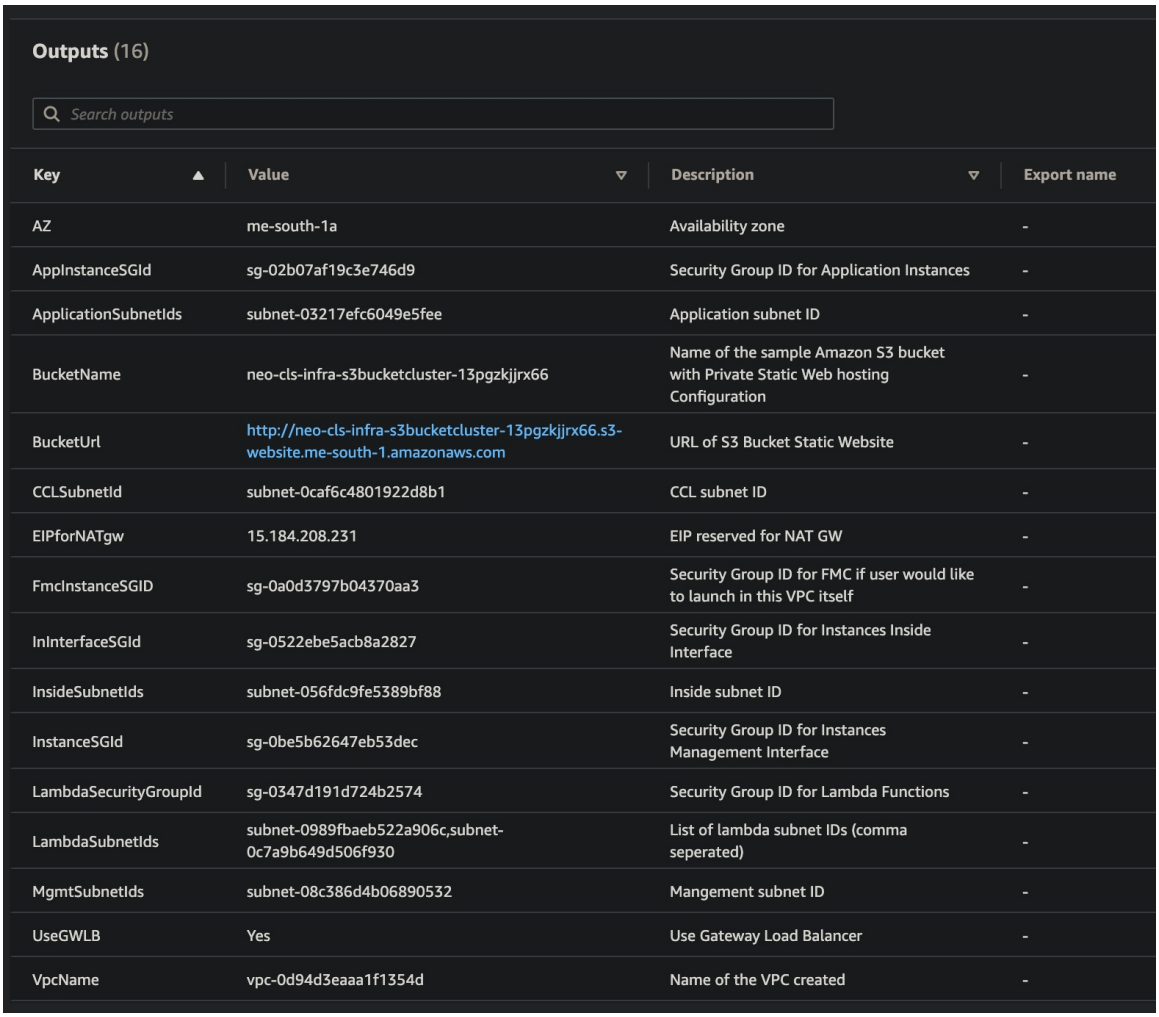
b) **Upload a template file(**템플릿 파일 업로드**)**을 선택하고 **Choose file(**파일 선택**)**을 클릭한 후 대상 폴더에서 **Infrastructure.yaml**을 선택합니다.

c) **Next(**다음**)**를 클릭하고 필수 정보를 제공합니다.

d) **Next(**다음**)**, **Create stack(**스택 생성**)**을 차례로 클릭합니다.

e) 구축이 완료되면 **Outputs(**출력**)**로 이동하여 S3 **BucketName**을 확인합니다.

그림 ***2: Infrastructure.yaml***의 출력



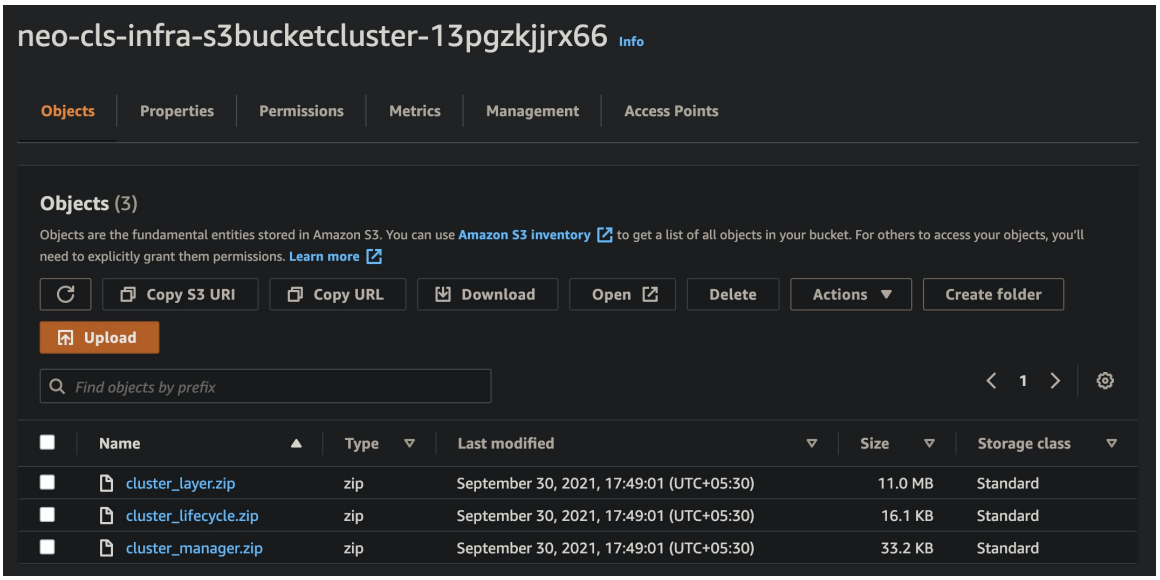
Outputs (16)

Search outputs

Key	Value	Description	Export name
AZ	me-south-1a	Availability zone	-
AppInstanceSGId	sg-02b07af19c3e746d9	Security Group ID for Application Instances	-
ApplicationSubnetIds	subnet-03217efc6049e5fee	Application subnet ID	-
BucketName	neo-cls-infra-s3bucketcluster-13pgzkjjrx66	Name of the sample Amazon S3 bucket with Private Static Web hosting Configuration	-
BucketUrl	http://neo-cls-infra-s3bucketcluster-13pgzkjjrx66.s3-website.me-south-1.amazonaws.com	URL of S3 Bucket Static Website	-
CCLSubnetId	subnet-0caf6c4801922d8b1	CCL subnet ID	-
EIPforNATgw	15.184.208.231	EIP reserved for NAT GW	-
FmcInstanceSGID	sg-0a0d3797b04370aa3	Security Group ID for FMC if user would like to launch in this VPC itself	-
InInterfaceSGId	sg-0522ebe5acb8a2827	Security Group ID for Instances Inside Interface	-
InsideSubnetIds	subnet-056fdc9fe5389bf88	Inside subnet ID	-
InstanceSGId	sg-0be5b62647eb53dec	Security Group ID for Instances Management Interface	-
LambdaSecurityGroupId	sg-0347d191d724b2574	Security Group ID for Lambda Functions	-
LambdaSubnetIds	subnet-0989fbaeb522a906c,subnet-0c7a9b649d506f930	List of lambda subnet IDs (comma seperated)	-
MgmtSubnetIds	subnet-08c386d4b06890532	Mangement subnet ID	-
UseGWLB	Yes	Use Gateway Load Balancer	-
VpcName	vpc-0d94d3eaaa1f1354d	Name of the VPC created	-

단계 **3** **cluster_layer.zip**, **cluster_lifecycle.zip** 및 **cluster_manager.zip**을 Infrastructure.yaml에서 생성한 S3 버킷에 업로드합니다.

그림 ***3: S3*** 버킷



단계 **4** **deploy_ngfw_cluster.yaml**을 구축합니다.

a) **CloudFormation**로 이동하고 **Create stack(**스택 생성**)**을 클릭합니다. **With new resources (standard)(**새 리소스 포함**(**표준**))**를 선택합니다.

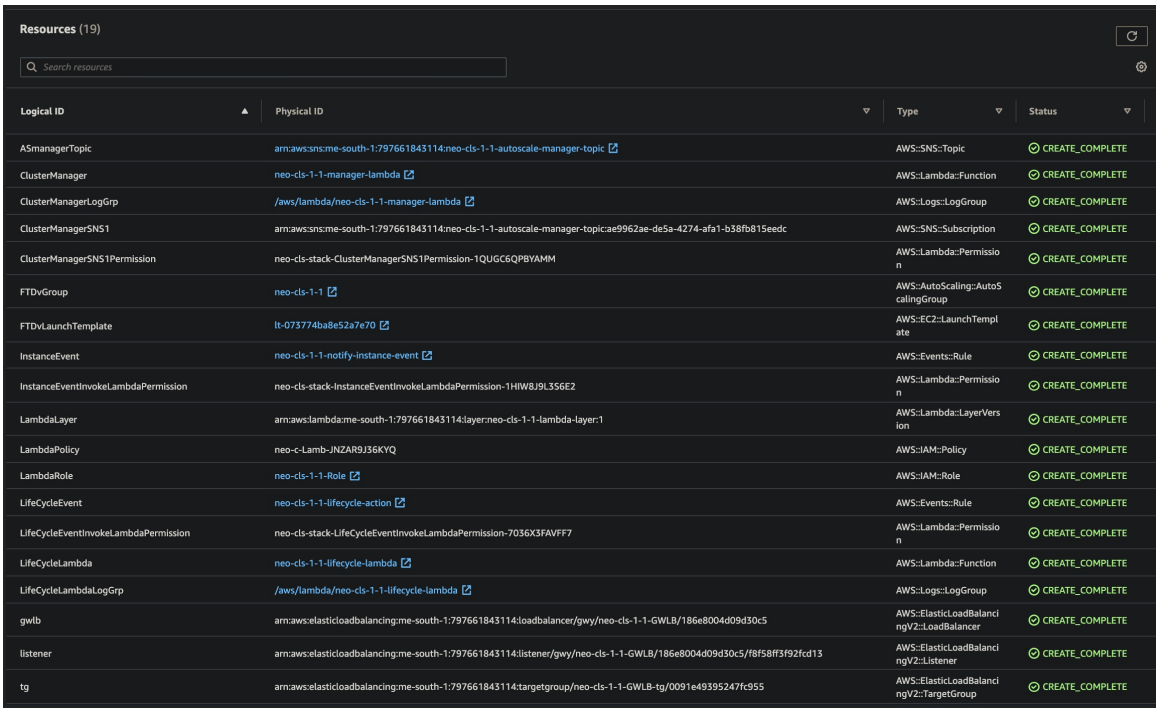
b) **Upload a template file(**템플릿 파일 업로드**)**을 선택하고 **Choose file(**파일 선택**)**을 클릭한 후 대상 폴더에서 **deploy_ngfw_cluster.yaml**을 선택합니다.

c) **Next(**다음**)**를 클릭하고 필수 정보를 제공합니다.

d) **Next(**다음**)**, **Create stack(**스택 생성**)**을 차례로 클릭합니다.

Lambda 함수는 프로세스의 나머지를 관리하며 threat defense virtual는 management center에 자동으로 등록됩니다.

그림 ***4:*** 구축된 리소스



단계 **5** 노드 중 하나에 로그인하고 **show cluster info** 명령을 사용하여 클러스터 구축을 확인합니다.

그림 ***5:*** 클러스터 노드

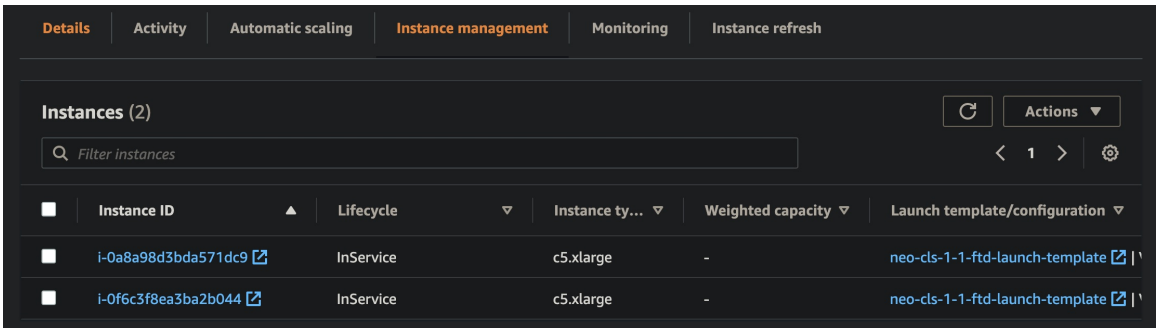


그림 ***6: show cluster info***

AWS에서 수동으로 클러스터 구축

클러스터를 수동으로 구축하려면 Day 0 구성을 준비하고 각 노드를 구축한 다음 management center에 제어 노드를 추가합니다.

AWS에 대한 **Day0** 구성 생성

고정 구성 또는 맞춤형 구성을 사용할 수 있습니다. 고정 구성을 사용하는 것이 좋습니다.

AWS에 대한 고정 구성으로 **Day0** 구성 생성

고정 구성은 클러스터 부트스트랩 구성을 자동으로 생성합니다.

```
{
    "AdminPassword": "password",
    "Hostname": "hostname",
    "FirewallMode": "Routed",
    "ManageLocally": "No",
    "Cluster": {
        "CclSubnetRange": "ip_address_start ip_address_end",
        "ClusterGroupName": "cluster_name",
        [For Gateway Load Balancer] "Geneve": "{Yes | No}",
```

```
        [For Gateway Load Balancer] "HealthProbePort": "port"
    }
}
```

대표적인 예는 다음과 같습니다.

```
{
 "AdminPassword": "Sup3rnatural",
 "Hostname": "ciscoftdv",
 "FirewallMode": "Routed",
 "ManageLocally": "No",
 "Cluster": {
  "CclSubnetRange": "10.10.55.4 10.10.55.30",     //mandatory user input
  "ClusterGroupName": "ftdv-cluster",             //mandatory user input
  "Geneve": "Yes",
  "HealthProbePort": "7777"
 }
}
```

참고 위의 설정을 복사하여 붙여넣는 경우에는 //필수 사용자 입력을 구성에서 제거해야 합니다.

CclSubnetRange 변수에 xxx4에서 시작하는 IP 주소 범위를 지정합니다. 클러스터링에 16개 이상의 사용 가능한 IP 주소가 있는지 확인합니다. 시작(*ip_address_start*) 및 종료(*ip_address_end*) IP 주소의 몇 가지 예는 아래에 나와 있습니다.

표 ***3:*** 시작 및 종료 ***IP*** 주소의 예

CIDR	시작 **IP** 주소	종료 **IP** 주소
10.1.1.0/27	10.1.1.4	10.1.1.30
10.1.1.32/27	10.1.1.36	10.1.1.62
10.1.1.64/27	10.1.1.68	10.1.1.94
10.1.1.96/27	10.1.1.100	10.1.1.126
10.1.1.128/27	10.1.1.132	10.1.1.158
10.1.1.160/27	10.1.1.164	10.1.1.190
10.1.1.192/27	10.1.1.196	10.1.1.222
10.1.1.224/27	10.1.1.228	10.1.1.254
10.1.1.0/24	10.1.1.4	10.1.1.254

AWS에 대한 사용자 지정 구성을 사용하여 **Day0** 구성 생성

명령을 사용하여 전체 클러스터 부트스트랩 구성을 입력할 수 있습니다.

```
{
    "AdminPassword": "password",
    "Hostname": "hostname",
```

```
    "FirewallMode": "Routed",
    "ManageLocally": "No",
    "run_config": [comma_separated_threat_defense_configuration]
    }
```

게이트웨이 로드 밸런서 예

다음 예에서는 U-turn 트래픽용 Geneve 인터페이스 1개 및 클러스터 제어 링크용 VXLAN 인터페이스 1개가 있는 게이트웨이 로드 밸런서에 대한 구성을 생성합니다. 굵게 표시된 값은 노드별로 고유해야 합니다.

버전 **7.4** 이상용 Day 0 구성 샘플이 아래에 나와 있습니다.

```
{
 "AdminPassword": "Sam&Dean",
 "Hostname": "ftdv1",
 "FirewallMode": "Routed",
 "ManageLocally": "No",
 "run_config": [
  "cluster interface-mode individual force",
  "interface TenGigabitEthernet0/0",
  "nameif geneve-vtep-ifc",
  "ip address dhcp",
  "no shutdown",
  "interface TenGigabitEthernet0/1",
  "nve-only cluster",
  "nameif ccl_link",
  "ip address dhcp",
  "no shutdown",
  "interface vni1",
  "description Clustering Interface",
  "segment-id 1",
  "vtep-nve 1",
  "interface vni2",
  "proxy single-arm",
  "nameif uturn-ifc",
  "vtep-nve 2",
  "object network ccl#link",
  "range 10.1.90.4 10.1.90.19",
  "object-group network cluster#group",
  "network-object object ccl#link",
  "nve 2",
  "encapsulation geneve",
  "source-interface geneve-vtep-ifc",
  "nve 1",
  "encapsulation vxlan",
  "source-interface ccl_link",
  "peer-group cluster#group",
  "jumbo-frame reservation",
  "mtu geneve-vtep-ifc 1826",
  "mtu ccl_link 1980",
  "cluster group ftdv-cluster",
  "local-unit 1",
  "cluster-interface vni1 ip 10.1.1.1 255.255.255.0",
  "priority 1",
  "enable",
  "aaa authentication listener http geneve-vtep-ifc port 7777"
 ]
}
```

버전 **7.3** 이하용 Day 0 구성 샘플이 아래에 나와 있습니다.

```
{
 "AdminPassword": "Sam&Dean",
 "Hostname": "ftdv1",
 "FirewallMode": "Routed",
 "ManageLocally": "No",
 "run_config": [
  "cluster interface-mode individual force",
  "interface TenGigabitEthernet0/0",
  "nameif geneve-vtep-ifc",
  "ip address dhcp",
  "no shutdown",
  "interface TenGigabitEthernet0/1",
  "nve-only cluster",
  "nameif ccl_link",
  "ip address dhcp",
  "no shutdown",
  "interface vni1",
  "description Clustering Interface",
  "segment-id 1",
  "vtep-nve 1",
  "interface vni2",
  "proxy single-arm",
  "nameif uturn-ifc",
  "vtep-nve 2",
  "object network ccl#link",
  "range 10.1.90.4 10.1.90.19",
  "object-group network cluster#group",
  "network-object object ccl#link",
  "nve 2",
  "encapsulation geneve",
  "source-interface geneve-vtep-ifc",
  "nve 1",
  "encapsulation vxlan",
  "source-interface ccl_link",
  "peer-group cluster#group",
  "jumbo-frame reservation",
  "mtu geneve-vtep-ifc 1806",
  "mtu ccl_link 1960",
  "cluster group ftdv-cluster",
  "local-unit 1",
  "cluster-interface vni1 ip 10.1.1.1 255.255.255.0",
  "priority 1",
  "enable",
  "aaa authentication listener http geneve-vtep-ifc port 7777"
 ]
}
```

참고 CCL 서브넷 범위의 경우 CCL 서브넷 CIDR에서 예약된 IP 주소를 제외하고 IP 주소를 지정합니다. 몇 가지 예시는 위의 표 3: 시작 및 종료 IP 주소의 예를 참고하십시오.

AWS 상태 확인 설정의 경우 여기에서 설정한 **aaa authentication listener http** 포트를 지정해야 합니다.

비 기본 로드 밸런서 예

다음 예에서는 관리, 내부 및 외부 인터페이스가 있는 비 기본 로드 밸런서에 사용할 구성과 클러스터 제어 링크용 VXLAN 인터페이스를 생성합니다. 굵게 표시된 값은 노드별로 고유해야 합니다.

```
{
 "AdminPassword": "W1nch3sterBr0s",
 "Hostname": "ftdv1",
 "FirewallMode": "Routed",
 "ManageLocally": "No",
 "run_config": [
  "cluster interface-mode individual force",
  "interface Management0/0",
  "management-only",
  "nameif management",
  "ip address dhcp",
  "interface GigabitEthernet0/0",
  "no shutdown",
  "nameif outside",
  "ip address dhcp",
  "interface GigabitEthernet0/1",
  "no shutdown",
  "nameif inside",
  "ip address dhcp",
  "interface GigabitEthernet0/2",
  "nve-only cluster",
  "nameif ccl_link",
  "ip address dhcp",
  "no shutdown",
  "interface vni1",
  "description Clustering Interface",
  "segment-id 1",
  "vtep-nve 1",
  "jumbo-frame reservation",
  "mtu ccl_link 1654",
  "object network ccl#link",
  "range 10.1.90.4 10.1.90.19",          //mandatory user input
  "object-group network cluster#group",
  "network-object object ccl#link",
  "nve 1",
  "encapsulation vxlan",
  "source-interface ccl_link",
  "peer-group cluster#group",
  "cluster group ftdv-cluster",          //mandatory user input
  "local-unit 1",
  "cluster-interface vni1 ip 10.1.1.1 255.255.255.0",
  "priority 1",
  "enable"
 ]
}
```

클러스터 제어 링크 네트워크 개체의 경우, 필요한 만큼의 주소만 지정합니다(최대 16개). 범위가 클수록 성능에 영향을 줄 수 있습니다.

참고 위의 설정을 복사하여 붙여넣는 경우에는 //필수 사용자 입력을 구성에서 제거해야 합니다.

클러스터 노드 구축

클러스터를 구성하도록 클러스터 노드를 구축합니다.

프로시저

단계 **1** 필요한 인터페이스 수(게이트웨이 로드 밸런서(GLLB)를 사용하는 경우 인터페이스 4개, 비 기본 로드 밸런서를 사용하는 경우 인터페이스 5개)로 클러스터 Day 0 구성을 사용하여 Threat Defense Virtual 인스턴스를 구축합니다. 이렇게 하려면 **Configure Instance Details(**인스턴스 세부 정보 구성**) > Advanced Details(**고급 세부 정보**)** 섹션에 Day 0 구성을 붙여 넣습니다.

참고 아래의 순서대로 인스턴스에 인터페이스를 연결합니다.

- AWS 게이트웨이 로드 밸런서 - 인터페이스 4개 - 관리, 진단, 내부 및 클러스터 제어 링크.
- 비 기본 로드 밸런서 - 인터페이스 5개 - 관리, 진단, 내부, 외부 및 클러스터 제어 링크.

AWS에서 Threat Defense Virtual을 구축하는 방법에 대한 자세한 내용은 Deploy Threat Defense Virtual on AWS(AWS에서 Threat Defense Virtual 구축)를 참고하십시오.

단계 **2** 필요한 수의 추가 노드를 구축하려면 1단계를 반복합니다.

단계 **3** Threat Defense Virtual 콘솔에서 **show cluster info** 명령을 사용하여 모든 노드가 클러스터에 성공적으로 조인되었는지 확인합니다.

단계 **4** AWS 게이트웨이 로드 밸런서를 구성합니다.

a) 대상 그룹 및 GWLB를 생성합니다.

b) GWLB에 대상 그룹을 연결합니다.

참고 올바른 보안 그룹, 리스너 구성 및 상태 확인 설정을 사용하도록 GWLB를 구성하십시오.

c) IP 주소를 사용하여 대상 그룹에 데이터 인터페이스(인터페이스 내부)를 등록합니다.

자세한 내용은 게이트웨이 로드 밸런서 생성을 참고하십시오.

단계 **5** Management Center에 제어 노드를 추가합니다. Management Center에 클러스터 추가(수동 구축), 54 페이지의 내용을 참조하십시오.

AWS에서 **GWLB**를 사용하는 **Secure Firewall Threat Defense Virtual** 클러스터링을 위한 대상 페일오버 구성

AWS의 Threat Defense Virtual 클러스터링은 GFLB(게이트웨이 로드 밸런서)를 사용하여 검사를 위해 네트워크 패킷의 균형을 유지하고 지정된 Threat Defense Virtual 노드로 전달합니다. GLLB는 대상 노드의 페일오버 또는 등록 취소 이벤트가 발생하는 경우 대상 노드로 네트워크 패킷을 계속 전송하도록 설계됩니다.

AWS의 대상 페일오버 기능을 사용하면 계획된 유지 관리 또는 대상 노드 오류 중에 노드 등록이 취소되는 경우 GBLB에서 네트워크 패킷을 정상 대상 노드로 리디렉션할 수 있습니다. 클러스터의 스테이트풀 장애 조치를 활용합니다.

AWS에서는 AWS ELB(Elastic Load Balancing) API 또는 AWS 콘솔을 통해 대상 페일오버를 구성할 수 있습니다.

참고 게이트웨이가 SSH, SCP, CURL 등의 특정 프로토콜을 사용하여 트래픽을 라우팅하는 동안 대상 노드에 오류가 발생하면 트래픽을 정상 대상으로 리디렉션하는 데 지연이 발생할 수 있습니다. 이 지연은 트래픽 플로우의 재밸런싱 및 경로 재지정으로 인한 것입니다.

AWS에서는 AWS ELB API 또는 AWS 콘솔을 통해 대상 페일오버를 구성할 수 있습니다.

- AWS API - AWS ELB API - *modify-target-group-attributes*에서 다음 두 가지 새로운 매개변수를 수정하여 플로우 처리 동작을 정의할 수 있습니다.
  - target_failover.on_unhealthy - 대상이 비정상 상태가 될 때 GWLB가 네트워크 흐름을 처리하는 방법을 정의합니다.
  - target_failover.on_deregistration - 대상이 등록 취소될 때 GWLB가 네트워크 흐름을 처리하는 방법을 정의합니다.

  다음 명령은 이러한 두 매개변수를 정의하는 샘플 API 매개변수 구성을 보여줍니다.

```
aws elbv2 modify-target-group-attributes \
--target-group-arn arn:aws:elasticloadbalancing:…/my-targets/73e2d6bc24d8a067 \
--attributes \
Key=target_failover.on_unhealthy, Value=rebalance[no_rebalance] \
Key=target_failover.on_deregistration, Value=rebalance[no_rebalance]
```

  자세한 내용은 AWS 설명서의 TargetGroupAttribute를 참조하십시오.

- AWS 콘솔 – EC2 콘솔에서 다음 옵션을 구성하여 Target Group(대상 그룹) 페이지에서 Target Failover(대상 페일오버) 옵션을 활성화할 수 있습니다.
  - 대상 그룹 속성 편집
  - 대상 페일오버 활성화
  - 리밸런싱 플로우 확인

  대상 페일오버를 활성화하는 방법에 대한 자세한 내용은 AWS에서 Secure Firewall Threat Defense Virtual 클러스터링을 위한 대상 페일오버 활성화, 23 페이지 섹션을 참조하십시오.

AWS에서 Secure Firewall Threat Defense Virtual 클러스터링을 위한 대상 페일오버 활성화

threat defense virtual의 데이터 인터페이스는 AWS에서 GWLB의 대상 그룹에 등록됩니다. threat defense virtual 클러스터링에서 각 인스턴스는 대상 그룹과 연결됩니다. GWLB는 대상 그룹에서 대상 노드로 식별되거나 등록된 이 정상 인스턴스로 트래픽을 로드 밸런싱하 고 전송합니다.

시작하기 전에

수동 방법으로 또는 CloudFormation 템플릿을 사용하여 AWS에 클러스터를 구축해야 합니다.

CloudFormation 템플릿을 사용하여 클러스터를 구축하는 경우 클러스터 구축 파일(`deploy_ftdv_clustering.yaml`)의 **GWLB** 구성 섹션에서 사용 가능한 **rebalance** 속성을 할당하여 대상 페일오버 매개변수를 활성화할 수도 있습니다. 템플릿에서 이 매개변수의 값은 기본적으로 **rebalance**로 설정됩니다. 그러나 AWS 콘솔에서 이 매개변수의 기본값은 **no_rebalance**로 설정됩니다.

여기서,

- **no_rebalance** - GWLB가 실패하거나 등록 취소된 대상으로 네트워크 흐름을 계속 전송합니다.
- **rebalance** - GWLB는 기존 대상이 실패하거나 등록 취소된 경우 네트워크 흐름을 다른 정상 대상으로 전송합니다.

AWS에서 스택을 구축하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS에서 수동으로 클러스터 구축
- CloudFormation 템플릿을 사용하여 AWS에서 스택 구축

프로시저

단계 **1** AWS 콘솔에서 **Services(**서비스**) > EC2**

단계 **2** 대상 그룹 페이지를 보려면 **Target Groups(**대상 그룹**)**를 클릭합니다.

단계 **3** threat defense virtual 데이터 인터페이스 IP가 등록된 대상 그룹을 선택합니다. 대상 페일오버 특성을 활성화할 수 있는 대상 그룹 세부 정보 페이지가 표시됩니다.

단계 **4** **Attributes(**속성**)** 메뉴로 이동합니다.

단계 **5** 속성을 편집하려면 **Edit(**편집**)**를 클릭합니다.

단계 **6** **Rebalance flows(**플로우 재조정**)** 슬라이더 버튼을 오른쪽으로 전환하여 대상 페일오버를 활성화함으로써 대상 페일오버 또는 등록 취소 시 GWLB가 기존 네트워크 패킷의 균형을 재조정하여 정상적인 대상 노드로 전달하도록 구성합니다.

Azure에서 클러스터 구축

Azure 게이트웨이 로드 밸런서(GWLB) 또는 기본이 아닌 로드 밸런서와 함께 클러스터를 사용할 수 있습니다. Azure에서 클러스터를 구축하려면 ARM(Azure Resource Manager) 템플릿을 사용하여 가상 머신 확장 집합을 구축할 수 있습니다.

GWLB 기반 클러스터 구축을 위한 샘플 토폴로지

그림 ***7: GWLB***를 사용하는 인바운드 트래픽 활용 사례 및 토폴로지

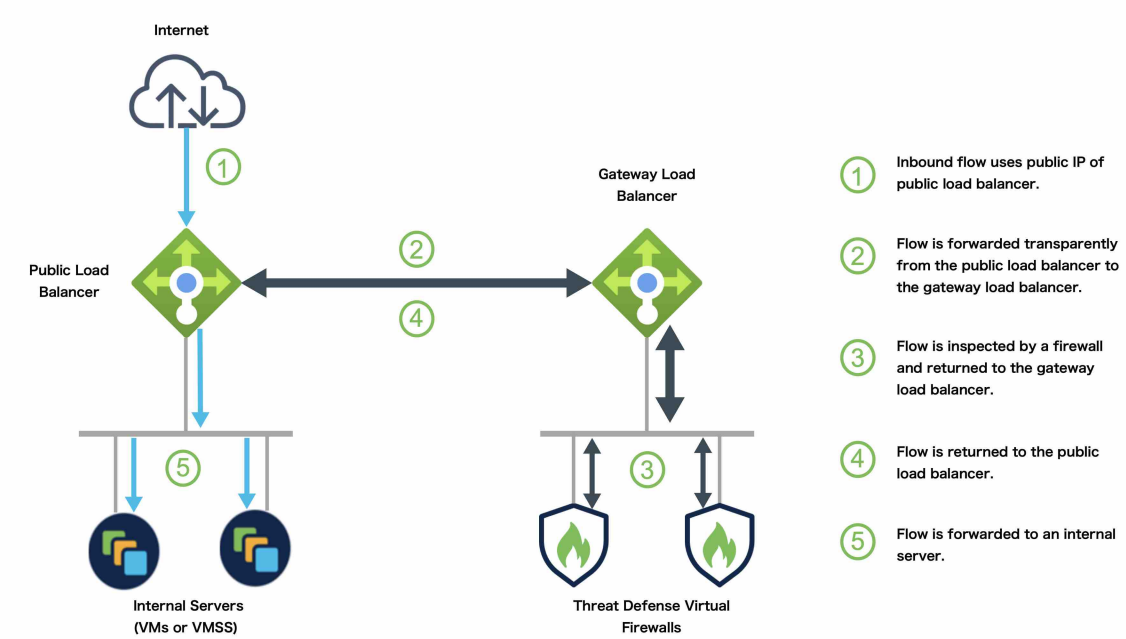
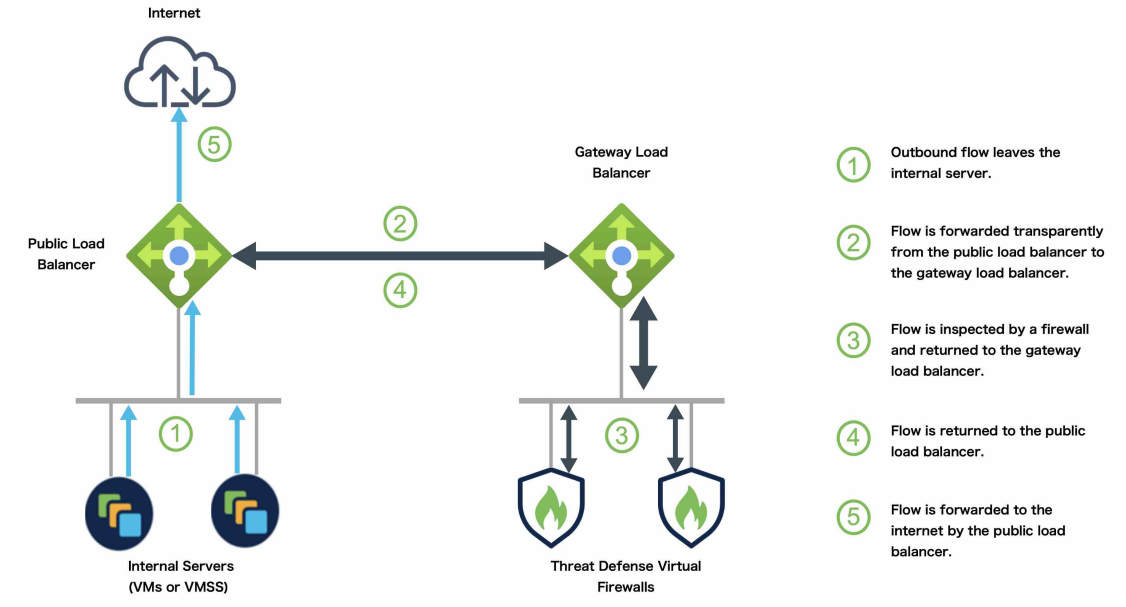


그림 ***8: GWLB***를 사용하는 아웃바운드 트래픽 활용 사례 및 토폴로지

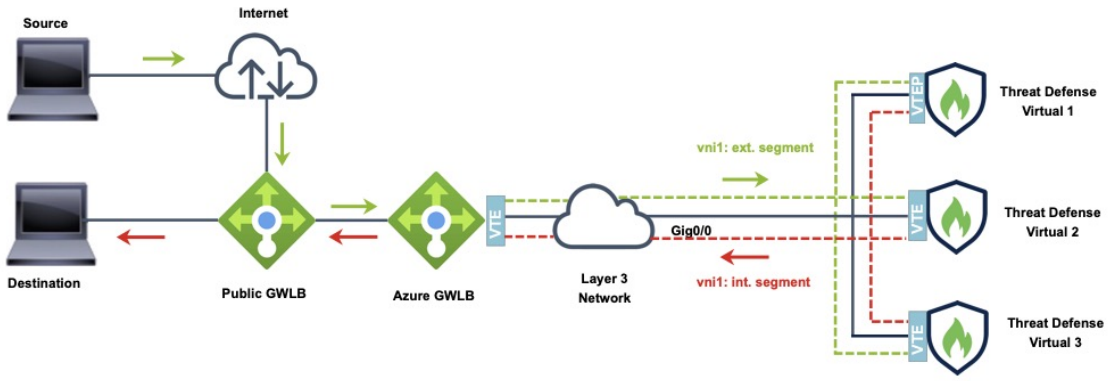


Azure 게이트웨이 로드 밸런서 및 페어링된 프록시

Azure 서비스 체인에서 Threat Defense Virtual은 인터넷과 고객 서비스 간의 패킷을 인터셉트할 수 있는 투명 게이트웨이 역할을 합니다. Threat Defense Virtual은 페어링된 프록시에서 VXLAN 세그먼트를 활용하여 단일 NIC에서 외부 인터페이스 및 내부 인터페이스를 정의합니다.

다음 그림에서는 외부 VXLAN 세그먼트의 공용 게이트웨이 로드 밸런서에서 Azure 게이트웨이 로드 밸런서로 전달되는 트래픽을 보여줍니다. 게이트웨이 로드 밸런서는 여러 Threat Defense Virtual 간에 트래픽을 밸런싱하며, 이를 삭제하거나 내부 VXLAN 세그먼트에서 게이트웨이 로드 밸런서로 다시 전송하기 전에 트래픽을 검사합니다. 그런 다음 Azure 게이트웨이 로드 밸런서는 퍼블릭 게이트웨이 로드 밸런서 및 대상으로 트래픽을 다시 전송합니다.

그림 ***9:*** 페어링된 프록시가 있는 ***Azure*** 게이트웨이 로드 밸런서

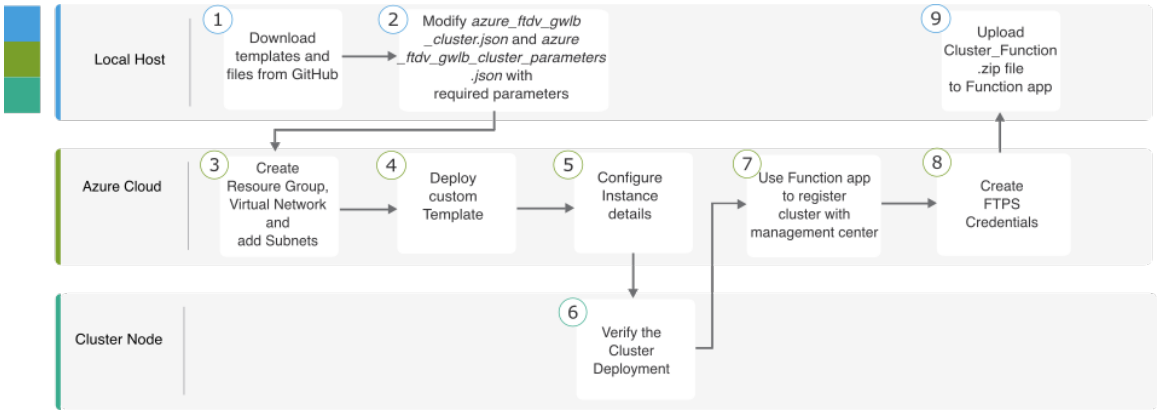


Traffic flow between GWLBe to GWLB (Geneve Single-Arm Proxy) in Azure

GWLB를 사용하여 **Azure**에서 **Threat Defense Virtual** 클러스터를 구축하기 위한 **End-to-End** 프로세스

템플릿 기반 구축

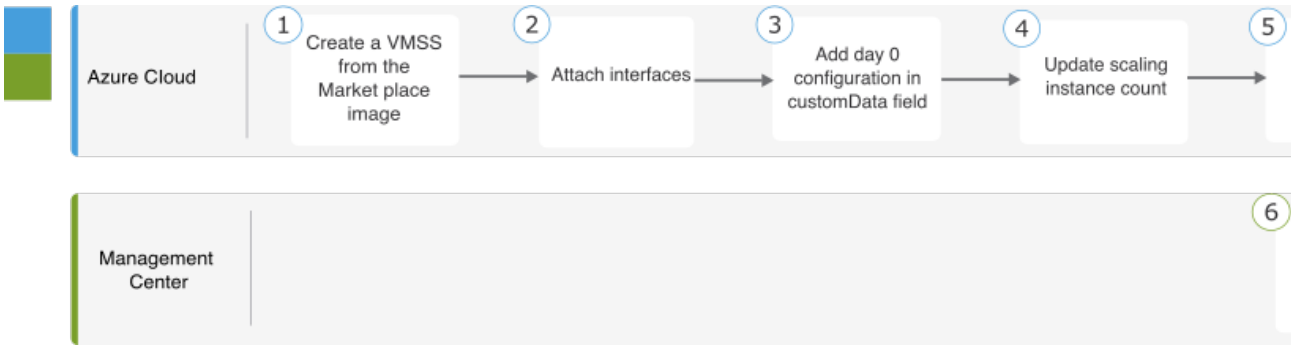
다음 순서도에는 GWLB를 사용하여 Azure에서 Threat Defense Virtual 클러스터를 템플릿 기반으로 구축하는 워크플로우가 나와 있습니다.



	업무 환경	단계
1	로컬 호스트	GitHub에서 템플릿 및 파일을 다운로드합니다.
2	로컬 호스트	필수 매개변수를 사용하여 *azure_ftdv_gwlb_cluster.json* 및 *azure_ftdv_gwlb_cluster_parameters.json*을 수정합니다.
3	Azure Cloud	리소스 그룹, 가상 네트워크, 서브넷을 생성합니다.
4	Azure Cloud	사용자 지정 템플릿을 구축합니다.
5	Azure Cloud	인스턴스 세부 정보를 구성합니다.
6	클러스터 노드	클러스터 구축을 확인합니다.
7	Azure Cloud	Function 앱을 사용하여 클러스터를 Management Center에 등록합니다.
8	Azure Cloud	FTPS 자격 증명을 생성합니다.
9	로컬 호스트	*Cluster_Function.zip* 파일을 Function 앱에 업로드합니다.

수동 구축

다음 순서도에는 GWLB를 사용하여 Azure에서 Threat Defense Virtual 클러스터를 수동으로 구축하는 워크플로우가 나와 있습니다.



	업무 환경	단계
1	로컬 호스트	Marketplace 이미지에서 VMSS를 생성합니다.
2	로컬 호스트	인터페이스를 연결합니다.
3	로컬 호스트	customData 필드에 Day 0 구성을 추가합니다.
4	로컬 호스트	확장 인스턴스 수를 업데이트합니다.
5	로컬 호스트	GLLB를 구성합니다.
6	Management Center	제어 노드를 추가합니다.

템플릿

아래에 제공된 템플릿은 GitHub에서 사용할 수 있습니다. 매개변수 값은 템플릿에 지정된 매개변수 이름 및 값을 통해 이해할 수 있습니다.

Secure Firewall 버전 7.4.1부터 진단 인터페이스 없이 클러스터를 구축할 수 있습니다. Outside, Inside, Management 및 CCL 인터페이스만 사용하여 클러스터를 구축하려면 WithoutDiagnostic 템플릿 - azure_withoutDiagnostic_ftdv_gwlb_cluster_parameters.json 및 azure_withoutDiagnostic_ftdv_gwlb_cluster.json 파일을 사용합니다.

진단 인터페이스로 구축할 템플릿:

- azure_ftdv_gwlb_cluster_parameters.json – GWLB를 사용하는 Threat Defense Virtual 클러스터에 대한 매개변수를 입력하는 템플릿.
- azure_ftdv_gwlb_cluster.json – GWLB를 사용하는 Threat Defense Virtual 클러스터를 구축하는 템플릿.

진단 인터페이스 없이 구축할 템플릿:

- azure_withoutDiagnostic_ftdv_gwlb_cluster_parameters.json - 진단 인터페이스 없이 GWLB 구축을 사용하는 Threat Defense Virtual 클러스터에 대한 매개변수를 입력하는 템플릿.

- azure_withoutDiagnostic_ftdv_grlb_cluster.json – 진단 인터페이스 없이 GWLB 를 사용하는 Threat Defense Virtual 클러스터를 구축하는 템플릿.

사전 요구 사항

- 클러스터가 Management Center에 자동 등록되도록 허용하려면 Management Center에서 네트워크 관리자 및 유지 보수 사용자 권한을 가진 사용자를 생성합니다. 이러한 권한이 있는 사용자는 REST API를 사용할 수 있습니다. Cisco Secure Firewall Management Center 관리 가이드를 참고하십시오.
- 템플릿 구축 중에 지정할 정책의 이름과 일치하는 액세스 정책을 Management Center에 추가합니다.
- Management Center Virtual에 적절한 라이선스가 부여되었는지 확인합니다.
- 클러스터가 Management Center Virtual에 추가된 후 아래의 단계를 수행합니다.
  1. Management Center에서 상태 확인 포트 번호로 플랫폼 설정을 구성합니다. 이 구성에 대한 자세한 내용은 플랫폼 설정을 참고하십시오.
  2. 데이터 트래픽에 대한 고정 경로를 생성합니다. 고정 경로 생성에 대한 자세한 내용은 고정 경로 추가를 참고하십시오.

     고정 경로 구성 샘플:

```
Network: any-ipv4
Interface: vxlan_tunnel
Leaked from Virtual Router: Global
Gateway: vxlan_tunnel_gw
Tunneled: false
Metric: 2
```

참고 *vxlan_tunnel_gw*는 데이터 서브넷의 게이트웨이 IP 주소입니다.

Azure Resource Manager 템플릿을 통해 GWLB를 사용하여 Azure에서 클러스터 구축

사용자 지정된 ARM(Azure Resource Manager) 템플릿을 사용하여 Azure GWLB용 가상 시스템 확장 세트를 배포합니다.

프로시저

단계 **1** 템플릿을 준비합니다.

a) 로컬 폴더에 github 리포지토리를 복제합니다. https://github.com/CiscoDevNet/cisco-ftdv/tree/master/cluster/azure을 참조하십시오.

b) 필수 매개변수를 사용하여 *azure_ftdv_gwlb_cluster.json* 및 *azure_ftdv_gwlb_cluster_parameters.json*을 수정합니다.

또는

진단 인터페이스 없이 클러스터를 구축하기 위한 필수 매개변수를 사용하여withoutDiagnostic 템플릿, *azure_withoutDiagnostic_ftdv_gwlb_cluster_parameters.json* 및 *azure_withoutDiagnostic_ftdv_gwlb_cluster.json*을 수정합니다.

단계 **2** Azure 포털: https://portal.azure.com에 로그인합니다.

단계 **3** 리소스 그룹을 생성합니다.

a) **Basics(**기본 사항**)** 탭의 드롭다운 목록에서 **Subscription(**구독**)** 및 **Resource Group(**리소스 그룹**)**을 선택합니다.

b) 필요한 **Region(**지역**)**을 선택합니다.

단계 **4** 4개의 서브넷(관리, 진단, 외부 및 CCL(Cluster Control Link))이 있는 가상 네트워크를 생성합니다.

Secure Firewall 버전 7.4.1부터 진단 인터페이스 없이 클러스터를 구축할 수 있습니다. Outside, Inside, Management 및 CCL 인터페이스만 사용하여 클러스터를 구축하려면 WithoutDiagnostic 템플릿 - azure_withoutDiagnostic_ftdv_gwlb_cluster_parameters.json 및 azure_withoutDiagnostic_ftdv_gwlb_cluster.json 파일을 사용합니다.

a) 가상 네트워크를 생성합니다.

1. **Basics(**기본 사항**)** 탭의 드롭다운 목록에서 **Subscription(**구독**)** 및 **Resource Group(**리소스 그룹**)**을 선택합니다.
2. 필요한 **Region(**지역**)**을 선택합니다. **Next: IP addresses(**다음**: IP** 주소**)**를 선택합니다.

IP Addresses(IP 주소**)** 탭에서 **Add subnet(**서브넷 추가**)**을 클릭하고 서브넷(관리, 진단, 데이터 및 CCL)을 추가합니다.

진단 인터페이스 없이 Threat Defense Virtual 7.4.1 클러스터를 구축하는 경우 진단 서브넷 생성을 건너뛰어야 합니다.

b) 서브넷을 추가합니다.

단계 **5** 사용자 지정 템플릿을 구축합니다.

a) **Create(**생성**)** > **Template deployment(**템플릿 구축**)(**사용자 지정 템플릿을 사용하여 구축**)**을 클릭합니다.

b) **Build your own template in the editor(**편집기에서 자체 템플릿 구축**)**를 클릭합니다.

c) 진단 인터페이스 구축 없이 를 선택한 경우 **Load File(**파일 로드**)**을 클릭하고 **azure_ftdv_grlb_cluster.json** 또는 **azure_withoutDiagnostic_ftdv_gwlb_cluster.json**를 업로드합니다.

d) **Save(**저장**)**를 클릭합니다.

단계 **6** 인스턴스 세부 정보를 구성합니다.

a) 필요한 값을 입력하고 **Review + create(**검토 + 생성**)**를 클릭합니다.

b) 검증이 통과되면 **Create(**생성**)**를 클릭합니다.

단계 **7** 인스턴스를 실행한 후 노드 중 하나에 로그인하고 **show cluster info** 명령을 입력하여 클러스터 구축을 확인합니다.

그림 ***10: show cluster info***

```
> show cluster info
Cluster gwlb-cluster-template-with-AN: On
    Interface mode: individual
Cluster Member Limit : 16
    This is "12" in state CONTROL_NODE
        ID        : 0
        Version   : 99.19(1)180
        Serial No.: 9AKGFV8VH4G
        CCL IP    : 10.1.1.12
        CCL MAC   : 000d.3a55.5470
        Module    : NGFWv
        Resource  : 8 cores / 28160 MB RAM
        Last join : 11:13:24 UTC Sep 5 2022
        Last leave: N/A
```

단계 **8** Azure 포털에서 Function 앱을 클릭하여 클러스터를 Management Center에 등록합니다.

참고 함수 앱을 사용하지 않으려면 **Add(**추가**)** > **Device(**디바이스**)(Add(**추가**)** > **Cluster(**클러스터**)**가 아님)를 사용하여 management center에 직접 제어 노드를 등록할 수도 있습니다. 나머지 클러스터 노드는 자동으로 등록됩니다.

단계 **9** **Deployment Center(**구축 센터**)** > **FTPS credentials(FTPS** 자격 증명**)** > **User scope(**사용자 범위**)** > **Configure Username and Password(**사용자 이름 및 비밀번호 구성**)**를 클릭한 다음 **Save(**저장**)**를 클릭합니다.

단계 **10** 로컬 터미널에서 다음 **curl** 명령을 실행하여 Cluster_Function.zip 파일을 Function 앱에 업로드합니다.

curl -X POST -u *username* **--data-binary @"Cluster_Function.zip" https://** *Function_App_Name***.scm.azurewebsites.net/api/zipdeploy**

참고 **curl** 명령 실행을 완료하는 데 몇 분(2~3분)이 걸릴 수 있습니다.

함수가 함수 앱에 업로드됩니다. 함수가 시작되고 스토리지 계정의 outqueue에서 로그를 볼 수 있습니다. Management Center에 대한 디바이스 등록이 시작됩니다.

그림 ***11:*** 카탈로그

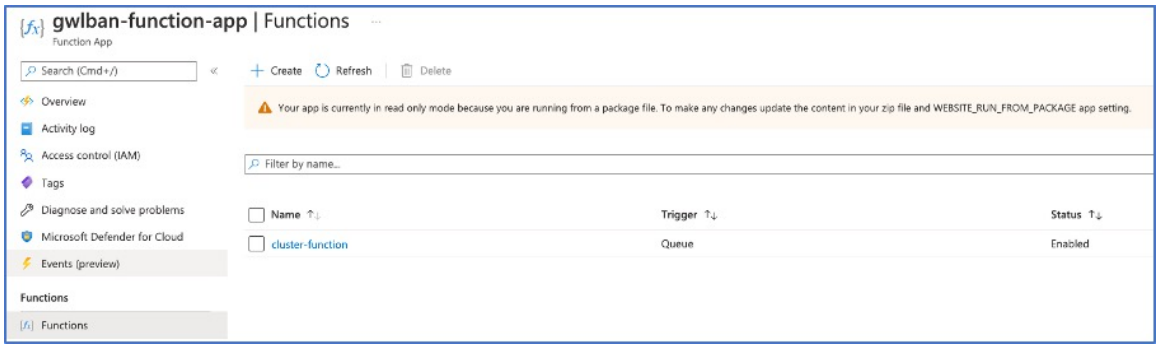


그림 ***12:*** 큐

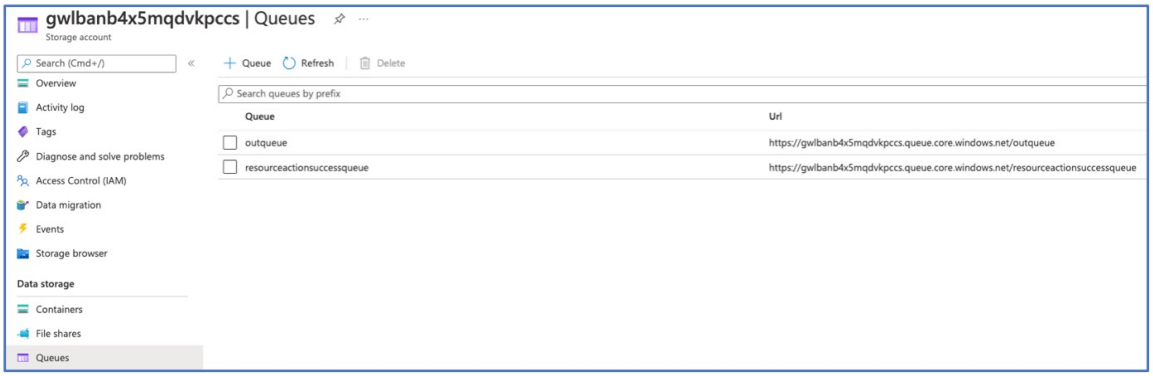
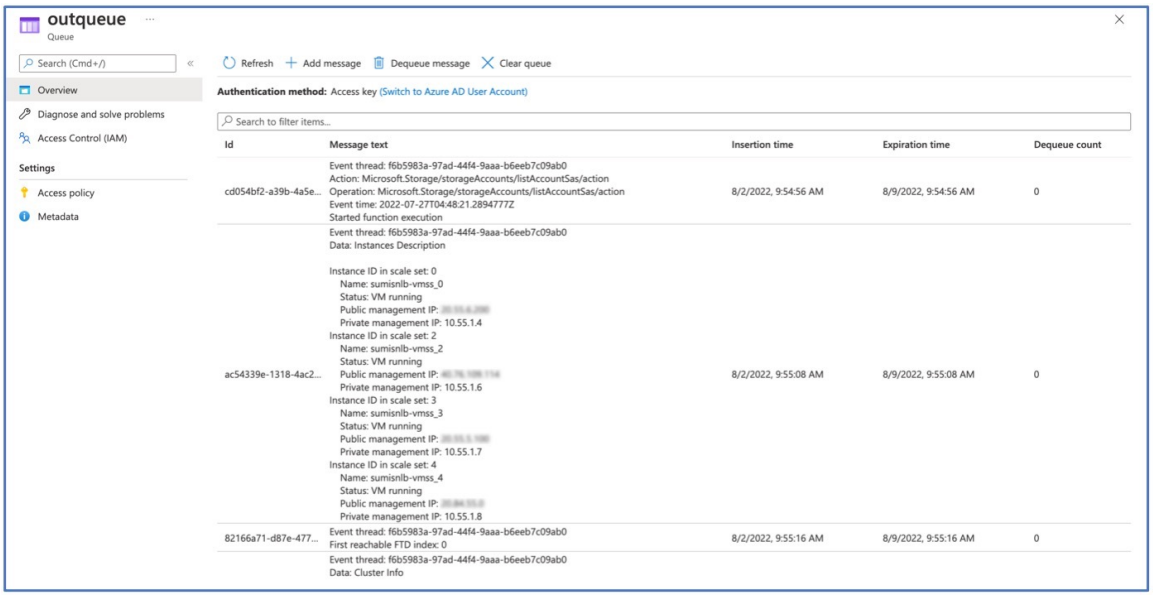
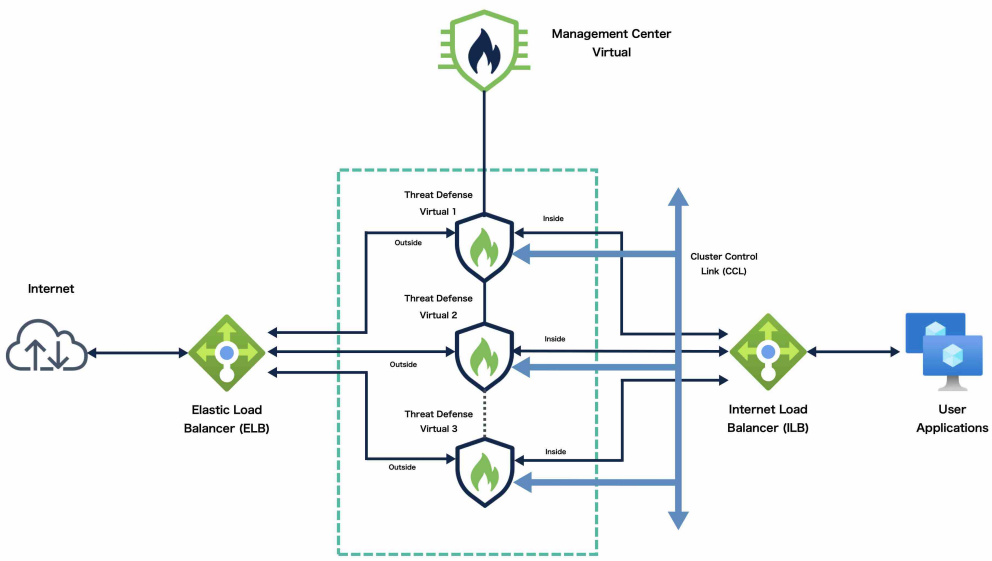


그림 ***13: Outqueue***



NLB 기반 클러스터 구축을 위한 샘플 토폴로지



이 토폴로지에는 인바운드 및 아웃바운드 트래픽 플로우가 모두 나와 있습니다. Threat Defense Virtual 클러스터는 내부 로드 밸런서와 외부 로드 밸런서 사이에 위치합니다. Management Center Virtual 인스턴스는 클러스터를 관리하는 데 사용됩니다.

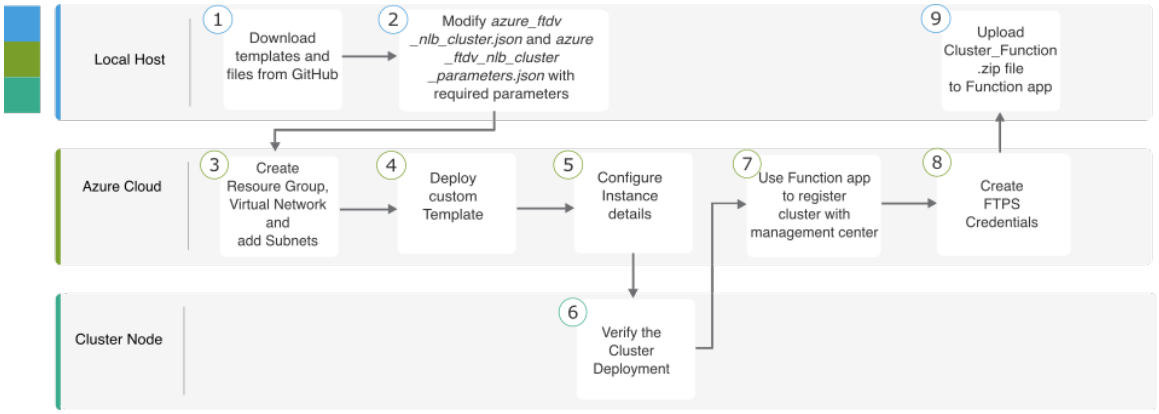
인터넷의 인바운드 트래픽은 외부 로드 밸런서로 이동된 다음 Threat Defense Virtual 클러스터로 전송됩니다. 트래픽은 클러스터의 Threat Defense Virtual 인스턴스에서 검사된 후 애플리케이션 VM으로 전달됩니다.

애플리케이션 VM의 아웃바운드 트래픽은 내부 로드 밸런서로 전송됩니다. 그런 다음 트래픽은 Threat Defense Virtual 클러스터로 전달된 후 인터넷으로 전송됩니다.

NLB를 사용하여 **Azure**에서 **Threat Defense Virtual** 클러스터를 구축하기 위한 **End-to-End** 프로세스

템플릿 기반 구축

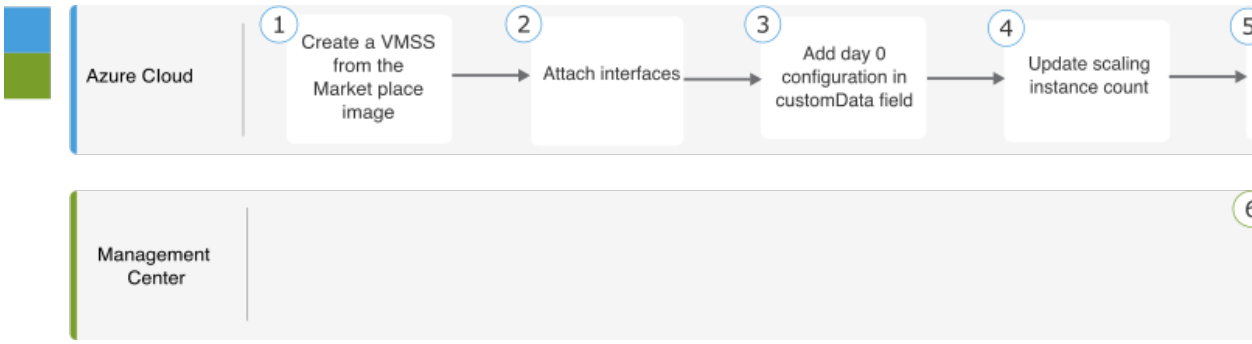
다음 순서도에는 NLB를 사용하여 Azure에서 Threat Defense Virtual 클러스터를 템플릿 기반으로 구축하는 워크플로우가 나와 있습니다.



	업무 환경	단계
1	로컬 호스트	GitHub에서 템플릿 및 파일을 다운로드합니다.
2	로컬 호스트	필수 매개변수를 사용하여 *azure_ftdv_nlb_cluster.json* 및 *azure_ftdv_nlb_cluster_parameters.json*을 수정합니다.
3	Azure Cloud	리소스 그룹, 가상 네트워크, 서브넷을 생성합니다.
4	Azure Cloud	사용자 지정 템플릿을 구축합니다.
5	Azure Cloud	인스턴스 세부 정보를 구성합니다.
6	클러스터 노드	클러스터 구축을 확인합니다.
7	Azure Cloud	Function 앱을 사용하여 클러스터를 Management Center에 등록합니다.
8	Azure Cloud	FTPS 자격 증명을 생성합니다.
9	로컬 호스트	*Cluster_Function.zip* 파일을 Function 앱에 업로드합니다.

수동 구축

다음 순서도에는 NLB를 사용하여 Azure에서 Threat Defense Virtual 클러스터를 수동으로 구축하는 워크플로우가 나와 있습니다.



	업무 환경	단계
1	로컬 호스트	Marketplace 이미지에서 VMSS를 생성합니다.
2	로컬 호스트	인터페이스를 연결합니다.
3	로컬 호스트	customData 필드에 Day 0 구성을 추가합니다.
4	로컬 호스트	확장 인스턴스 수를 업데이트합니다.
5	로컬 호스트	NLB를 구성합니다.
6	Management Center	제어 노드를 추가합니다.

템플릿

아래에 제공된 템플릿은 GitHub에서 사용할 수 있습니다. 매개변수 값은 템플릿에 지정된 매개변수 이름 및 값을 통해 이해할 수 있습니다.

Secure Firewall 버전 7.4.1부터 진단 인터페이스 없이 클러스터를 구축할 수 있습니다. Outside, Inside, Management 및 CCL 인터페이스만 있는 클러스터를 구축하려면 WithoutDiagnostic 템플릿 - azure_withoutDiagnostic_ftdv_nlb_cluster_parameters.json 및 azure_withoutDiagnostic_ftdv_nlb_cluster.json 파일을 사용합니다.

진단 인터페이스로 구축할 템플릿:

- azure_ftdv_nlb_cluster_parameters.json – NLB를 사용하는 Threat Defense Virtual 클러스터에 대한 매개변수를 입력하는 템플릿
- azure_ftdv_nlb_cluster.json – NLB를 사용하는 Threat Defense Virtual 클러스터를 구축하기 위한 템플릿

진단 인터페이스 없이 구축할 템플릿:

- azure_withoutDiagnostic_ftdv_nlb_cluster_parameters.json - 진단 인터페이스가 없는 NLB 구축이 있는 Threat Defense Virtual 클러스터에 대한 매개변수를 입력하는 템플릿입니다.

- azure_withoutDiagnostic_ftdv_nlb_cluster.json - 진단 인터페이스 없이 NLB가 있는 Threat Defense Virtual 클러스터를 구축하기 위한 템플릿입니다.

사전 요구 사항

- 클러스터가 Management Center에 자동 등록되도록 허용하려면 Management Center에서 네트워크 관리자 및 유지 보수 사용자 권한을 가진 사용자를 생성합니다. 이러한 권한이 있는 사용자는 REST API를 사용할 수 있습니다. Cisco Secure Firewall Management Center 관리 가이드를 참고하십시오.
- 템플릿 구축 중에 지정할 정책의 이름과 일치하는 액세스 정책을 Management Center에 추가합니다.
- Management Center Virtual에 적절한 라이선스가 부여되었는지 확인합니다.
- 클러스터가 Management Center Virtual에 추가된 후:
  1. Management Center에서 상태 확인 포트 번호로 플랫폼 설정을 구성합니다. 이 구성에 대한 자세한 내용은 플랫폼 설정을 참고하십시오.
  2. 외부 및 내부 인터페이스에서 발생하는 트래픽에 대한 고정 경로를 생성합니다. 고정 경로 생성에 대한 자세한 내용은 고정 경로 추가를 참고하십시오.

     외부 인터페이스에 대한 샘플 고정 경로 구성:

     ```
     Network: any-ipv4
     Interface: outside
     Leaked from Virtual Router: Global
     Gateway: ftdv-cluster-outside
     Tunneled: false
     Metric: 10
     ```

     참고 *ftdv-cluster-outside*는 외부 서브넷의 게이트웨이 IP 주소입니다.

     내부 인터페이스에 대한 샘플 고정 경로 구성:

     ```
     Network: any-ipv4
     Interface: inside
     Leaked from Virtual Router: Global
     Gateway: ftdv-cluster-inside-gw
     Tunneled: false
     Metric: 11
     ```

     참고 *ftdv-cluster-inside-gw*는 내부 서브넷의 게이트웨이 IP 주소입니다.

  3. 데이터 트래픽에 대한 NAT 규칙을 구성합니다. NAT 규칙 구성에 대한 자세한 내용은 네트워크 주소 변환을 참고하십시오.

Azure Resource Manager 템플릿을 통해 NLB를 사용하여 Azure에서 클러스터 구축

사용자 지정된 ARM(Azure Resource Manager) 템플릿을 사용하여 Azure NLB용 클러스터를 구축합니다.

프로시저

단계 **1** 템플릿을 준비합니다.

a) 로컬 폴더에 github 리포지토리를 복제합니다. https://github.com/CiscoDevNet/cisco-ftdv/tree/master/cluster/azure을 참조하십시오.

b) 필수 매개변수를 사용하여 *azure_ftdv_nlb_cluster.json* 및 *azure_ftdv_nlb_cluster_parameters.json*을 수정합니다.

진단 인터페이스 없이 클러스터를 구축하기 위한 필수 매개변수를 사용하여withoutDiagnostic 템플릿, *azure_withoutDiagnostic_ftdv_nlb_cluster_parameters.json* 및 *azure_withoutDiagnostic_ftdv_nlb_cluster.json*을 수정합니다.

단계 **2** Azure 포털: https://portal.azure.com에 로그인합니다.

단계 **3** 리소스 그룹을 생성합니다.

a) **Basics(**기본 사항**)** 탭의 드롭다운 목록에서 **Subscription(**구독**)** 및 **Resource Group(**리소스 그룹**)**을 선택합니다.

b) 필요한 **Region(**지역**)**을 선택합니다.

단계 **4** 5개의 서브넷(관리, 진단, 외부 및 CCL(Cluster Control Link))이 있는 가상 네트워크를 생성합니다.

Secure Firewall 버전 7.4.1부터 진단 인터페이스 없이 클러스터를 구축할 수 있습니다. Outside, Inside, Management 및 Cluster Control Link 인터페이스만 있는 클러스터를 구축하려면 withDiagnostic 템플릿 - azure_withoutDiagnostic_ftdv_nlb_cluster_parameters.json 및 azure_withoutDiagnostic_ftdv_nlb_cluster.json 파일을 사용합니다.

a) 가상 네트워크를 생성합니다.

1. **Basics(**기본 사항**)** 탭의 드롭다운 목록에서 **Subscription(**구독**)** 및 **Resource Group(**리소스 그룹**)**을 선택합니다.
2. b) 필요한 **Region(**지역**)**을 선택합니다. **Next: IP addresses(**다음**: IP** 주소**)**를 선택합니다.

b) 서브넷을 추가합니다.

IP Addresses(IP 주소**)** 탭에서 **Add subnet(**서브넷 추가**)**을 클릭하고 서브넷(관리, 진단, 내부, 외부 및 클러스터 제어 링크)을 추가합니다.

진단 인터페이스 없이 Threat Defense Virtual 7.4.1 클러스터를 구축하는 경우 진단 서브넷 생성을 건너뛰어야 합니다.

단계 **5** 사용자 지정 템플릿을 구축합니다.

a) **Create(**생성**)** > **Template deployment(**템플릿 구축**)(**사용자 지정 템플릿을 사용하여 구축**)**을 클릭합니다.

b) **Build your own template in the editor(**편집기에서 자체 템플릿 구축**)**를 클릭합니다.
c) 진단 인터페이스 구축 없이를 선택한 경우 **Load File(**파일 로드**)**을 클릭하고 **azure_ftdv_nlb_cluster.json** 또는 **azure_withoutDiagnostic_ftdv_nlb_cluster.json**을 업로드합니다.
d) **Save(**저장**)**를 클릭합니다.

단계 **6** 인스턴스 세부 정보를 구성합니다.

a) 필요한 값을 입력하고 **Review + create(**검토 + 생성**)**를 클릭합니다.

참고 클러스터 제어 링크 시작 및 종료 주소의 경우 필요한 만큼만 주소를 지정합니다(최대 16개). 범위가 클수록 성능에 영향을 줄 수 있습니다.

b) 검증이 통과되면 **Create(**생성**)**를 클릭합니다.

단계 **7** 인스턴스를 실행한 후 노드 중 하나에 로그인하고 **show cluster info** 명령을 사용하여 클러스터 구축을 확인합니다.

그림 ***14: show cluster info***

```
> show cluster info
Cluster gwlb-cluster-template-with-AN: On
    Interface mode: individual
Cluster Member Limit : 16
    This is "12" in state CONTROL_NODE
        ID        : 0
        Version   : 99.19(1)180
        Serial No.: 9AKGFV8VH4G
        CCL IP    : 10.1.1.12
        CCL MAC   : 000d.3a55.5470
        Module    : NGFWv
        Resource  : 8 cores / 28160 MB RAM
        Last join : 11:13:24 UTC Sep 5 2022
        Last leave: N/A
```

단계 **8** Azure 포털에서 함수 앱을 클릭하여 클러스터를 management center에 등록합니다.

참고 Function 앱을 사용하지 않으려면 **Add(**추가**)** > **Device(**디바이스**)**(**Add(**추가**)** > **Cluster(**클러스터**)**가 아님)를 사용하여 Management Center에 직접 제어 노드를 등록할 수도 있습니다. 나머지 클러스터 노드는 자동으로 등록됩니다.

단계 **9** **Deployment Center(**구축 센터**)** > **FTPS credentials(FTPS** 자격 증명**)** > **User scope(**사용자 범위**)** > **Configure Username and Password(**사용자 이름 및 비밀번호 구성**)**를 클릭한 다음 **Save(**저장**)**를 클릭합니다.

단계 **10** 로컬 터미널에서 다음 **curl** 명령을 실행하여 Cluster_Function.zip 파일을 Function 앱에 업로드합니다.

curl -X POST -u *username* **--data-binary @"Cluster_Function.zip" https://** *Function_App_Name***.scm.azurewebsites.net/api/zipdeploy**

참고 **curl** 명령 실행을 완료하는 데 몇 분(2~3분)이 걸릴 수 있습니다.

함수가 함수 앱에 업로드됩니다. 함수가 시작되고 스토리지 계정의 outqueue에서 로그를 볼 수 있습니다. Management Center에 대한 디바이스 등록이 시작됩니다.

Azure에서 수동으로 클러스터 구축

클러스터를 수동으로 구축하려면 day0 구성을 준비하고 각 노드를 구축한 다음 management center에 제어 노드를 추가합니다.

Azure용 **Day0** 구성 생성

고정 구성 또는 맞춤형 구성을 사용할 수 있습니다.

Azure용 고정 구성을 사용하여 **Day0** 구성 생성

고정 구성은 클러스터 부트스트랩 구성을 자동으로 생성합니다.

```
{
 "AdminPassword": "password",
 "FirewallMode": "Routed",
 "ManageLocally": "No",
 "Diagnostic": "OFF",    //For deployment of version 7.4.1 and later without Diagnostics
template, set this parameter to OFF.
. "FmcIp": "<FMC_IP>",
 "FmcRegKey": "<REGISTRATION_KEY>",
 "FmcNatId": "<NAT_ID>",
 "Cluster": {
  "CclSubnetRange": "ip_address_start ip_address_end",
  "ClusterGroupName": "cluster_name",
  "HealthProbePort": "port_number",
  "GatewayLoadBalancerIP": "ip_address",
  "EncapsulationType": "vxlan",
  "InternalPort": "internal_port_number",
  "ExternalPort": "external_port_number",
  "InternalSegId": "internal_segment_id",
  "ExternalSegId": "external_segment_id"
 }
}
```

예

Day 0 구성 샘플이 아래에 나와 있습니다.

```
{
 "AdminPassword": "password",
 "FirewallMode": "routed",
 "ManageLocally": "No",
 "Diagnostic": "OFF",    //For deployment of version 7.4.1 and later without Diagnostics
template, set this parameter to OFF.
 "FmcIp":"<FMC_IP>",
 "FmcRegKey":"<REGISTRATION_KEY>",
 "FmcNatId":"<NAT_ID>",
 "Cluster": {
  "CclSubnetRange": "10.45.3.4 10.45.3.30",    //mandatory user input
  "ClusterGroupName": "ngfwv-cluster",         //mandatory user input
  "HealthProbePort": "7777",                   //mandatory user input
  "GatewayLoadBalanceIP": "10.45.2.4",         //mandatory user input
  "EncapsulationType": "vxlan",
  "InternalPort": "2000",
  "ExternalPort": "2001",
  "InternalSegId": "800",
  "ExternalSegId": "801"
```

```
  }
}
```

참고 위의 설정을 복사하여 붙여넣는 경우에는 //필수 사용자 입력을 구성에서 제거해야 합니다.

Azure 상태 확인 설정의 경우 여기에서 설정한 **HealthProbePort**를 지정해야 합니다.

CclSubnetRange변수에 xxx4에서 시작하는 IP 주소 범위를 지정합니다. 클러스터링에 16개 이상의 사용 가능한 IP 주소가 있는지 확인합니다. 시작 및 종료 IP 주소의 몇 가지 예가 아래에 나와 있습니다.

표 ***4:*** 시작 및 종료 ***IP*** 주소의 예

CIDR	시작 **IP** 주소	종료 **IP** 주소
10.1.1.0/27	10.1.1.4	10.1.1.30
10.1.1.32/27	10.1.1.36	10.1.1.62
10.1.1.64/27	10.1.1.68	10.1.1.94
10.1.1.96/27	10.1.1.100	10.1.1.126
10.1.1.128/27	10.1.1.132	10.1.1.158
10.1.1.160/27	10.1.1.164	10.1.1.190
10.1.1.192/27	10.1.1.196	10.1.1.222
10.1.1.224/27	10.1.1.228	10.1.1.254

Azure용 사용자 지정 구성을 사용하여 **Day0** 구성 생성

명령을 사용하여 전체 클러스터 부트스트랩 구성을 입력할 수 있습니다.

```
{
 "AdminPassword": "password",
 "FirewallMode": "Routed",
 "ManageLocally": "No",
 "Diagnostic": "OFF",    //For deployment of version 7.4.1 and later without Diagnostics
template, set this parameter to OFF.
 "FmcIp": "<FMC_IP>",
 "FmcRegKey": "<REGISTRATION_KEY>",
 "FmcNatId": "<NAT_ID>",
 "Cluster": {
  "CclSubnetRange": "ip_address_start ip_address_end",
  "ClusterGroupName": "cluster_name",
  "HealthProbePort": "port_number",
  "GatewayLoadBalancerIP": "ip_address",
  "EncapsulationType": "vxlan",
  "InternalPort": "internal_port_number",
  "ExternalPort": "external_port_number",
  "InternalSegId": "internal_segment_id",
  "ExternalSegId": "external_segment_id"
 }
}
```

예

버전 **7.4** 이상용 Day 0 구성 샘플이 아래에 나와 있습니다.

```
{
 "AdminPassword": "Sup3rnatural",
 "Hostname": "clusterftdv",
 "FirewallMode": "routed",
 "ManageLocally": "No",
 "Diagnostic": "OFF",    //For deployment of version 7.4.1 and later without Diagnostics
template, set this parameter to OFF.
 "FmcIp": "<FMC_IP>",
 "FmcRegKey": "<REGISTRATION_KEY>",
 "FmcNatId": "<NAT_ID>",
 "run_config": [
  "cluster interface-mode individual force",
  "policy-map global_policy",
  "class inspection_default",
  "no inspect h323 h225",
  "no inspect h323 ras",
  "no inspect rtsp",
  "no inspect skinny",
  "interface Management0/0",
  "management-only",
  "nameif management",
  "security-level 0",
  "ip address dhcp",
  "interface GigabitEthernet0/0",
  "no shutdown",
  "nameif vxlan_tunnel",
  "security-level 0",
  "ip address dhcp",
  "interface GigabitEthernet0/1",
  "no shutdown",
  "nve-only cluster",
  "nameif ccl_link",
  "security-level 0",
  "ip address dhcp",
  "interface vni1",
  "description Clustering Interface",
  "segment-id 1",
  "vtep-nve 1",
  "interface vni2",
  "proxy paired",
  "nameif GWLB-backend-pool",
  "internal-segment-id 800",
  "external-segment-id 801",
  "internal-port 2000",
  "external-port 2001",
  "security-level 0",
  "vtep-nve 2",
  "object network ccl#link",
  "range 10.45.3.4 10.45.3.30",                          //mandatory user input
  "object-group network cluster#group",
  "network-object object ccl#link",
  "nve 1 ",
  "encapsulation vxlan",
  "source-interface ccl_link",
  "peer-group cluster#group",
  "nve 2 ",
  "encapsulation vxlan",
  "source-interface vxlan_tunnel",
  "peer ip <GatewayLoadbalancerIP>",
```

```
  "cluster group ftdv-cluster",                          //mandatory user input
  "local-unit 1",
  "cluster-interface vni1 ip 1.1.1.1 255.255.255.0",
  "priority 1",
  "enable",
  "mtu vxlan_tunnel 1454",
  "mtu ccl_link 1454"
 ]
}
```

버전 **7.3** 이하용 Day 0 구성 샘플이 아래에 나와 있습니다.

```
{
 "AdminPassword": "Sup3rnatural",
 "Hostname": "clusterftdv",
 "FirewallMode": "routed",
 "ManageLocally": "No",
 "FmcIp": "<FMC_IP>",
 "FmcRegKey": "<REGISTRATION_KEY>",
 "FmcNatId": "<NAT_ID>",
 "run_config": [
  "cluster interface-mode individual force",
  "policy-map global_policy",
  "class inspection_default",
  "no inspect h323 h225",
  "no inspect h323 ras",
  "no inspect rtsp",
  "no inspect skinny",
  "interface Management0/0",
  "management-only",
  "nameif management",
  "security-level 0",
  "ip address dhcp",
  "interface GigabitEthernet0/0",
  "no shutdown",
  "nameif vxlan_tunnel",
  "security-level 0",
  "ip address dhcp",
  "interface GigabitEthernet0/1",
  "no shutdown",
  "nve-only cluster",
  "nameif ccl_link",
  "security-level 0",
  "ip address dhcp",
  "interface vni1",
  "description Clustering Interface",
  "segment-id 1",
  "vtep-nve 1",
  "interface vni2",
  "proxy paired",
  "nameif GWLB-backend-pool",
  "internal-segment-id 800",
  "external-segment-id 801",
  "internal-port 2000",
  "external-port 2001",
  "security-level 0",
  "vtep-nve 2",
  "object network ccl#link",
  "range 10.45.3.4 10.45.3.30",                          //mandatory user input
  "object-group network cluster#group",
  "network-object object ccl#link",
  "nve 1 ",
  "encapsulation vxlan",
  "source-interface ccl_link",
```

```
  "peer-group cluster#group",
  "nve 2 ",
  "encapsulation vxlan",
  "source-interface vxlan_tunnel",
  "peer ip <GatewayLoadbalancerIP>",
  "cluster group ftdv-cluster",                          //mandatory user input
  "local-unit 1",
  "cluster-interface vni1 ip 1.1.1.1 255.255.255.0",
  "priority 1",
  "enable",
  "mtu vxlan_tunnel 1454",
  "mtu ccl_link 1554"
 ]
}
```

참고 위의 설정을 복사하여 붙여넣는 경우에는 //필수 사용자 입력을 구성에서 제거해야 합니다.

수동으로 클러스터 노드 구축 - **GWLB** 기반 구축

클러스터를 구성하도록 클러스터 노드를 구축합니다.

프로시저

단계 **1** **az vmss create** CLI를 사용하여 인스턴스 수가 0인 마켓플레이스 이미지에서 VMSS(Virtual Machine Scale Set)를 생성합니다.

az vmss create --resource-group <*ResourceGroupName*> **--name** <*VMSSName*> **--vm-sku** <*InstanceSize*> **--image** <*FTDvImage*> **--instance-count 0 --admin-username** <*AdminUserName*> **--admin-password** <*AdminPassword*> **--plan-name** <*ftdv-azure-byol/ftdv-azure-payg*> **--plan-publisher cisco --plan-product cisco-ftdv --plan-promotion-code** <*ftdv-azure-byol/ftdv-azure-payg*> **--vnet-name** <*VirtualNetworkName*> **--subnet** <*MgmtSubnetName*>

단계 **2** 진단, 데이터 및 클러스터 제어 링크의 3개 인터페이스를 연결합니다.

단계 **3** 생성한 VMSS(Virtual Machine Scale Set)로 이동하고 다음 단계를 수행합니다.

a) **Operating system(**운영 체제**)** 섹션 아래의 **customData** 필드에 Day 0 구성을 추가합니다.

b) **Save(**저장**)**를 클릭합니다.

c) **Scaling(**확장**)** 섹션에서 필요한 클러스터 노드로 인스턴스 수를 업데이트합니다. 최소 1에서 최대 16까지 인스턴스 수 범위를 설정할 수 있습니다.

단계 **4** Azure 게이트웨이 로드 밸런서를 구성합니다. 자세한 내용은 Azure 게이트웨이 로드 밸런서를 이용한 Auto Scale 사용 사례를 참조하십시오.

단계 **5** management center에 제어 노드를 추가합니다. Management Center에 클러스터 추가(수동 구축), 54 페이지의 내용을 참조하십시오.

수동으로 클러스터 노드 구축 - **NLB** 기반 구축

클러스터를 구성하도록 클러스터 노드를 구축합니다.

프로시저

단계 **1** **az vmss create** CLI를 사용하여 인스턴스 수가 0인 마켓플레이스 이미지에서 VMSS(Virtual Machine Scale Set)를 생성합니다.

az vmss create --resource-group <*ResourceGroupName*> **--name** <*VMSSName*> **--vm-sku** <*InstanceSize*> **--image** <*FTDvImage*> **--instance-count 0 --admin-username** <*AdminUserName*> **--admin-password** <*AdminPassword*> **--plan-name** <*ftdv-azure-byol/ftdv-azure-payg*> **--plan-publisher cisco --plan-product cisco-ftdv --plan-promotion-code** <*ftdv-azure-byol/ftdv-azure-payg*> **--vnet-name** <*VirtualNetworkName*> **--subnet** <*MgmtSubnetName*>

단계 **2** 진단, 내부, 외부 및 클러스터 제어 링크의 4개 인터페이스를 연결합니다.

단계 **3** 생성한 VMSS(Virtual Machine Scale Set)로 이동하고 다음을 수행합니다.

a) **Operating system(**운영 체제**)** 섹션 아래의 **customData** 필드에 Day 0 구성을 추가합니다.

b) **Save(**저장**)**를 클릭합니다.

c) **Scaling(**확장**)** 섹션에서 필요한 클러스터 노드로 인스턴스 수를 업데이트합니다. 최소 1에서 최대 16까지 인스턴스 수 범위를 설정할 수 있습니다.

단계 **4** Management Center에 제어 노드를 추가합니다. Management Center에 클러스터 추가(수동 구축), 54 페이지의 내용을 참조하십시오.

Azure에서 클러스터 구축 문제 해결

- 문제: 트래픽 흐름 없음

  문제 해결:

  - GWLB를 사용하여 구축된 Threat Defense Virtual 인스턴스의 상태 프로브 상태가 정상인지 확인합니다.
  - Threat Defense Virtual 인스턴스의 상태 프로브 상태가 비정상인 경우
    - Management Center Virtual에 고정 경로가 구성되어 있는지 확인합니다.
    - 기본 게이트웨이가 데이터 서브넷의 게이트웨이 IP인지 확인합니다.
    - Threat Defense Virtual 인스턴스가 상태 프로브 트래픽을 수신하고 있는지 확인합니다.
    - Management Center Virtual에 구성된 액세스 목록이 상태 프로브 트래픽을 허용하는지 확인합니다.

- 문제점: 클러스터가 형성되지 않았음

  문제 해결:

  - nve 전용 클러스터 인터페이스의 IP 주소를 확인합니다. 다른 노드의 nve 전용 클러스터 인터페이스를 ping할 수 있는지 확인합니다.
  - nve 전용 클러스터 인터페이스의 IP 주소가 개체 그룹의 일부인지 확인합니다.

- NVE 인터페이스가 개체 그룹으로 구성되어 있는지 확인합니다.
- 클러스터 그룹의 클러스터 인터페이스에 올바른 VNI 인터페이스가 있는지 확인합니다. 이 VNI 인터페이스에는 해당 개체 그룹이 있는 NVE가 있습니다.
- 노드가 서로 ping 가능한지 확인합니다. 각 노드에는 자체 클러스터 인터페이스 IP가 있으므로 서로 ping할 수 있어야 합니다.
- 템플릿 구축 중에 언급된 CCL 서브넷의 시작 및 종료 주소가 올바른지 확인하십시오. 시작 주소는 서브넷에서 사용 가능한 첫 번째 IP 주소로 시작해야 합니다. 예를 들어 서브넷이 192.168.1.0/24인 경우. 시작 주소는 192.168.1.4여야 합니다(시작 시 3개의 IP 주소는 Azure에서 예약됨).
- Management Center Virtual에 유효한 라이선스가 있는지 확인합니다.

- 문제점: 동일한 리소스 그룹에서 리소스를 다시 구축하는 동안 역할 관련 오류가 발생했습니다.

  문제 해결: 터미널에서 다음 명령을 사용하여 아래에 지정된 역할을 제거합니다.

  오류 메시지:

```
"error": {
"code": "RoleAssignmentUpdateNotPermitted",
"message": "Tenant ID, application ID, principal ID, and scope are not allowed to be
updated."}
```

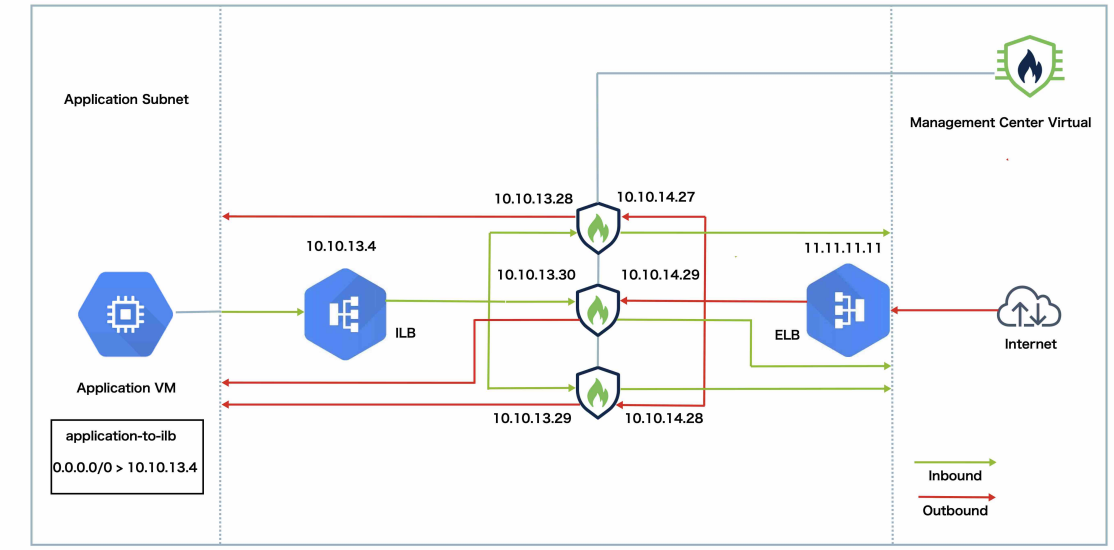
  - **az role assignment delete --resource-group** <리소스 그룹 이름> **--role "Storage Queue Data Contributor"**
  - **az role assignment delete --resource-group** <리소스 그룹 이름> **--role "Contributor"**

GCP에서 클러스터 구축

GCP에서 클러스터를 구축하려면 수동으로 구축하거나 인스턴스 템플릿을 사용하여 인스턴스 그룹을 구축할 수 있습니다. 기본 GCP 로드 밸런서 또는 Cisco Cloud Services Router와 같은 기본이 아닌 로드 밸런서와 함께 클러스터를 사용할 수 있습니다.

참고 아웃바운드 트래픽은 인터페이스 NAT를 필요로 하며 64K 연결로 제한됩니다.

샘플 토폴로지



이 토폴로지에는 인바운드 및 아웃바운드 트래픽 플로우가 모두 나와 있습니다. Threat Defense Virtual 클러스터는 내부 로드 밸런서와 외부 로드 밸런서 사이에 위치합니다. Management Center Virtual 인스턴스는 클러스터를 관리하는 데 사용됩니다.

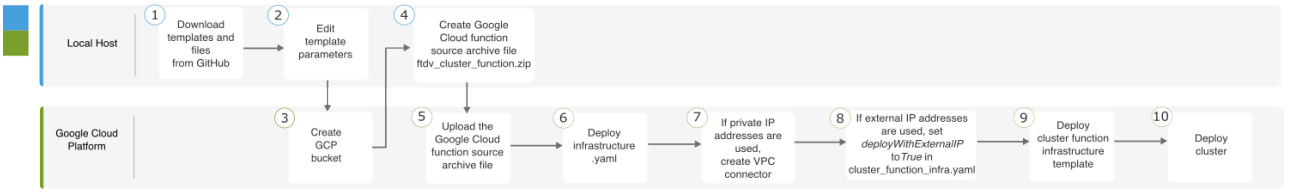
인터넷의 인바운드 트래픽은 외부 로드 밸런서로 이동된 다음 Threat Defense Virtual 클러스터로 전송됩니다. 트래픽은 클러스터의 Threat Defense Virtual 인스턴스에서 검사된 후 애플리케이션 VM으로 전달됩니다.

애플리케이션 VM의 아웃바운드 트래픽은 내부 로드 밸런서로 전송됩니다. 그런 다음 트래픽은 Threat Defense Virtual 클러스터로 전달된 후 인터넷으로 전송됩니다.

GCP에서 **Threat Defense Virtual** 클러스터를 구축하기 위한 **End-to-End** 프로세스

템플릿 기반 구축

다음 순서도에는 GCP에서 Threat Defense Virtual 클러스터를 템플릿 기반으로 구축하는 워크플로우가 나와 있습니다.

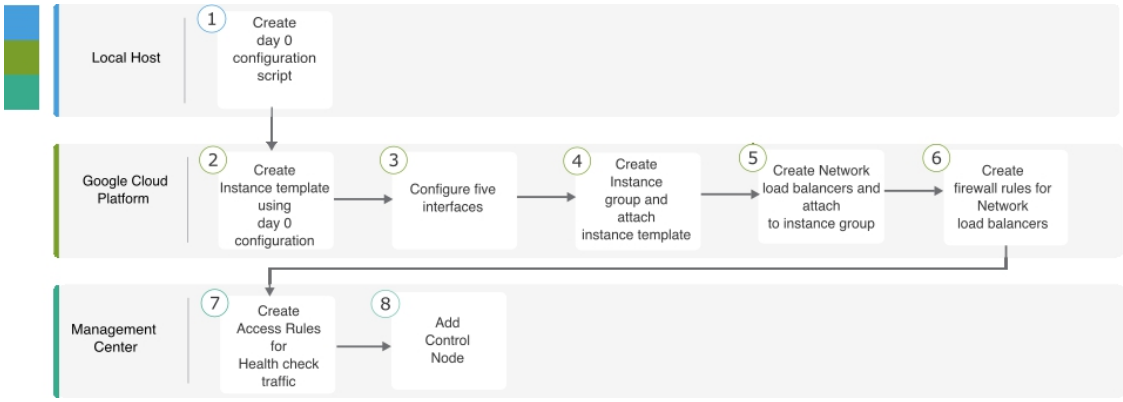


	업무 환경	단계
1	로컬 호스트	GitHub에서 템플릿 및 파일을 다운로드합니다.

	업무 환경	단계
2	로컬 호스트	템플릿 매개변수를 편집합니다.
3	Google Cloud Platform	GCP 버킷을 생성합니다.
4	로컬 호스트	Google Cloud 함수 소스 아카이브 파일 *ftdv_cluster_function.zip*을 생성합니다.
5	Google Cloud Platform	Google 함수 소스 아카이브 파일을 업로드합니다.
6	Google Cloud Platform	infrastructure.yaml을 구축합니다.
7	Google Cloud Platform	사설 IP 주소를 사용하는 경우 VPC 커넥터를 생성합니다.
8	Google Cloud Platform	외부 IP 주소를 사용하는 경우에는 *cluster_function_infra.yaml*에서 *deployWithExternalIP*를 *True*로 설정합니다.
9	Google Cloud Platform	클러스터 기능 인프라 템플릿을 구축합니다.
10	Google Cloud Platform	클러스터를 구축합니다.

수동 구축

다음 순서도에는 GCP에서 Threat Defense Virtual 클러스터를 수동으로 구축하는 워크플로우가 나와 있습니다.



	업무 환경	단계
1	로컬 호스트	GCP에 대한 Day0 구성 생성
2	Google Cloud Platform	Day 0 구성을 사용하여 인스턴스 템플릿을 생성합니다.

	업무 환경	단계
3	Google Cloud Platform	인터페이스를 구성합니다.
4	Google Cloud Platform	인스턴스 그룹을 생성하고 인스턴스 템플릿을 연결합니다.
5	Google Cloud Platform	NLB를 생성하고 인스턴스 그룹에 연결합니다.
6	Google Cloud Platform	NLB에 대한 방화벽 규칙을 생성합니다.
7	Management Center	상태 확인 트래픽에 대한 액세스 규칙을 생성합니다.
8	Management Center	제어 노드를 추가합니다.

템플릿

아래에 제공된 템플릿은 GitHub에서 사용할 수 있습니다. 매개변수 값은 템플릿에 지정된 매개변수 이름 및 값을 통해 이해할 수 있습니다.

- 이스트-웨스트 트래픽에 대한 클러스터 구축 템플릿 - *deploy_ngfw_cluster_yaml*
- 노스-사우스 트래픽에 대한 클러스터 구축 템플릿 - deploy_ngfw_cluster.yaml

인스턴스 템플릿을 사용하여 **GCP**에서 인스턴스 그룹 구축

인스턴스 템플릿을 사용하여 GCP에서 인스턴스 그룹을 구축합니다.

시작하기 전에

- 구축에 Google Cloud Shell을 사용합니다. 또는 모든 macOS/Linux/Windows 시스템에서 Google SDK를 사용할 수 있습니다.
- 클러스터가 Management Center에 자동 등록되도록 하려면 REST API를 사용할 수 있는 Management Center에 대한 관리 권한이 있는 사용자를 생성해야 합니다. Cisco Secure Firewall Management Center 관리 가이드를 참조하십시오.
- *cluster_function_infra.yaml*에서 지정한 정책의 이름과 일치하는 액세스 정책을 Management Center에 추가합니다.

프로시저

단계 **1** GitHub의 템플릿을 로컬 폴더에 다운로드합니다.

단계 **2** 필수 *resourceNamePrefix* 매개변수(예: ngfwvcls) 및 기타 필수 사용자 입력을 사용하여 **infrastructure.yaml** , **cluster_function_infra.yaml** 및 **deploy_ngfw_cluster.yaml**을 편집합니다.

Secure Firewall 버전 7.4.1부터 진단 인터페이스 없이 클러스터를 구축할 수 있습니다. 외부, 내부, 관리 및 CCL 인터페이스만 있는 클러스터를 구축하려면, **infrastructure.yaml** 및 **deploy_ngfw_cluster.yaml** 파일에서 *withDiagnostic* 변수를 **False**로 설정합니다.

deploy_ngfw_cluster.yaml 파일은 GitHub의 **east-west** 및 **north-south** 폴더에 있습니다. 트래픽 플로우 요구 사항에 따라 적절한 템플릿을 다운로드합니다.

단계 **3** Google Cloud Shell을 사용해 버킷을 생성하여 Google 클라우드 기능 소스 아카이브 파일 *ftdv_cluster_funtion.zip*을 업로드합니다.

gsutil mb --pap enforced gs://*resourceNamePrefix***-ftdv-cluster-bucket/**

여기의 *resourceNamePrefix* 변수가 **cluster_function_infra.yaml**에 지정한 *resourceNamePrefix* 변수와 일치하는지 확인합니다.

단계 **4** 클러스터 인프라용 아카이브 파일을 생성합니다.

예제**:**

```
zip -j ftdv_cluster_function.zip ./cluster-function/*
```

단계 **5** 이전에 생성한 Google 소스 아카이브를 업로드합니다.

gsutil cp ftdv_cluster_function.zip gs://*resourceNamePrefix***-ftdv-cluster-bucket/**

단계 **6** 클러스터용 인프라를 구축합니다.

gcloud deployment-manager deployments create *cluster_name* **--config infrastructure.yaml**

단계 **7** 사설 IP 주소를 사용하는 경우, 아래의 단계를 수행하십시오.

a) Threat Defense Virtual 관리 VPC로 Management Center Virtual을 실행하고 설정합니다.

b) VPC 커넥터를 생성하여 Google Cloud 기능을 Threat Defense Virtual 관리 VPC에 연결합니다.

gcloud compute networks vpc-access connectors create *vpc-connector-name* **--region us-central1 --subnet** *resourceNamePrefix***-ftdv-mgmt-subnet28**

단계 **8** Management Center가 Threat Defense Virtual에서 원격이고 Threat Defense Virtual에 외부 IP 주소가 필요한 경우 **cluster_function_infra.yaml**에서 **deployWithExternalIP**를 **True**로 설정해야 합니다.

단계 **9** 클러스터 기능 인프라를 구축합니다.

gcloud deployment-manager deployments create *cluster_name* **--config cluster_function_infra.yaml**

단계 **10** 클러스터를 구축합니다.

1. 노스-사우스 토폴로지 구축의 경우:

   gcloud deployment-manager deployments create *cluster_name* **--config north-south/deploy_ngfw_cluster.yaml**

2. 이스트-웨스트 토폴로지 구축의 경우:

   gcloud deployment-manager deployments create *cluster_name* **--config east-west/deploy_ngfw_cluster.yaml**

GCP에서 수동으로 클러스터 구축

클러스터를 수동으로 구축하려면 day0 구성을 준비하고 각 노드를 구축한 다음 management center에 제어 노드를 추가합니다.

GCP에 대한 Day0 구성 생성

고정 구성 또는 맞춤형 구성을 사용할 수 있습니다.

GCP에 대한 고정 구성으로 Day0 구성 생성

고정 구성은 클러스터 부트스트랩 구성을 자동으로 생성합니다.

```
{
    "AdminPassword": "password",
    "Hostname": "hostname",
    "FirewallMode": "Routed",
    "ManageLocally": "No",
                "Diagnostic": "OFF",       //Optional user input from version 7.4.1 - use
to deploy cluster without Diagnostic interface
    "Cluster": {
        "CclSubnetRange": "ip_address_start ip_address_end",
        "ClusterGroupName": "cluster_name"
    }
}
```

대표적인 예는 다음과 같습니다.

```
{
    "AdminPassword": "DeanW1nche$ter",
    "Hostname": "ciscoftdv",
    "FirewallMode": "Routed",
    "ManageLocally": "No",
    "Cluster": {
        "CclSubnetRange": "10.10.55.2 10.10.55.253",     //mandatory user input
        "ClusterGroupName": "ftdv-cluster"               //mandatory user input
    }
}
```

참고 위의 설정을 복사하여 붙여넣는 경우에는 //필수 사용자 입력을 구성에서 제거해야 합니다.

CclSubnetRange 변수에는 서브넷의 처음 2개 IP 주소와 마지막 2개 IP 주소를 사용할 수 없습니다. 자세한 내용은 IPv4 서브넷의 예약된 IP 주소를 참고하십시오. 클러스터링에 16개 이상의 사용 가능한 IP 주소가 있는지 확인합니다. 시작 및 종료 IP 주소의 몇 가지 예가 아래에 나와 있습니다.

표 *5:* 시작 및 종료 *IP* 주소의 예

CIDR	시작 IP 주소	종료 IP 주소
10.1.1.0/27	10.1.1.2	10.1.1.29
10.1.1.32/27	10.1.1.34	10.1.1.61
10.1.1.64/27	10.1.1.66	10.1.1.93

CIDR	시작 **IP** 주소	종료 **IP** 주소
10.1.1.96/27	10.1.1.98	10.1.1.125
10.1.1.128/27	10.1.1.130	10.1.1.157
10.1.1.160/27	10.1.1.162	10.1.1.189
10.1.1.192/27	10.1.1.194	10.1.1.221
10.1.1.224/27	10.1.1.226	10.1.1.253
10.1.1.0/24	10.1.1.2	10.1.1.253

GCP에 대한 사용자 지정 구성을 사용하여 **Day0** 구성 생성

명령을 사용하여 전체 클러스터 부트스트랩 구성을 입력할 수 있습니다.

```
{
    "AdminPassword": "password",
    "Hostname": "hostname",
    "FirewallMode": "Routed",
    "ManageLocally": "No",
    "run_config": [comma_separated_threat_defense_configuration]
    }
```

다음 예에서는 관리, 내부 및 외부 인터페이스가 있는 구성과 클러스터 제어 링크에 대한 VXLAN 인터페이스를 생성합니다. 굵게 표시된 값은 노드별로 고유해야 합니다.

```
{
 "AdminPassword": "W1nch3sterBr0s",
 "Hostname": "ftdv1",
 "FirewallMode": "Routed",
 "ManageLocally": "No",
 "run_config": [
  "cluster interface-mode individual force",
  "interface Management0/0",
  "management-only",
  "nameif management",
  "ip address dhcp",
  "interface GigabitEthernet0/0",
  "no shutdown",
  "nameif outside",
  "ip address dhcp",
  "interface GigabitEthernet0/1",
  "no shutdown",
  "nameif inside",
  "ip address dhcp",
  "interface GigabitEthernet0/2",
  "nve-only cluster",
  "nameif ccl_link",
  "ip address dhcp",
  "no shutdown",
  "interface vni1",
  "description Clustering Interface",
  "segment-id 1",
  "vtep-nve 1",
  "object network ccl#link",
  "range 10.1.90.2 10.1.90.17",
  "object-group network cluster#group",
  "network-object object ccl#link",
```

```
  "nve 1",
  "encapsulation vxlan",
  "source-interface ccl_link",
  "peer-group cluster#group",
  "cluster group ftdv-cluster",
  "local-unit 1",
  "cluster-interface vni1 ip 10.1.1.1 255.255.255.0",
  "priority 1",
  "enable",
  "mtu outside 1400",
  "mtu inside 1400"
 ]
}
```

참고 클러스터 제어 링크 네트워크 개체의 경우, 필요한 만큼의 주소만 지정합니다(최대 16개). 범위가 클수록 성능에 영향을 줄 수 있습니다.

수동으로 클러스터 노드 구축

클러스터를 구성하도록 클러스터 노드를 구축합니다. GCP에서 클러스터링하는 경우 4개의 vCPU 머신 유형을 사용할 수 없습니다. 4개의 vCPU 머신 유형은 4개의 인터페이스만 지원하며 5개가 필요합니다. 5가지 인터페이스를 지원하는 머신 유형(예: c2- Standard-8)을 사용합니다.

프로시저

단계 **1** 5가지 인터페이스(외부, 내부, 관리, 진단 및 클러스터 제어 링크)가 있는 Day 0 구성(**Metadata(**메타데이터**) > Startup Script(**시작 스크립트**)** 섹션)을 사용하여 인스턴스 템플릿을 생성합니다.

Cisco Secure Firewall Threat Defense Virtual 시작 가이드을 참조하십시오.

단계 **2** 인스턴스 그룹을 생성하고 인스턴스 템플릿을 연결합니다.

단계 **3** GCP 네트워크 로드 밸런서(내부 및 외부)를 생성하고 인스턴스 그룹을 연결합니다.

단계 **4** GCP 네트워크 로드 밸런서의 경우 Management Center의 보안 정책에서 상태 확인을 허용합니다. GCP 네트워크 로드 밸런서에 대한 상태 확인 허용, 52 페이지의 내용을 참조하십시오.

단계 **5** Management Center에 제어 노드를 추가합니다. Management Center에 클러스터 추가(수동 구축), 54 페이지의 내용을 참조하십시오.

GCP 네트워크 로드 밸런서에 대한 상태 확인 허용

Google Cloud는 백엔드가 트래픽에 응답하는지 확인하기 위한 상태 확인을 제공합니다.

네트워크 로드 밸런서에 대한 방화벽 규칙을 생성하려면 https://cloud.google.com/load-balancing/docs/health-checks의 내용을 참조하십시오. 그런 다음 management center에서 상태 확인 트래픽을 허용하는 액세스 규칙을 생성합니다. 필수 네트워크 범위는 https://cloud.google.com/load-balancing/docs/health-check-concepts의 내용을 참조하십시오.

또한 상태 확인 트래픽을 169.254.169.254의 Google 메타데이터 서버로 리디렉션하도록 동적 수동 NAT 규칙을 구성해야 합니다.

노스-사우스**(North-South) NAT** 규칙 샘플 구성

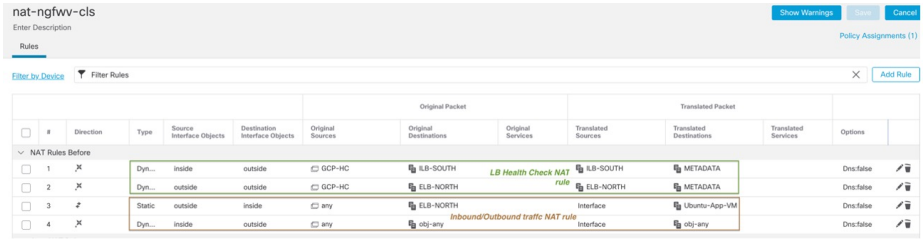
```
nat (inside,outside) source dynamic GCP-HC ILB-SOUTH destination static ILB-SOUTH METADATA
nat (outside,outside) source dynamic GCP-HC ELB-NORTH destination static ELB-NORTH METADATA

nat (outside,inside) source static any interface destination static ELB-NORTH Ubuntu-App-VM
nat (inside,outside) source dynamic any interface destination static obj-any obj-any

object network Metadata
 host 169.254.169.254

object network ILB-SOUTH
host <ILB_IP>
object network ELB-NORTH
host <ELB_IP>

object-group network GCP-HC
 network-object 35.191.0.0 255.255.0.0
 network-object 130.211.0.0 255.255.252.0
 network-object 209.85.204.0 255.255.252.0
 network-object 209.85.152.0 255.255.252.0
```



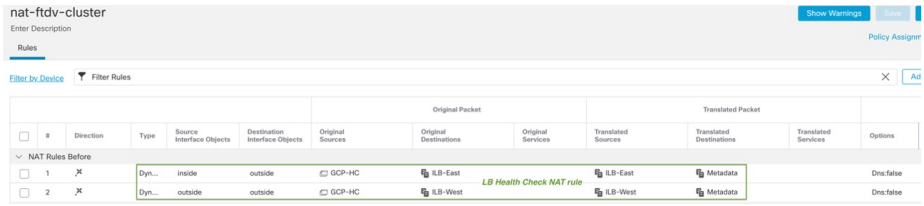
이스트-웨스트**(East-West) NAT** 규칙 샘플 구성

```
nat (inside,outside) source dynamic GCP-HC ILB-East destination static ILB-East Metadata
nat (outside,outside) source dynamic GCP-HC ILB-West destination static ILB-West Metadata

object network Metadata
 host 169.254.169.254

object network ILB-East
host <ILB_East_IP>
object network ILB-West
host <ILB_West_IP>

object-group network GCP-HC
 network-object 35.191.0.0 255.255.0.0
 network-object 130.211.0.0 255.255.252.0
 network-object 209.85.204.0 255.255.252.0
 network-object 209.85.152.0 255.255.252.0
```



Management Center에 클러스터 추가(수동 구축)

클러스터를 수동으로 구축한 경우 이 절차를 사용하여 management center에 클러스터를 추가합니다. 템플릿을 사용한 경우 클러스터가 management center에 자동 등록됩니다.

management center에 클러스터 유닛 중 하나를 새 장치로 추가합니다. management center는 다른 클러스터 멤버를 자동으로 감지합니다.

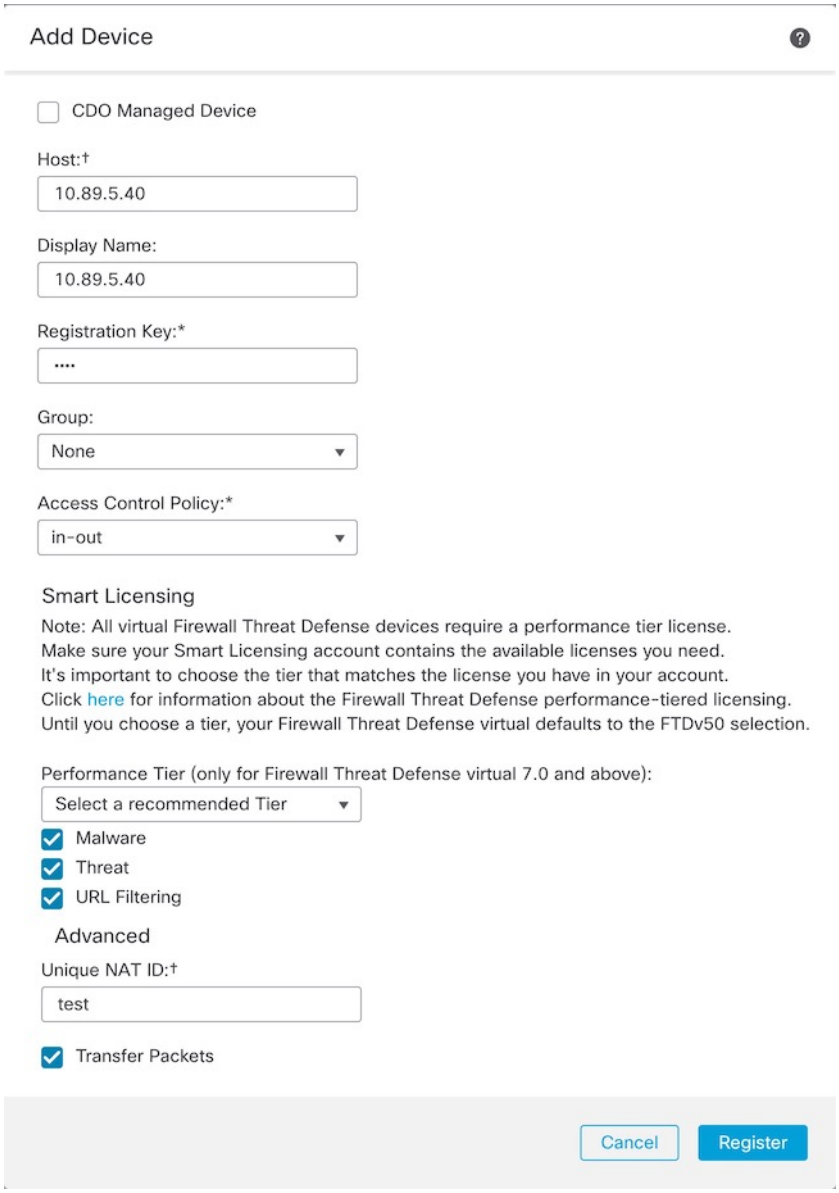
시작하기 전에

- management center에 추가하기 전에 모든 클러스터 유닛이 성공적으로 형성된 클러스터에 있어야 합니다. 제어 유닛을 확인하십시오. threat defense **show cluster info** 명령을 사용하십시오.

프로시저

단계 **1** management center에서 **Devices(**디바이스**)** > **Device Management(**디바이스 관리**)**를 선택하고 유닛의 관리 IP 주소를 사용해 제어 유닛을 추가하기 위해 **Add(**추가**)** > **Add Device(**디바이스 추가**)**를 선택합니다.

그림 ***15:*** 디바이스 추가



a) **Host(**호스트**)** 필드에 제어 유닛의 IP 주소나 호스트 이름을 입력합니다.

최적의 성능을 위해서는 제어 유닛을 추가하는 것이 좋습니다. 하지만 모든 클러스터 유닛을 추가할 수 있습니다.

디바이스 설정 중에 NAT ID를 사용한 경우 이 필드를 입력하지 않아도 됩니다.

b) management center에 표시할 제어 유닛의 이름을 표시 이름 필드에 입력합니다.

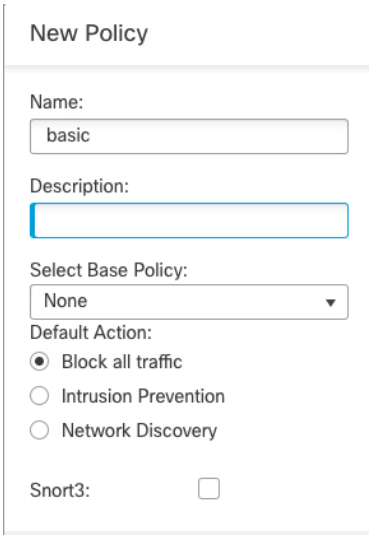
이 표시 이름은 클러스터용이 아닙니다. 추가하려는 제어 유닛에만 해당됩니다. 나중에 다른 클러스터 멤버의 이름과 클러스터 표시 이름을 변경할 수 있습니다.

c) **Registration Key(**등록 키**)** 필드에 디바이스 설치 중에 사용한 것과 동일한 등록 키를 입력합니다. 등록 키는 일회용 공유 암호입니다.

d) (선택 사항) 디바이스 그룹에 디바이스를 추가합니다.

e) 등록 시 디바이스를 구축하기 위해 초기 액세스 제어 정책을 선택하거나 새 정책을 생성합니다.

새 정책을 생성하는 경우 기본 정책만 생성합니다. 나중에 필요에 따라 정책을 사용자 정의할 수 있습니다.



f) 디바이스에 적용할 라이선스를 선택합니다.

g) 디바이스 설정 중 NAT ID를 사용하는 경우 고급 섹션을 확장하고 고유 **NAT ID** 필드에 동일한 NAT ID를 입력합니다.

h) 패킷 전송 체크 박스를 선택하여 디바이스가 management center에 패킷을 전송하도록 합니다.

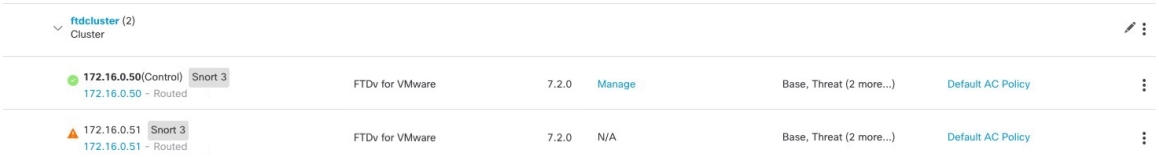
이 옵션은 기본적으로 활성화되어 있습니다. 이 옵션이 활성화되어 IPS 또는 Snort 같은 이벤트가 트리거되면 디바이스는 검사를 위해 이벤트 메타데이터 정보 및 패킷 데이터를 management center에 전송합니다. 이벤트를 비활성화하면 이벤트 정보는 management center에 전송되지만 패킷 데이터는 전송되지 않습니다.

i) **Register(**등록**)**를 클릭합니다.

management center은 제어 유닛을 식별해 등록하고 모든 데이터 유닛을 등록합니다. 제어 유닛이 성공적으로 등록되지 않는 경우 클러스터가 추가되지 않습니다. 클러스터가 없거나 다른 연결 문제로 등록이 실패할 수 있습니다. 이 경우 클러스터 유닛 추가를 다시 시도하시기를 권장합니다.

디바이스 > 디바이스 관리 페이지에 클러스터 이름이 표시됩니다. 클러스터 유닛을 보려면 클러스터를 확장합니다.

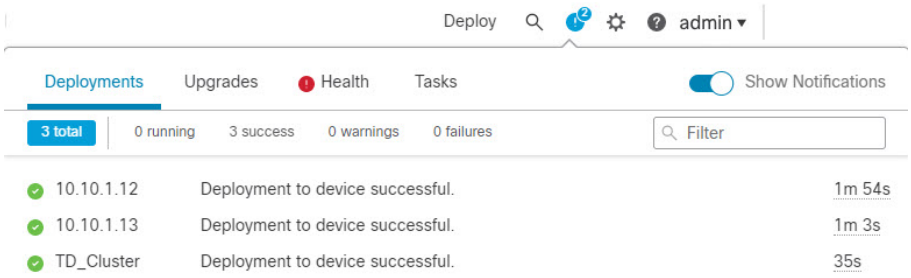
그림 ***16:*** 클러스터 관리



현재 등록되는 유닛에는 로딩 아이콘이 표시됩니다.

그림 ***17:*** 노드 등록

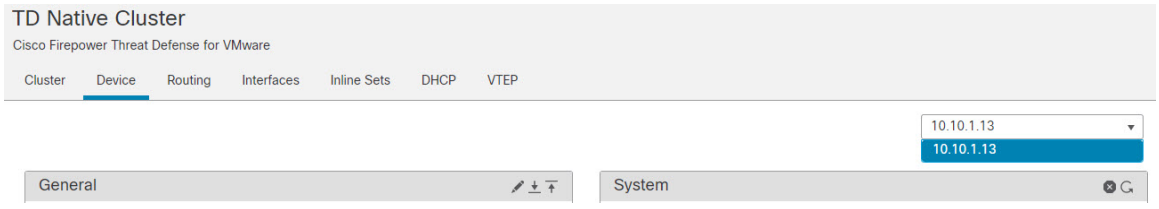
알림 아이콘을 클릭하고 작업을 선택하여 클러스터 유닛 등록을 모니터링할 수 있습니다. management center은 각 유닛이 등록될 때마다 클러스터 등록 작업을 업데이트합니다. 유닛 등록에 실패하는 경우 클러스터 노드 조정, 66 페이지의 내용을 참조하십시오.



단계 **2** 클러스터에 대해 편집 (✎)을 클릭하여 디바이스별 설정을 구성합니다.

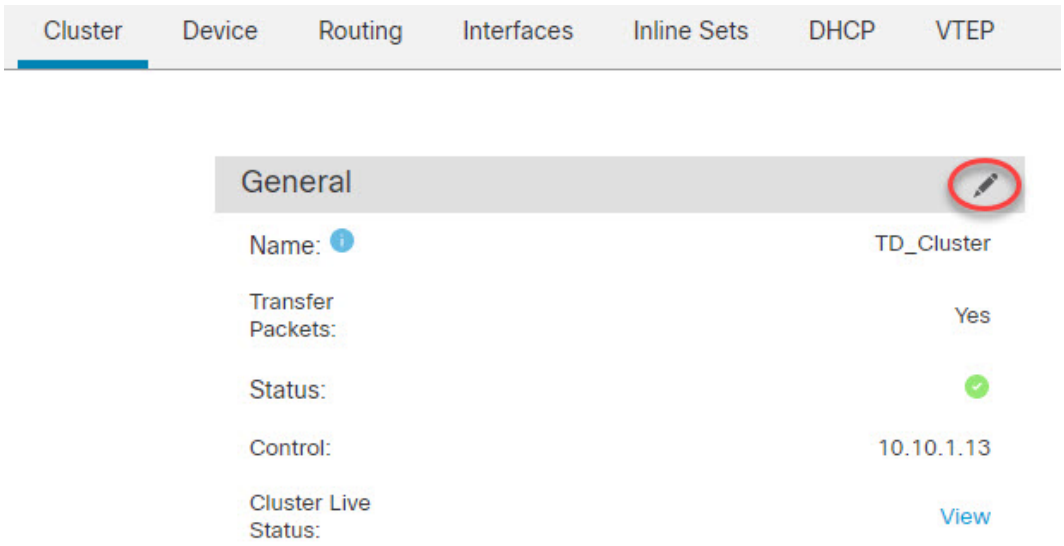
대부분의 구성은 클러스터의 노드가 아닌 클러스터 전체에 적용할 수 있습니다. 예를 들어 노드당 표시 이름을 변경할 수 있지만 전체 클러스터에 대해서만 인터페이스를 설정할 수 있습니다.

단계 **3** 디바이스 > 디바이스 관리 > 클러스터 화면에서 일반, 라이선스, 시스템및 상태 설정을 표시합니다.

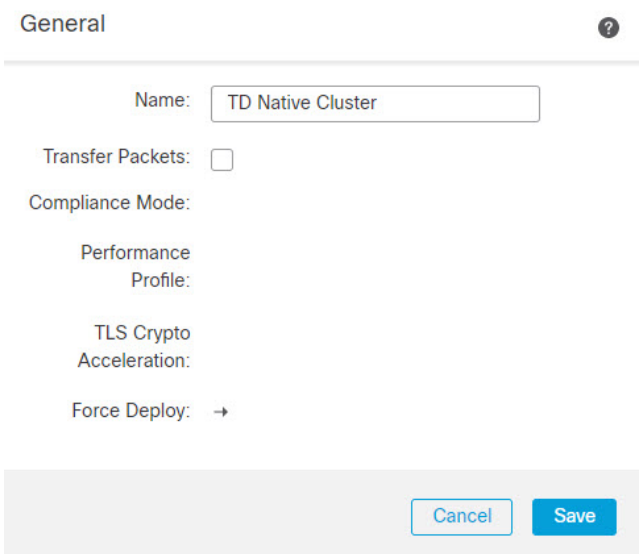


다음 클러스터별 항목을 참조하십시오.

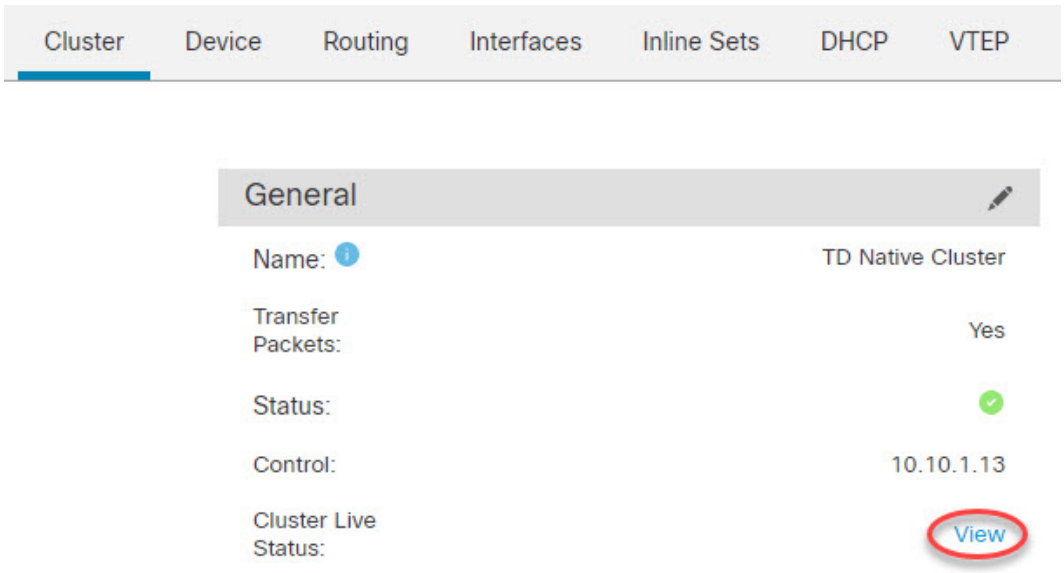
- **General (**일반**)> Name (**이름**)** - 편집 (✎)를 클릭하여 클러스터 표시 이름을 변경합니다.



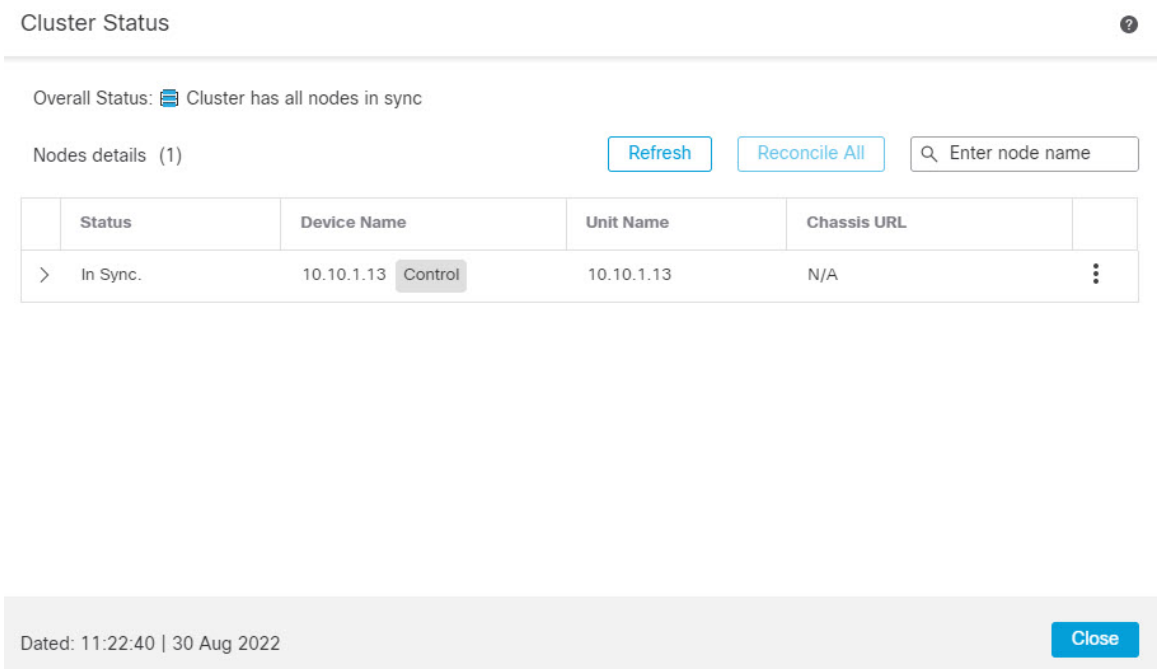
그런 다음 **Name(**이름**)** 필드를 설정합니다.



- **General(**일반**) > Cluster Live Status(**클러스터 라이브 상태**)**—**View(**보기**)** 링크를 클릭하여 **Cluster Status(**클러스터 상태**)** 대화 상자를 엽니다.

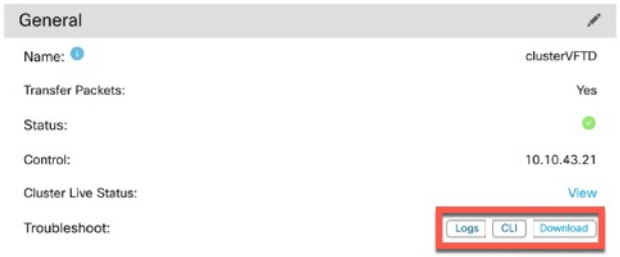


Cluster Status(클러스터 상태**)** 대화 상자에서 **Reconcile(**조정**)**을 클릭하면 데이터 유닛 등록을 다시 시도할 수도 있습니다.노드에서 클러스터 제어 링크를 ping할 수도 있습니다. 클러스터 제어 링크에서 ping 수행, 75 페이지의 내용을 참조하십시오.



- **General(**일반**) - Troubleshoot(**문제 해결**)** - 문제 해결 로그를 생성하고 다운로드할 수 있고 클러스터 CLI를 볼 수 있습니다. 클러스터 문제 해결, 74 페이지을 참조하십시오.

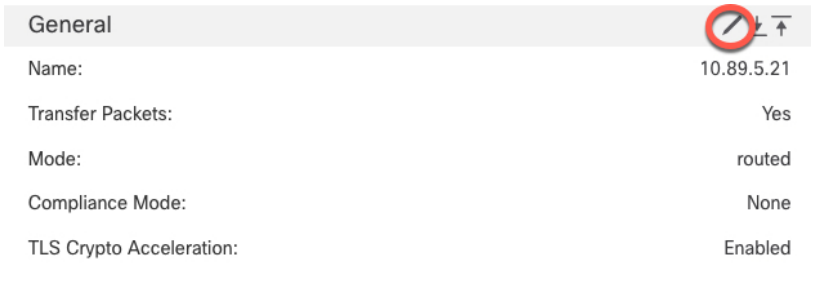
그림 ***18:*** 문제 해결



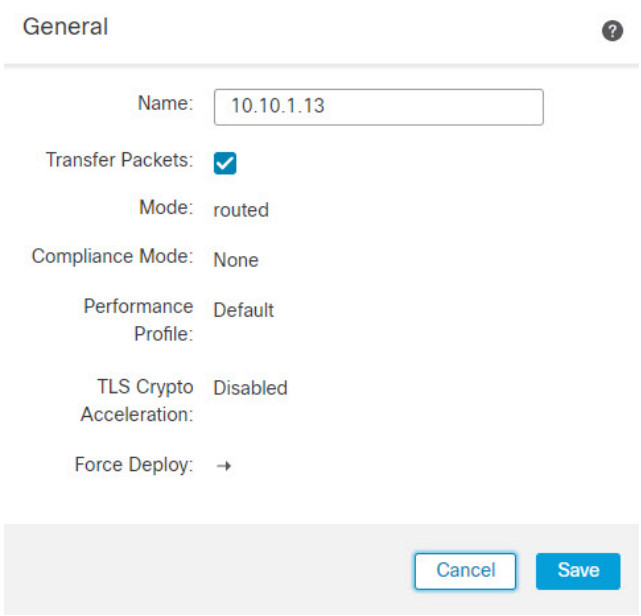
- **License(**라이선스**)** - 편집 (✎)을 클릭하여 라이선스 등록을 설정할 수 있습니다.

단계 **4** **Devices(**디바이스**)** > **Device Management(**디바이스 관리**)** > 디바이스**(**디바이스**)**의 오른쪽 상단 드롭다운 메뉴에서 클러스터의 각 멤버를 선택하고 다음 설정을 구성할 수 있습니다.

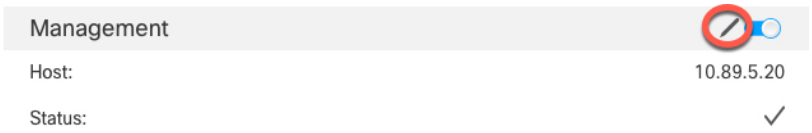
- **General(**일반**)** > **Name (**이름**)** - 편집 (✎)을 클릭하여 클러스터 멤버 표시 이름을 변경합니다.



그런 다음 **Name(**이름**)** 필드를 설정합니다.



- **Management(**관리**) > Host(**호스트**)**—디바이스 설정에서 관리 IP 주소를 변경하는 경우 새 주소를 management center에 일치시켜야 네트워크의 디바이스와 연결할 수 있습니다. **Management(**관리**)** 영역에서 **Host(**호스트**)** 주소를 편집합니다.



클러스터 상태 모니터링 설정 구성

Cluster(클러스터**)** 페이지의 **Cluster Health Monitor Settings(**클러스터 상태 모니터링 설정**)** 섹션은 아래 표의 설정을 표시합니다.

그림 ***19:*** 클러스터 상태 모니터링 설정

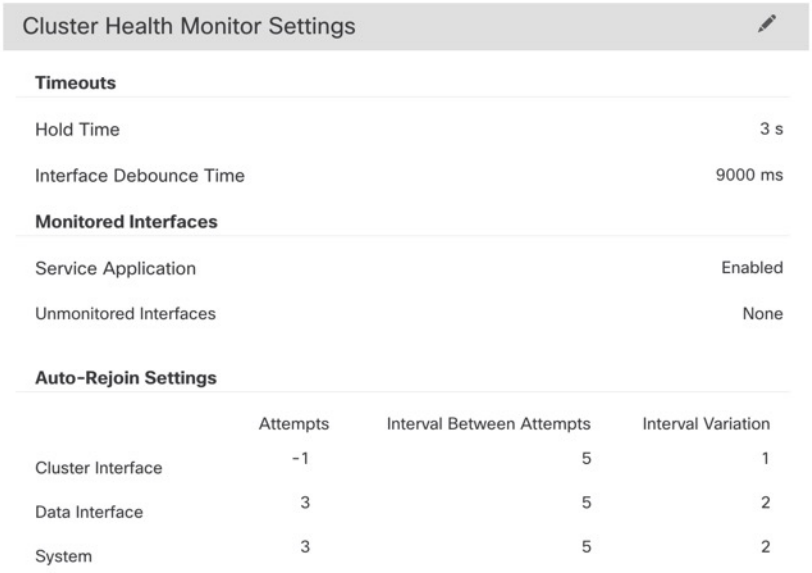


표 ***6:*** 클러스터 상태 모니터링 설정 섹션 표 필드

필드	설명
시간 초과	
보류 시간	노드 시스템 상태를 확인하기 위해 클러스터 노드에서는 다른 노드에 대한 클러스터 제어 링크에서 하트비트 메시지를 보냅니다. 노드가 피어 노드의 하트비트 메시지를 대기 시간 내에 수신하지 않을 경우, 해당 피어 노드는 응답하지 않거나 중지된 상태로 간주됩니다.

필드	설명
인터페이스 디바운스 시간	인터페이스 디바운스 시간은 노드가 인터페이스에 장애가 발생한 것으로 간주하고 노드가 클러스터에서 제거되기 전까지의 시간입니다.
모니터링 인터페이스	인터페이스 상태 검사에서는 링크 오류 여부를 모니터링합니다. 지정된 논리적 인터페이스에 대한 모든 물리적 포트가 특정 노드에서 오류가 발생했지만 다른 노드에 있는 동일한 논리적 인터페이스에서 활성 포트가 있는 경우 이 노드는 클러스터에서 제거됩니다. 노드에서 클러스터의 멤버를 제거하기 전까지 걸리는 시간은 인터페이스의 유형에 따라, 그리고 해당 노드가 설정된 노드인지 또는 클러스터에 참가하는지에 따라 달라집니다.
서비스 애플리케이션	Snort 및 disk-full 프로세스의 모니터링 여부를 표시합니다.
모니터링되지 않는 인터페이스	모니터링되지 않는 인터페이스를 표시합니다.
자동 재연결 설정	
클러스터 인터페이스	클러스터 제어 링크 장애에 대한 자동 다시 참가 설정을 표시합니다.
데이터 인터페이스	데이터 인터페이스 실패에 대한 자동 다시 참가 설정을 표시합니다.
시스템	내부 오류에 대한 자동 다시 참가 설정을 표시합니다. 내부 오류 포함: 애플리케이션 동기화 시간 초과, 일치하지 않는 애플리케이션 상태 등

참고 시스템 상태 확인을 비활성화하면 시스템 상태 확인이 비활성화되었을 때 적용되지 않는 필드가 표시되지 않습니다.

이 섹션에서 이러한 설정을 할 수 있습니다.

모든 포트 채널 ID, 단일 물리적 인터페이스 ID는 물론 Snort 및 디스크 전체 프로세스도 모니터링할 수 있습니다. 상태 모니터링은 VNI 또는 BVI 같은 VLAN 하위 인터페이스 또는 가상 인터페이스에서 수행되지 않습니다. 클러스터 제어 링크의 모니터링을 구성할 수 없습니다. 이 링크는 항상 모니터링됩니다.

프로시저

단계 **1** **Devices(**디바이스**)** > **Device Management(**디바이스 관리**)**을(를) 선택합니다.

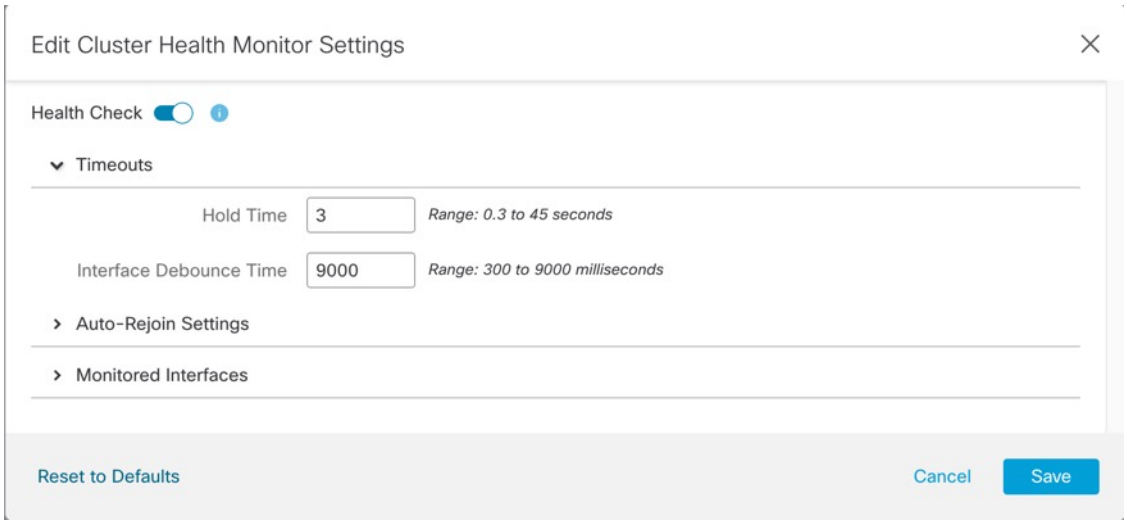
단계 **2** 수정할 클러스터 옆의 편집 (✎)을 클릭합니다.

단계 **3** **Cluster(**클러스터**)**를 클릭합니다.

단계 **4** **Cluster Health Monitor Settings(**클러스터 상태 모니터링 설정**)** 섹션에서 편집 (✎)을 클릭합니다.

단계 **5** **Health Check(**상태 확인**)** 슬라이더를 클릭하여 시스템 상태 확인을 비활성화합니다.

그림 ***20:*** 시스템 상태 확인 비활성화



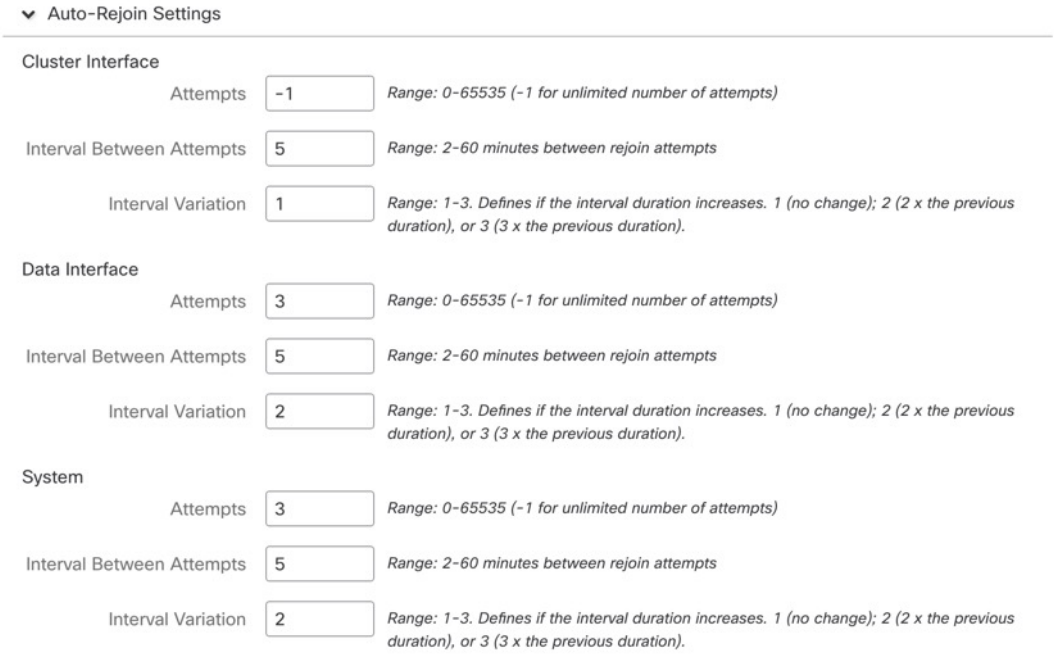
토폴로지 변경 사항(예: 데이터 인터페이스 추가 또는 제거, 노드 또는 스위치의 인터페이스 활성화 또는 비활성화, 추가 스위치를 추가하여 VSS 또는 vPC 구성)이 발생할 경우 시스템 상태 검사 기능을 비활성화하고 비활성화된 인터페이스에 대한 인터페이스 모니터링도 비활성화해야 합니다. 토폴로지 변경이 완료되고 구성 변경 사항이 모든 노드와 동기화되면 시스템 상태 검사 기능 및 모니터링되는 인터페이스를 다시 활성화할 수 있습니다.

단계 **6** 보류 시간 및 인터페이스 디바운스 시간을 구성합니다.

- 보류 시간—노드 하트비트 상태 메시지 사이의 시간을 결정하는 보류 시간을 0.3초에서 45초 사이로 설정합니다. 기본값은 3초입니다.
- **Interface Debounce Time(**인터페이스 디바운스 시간**)**—디바운스 시간을 300~9000밀리초 범위에서 설정합니다. 기본값은 500ms입니다. 값이 낮을수록 인터페이스 오류 탐지를 더 빠르게 수행할 수 있습니다. 디바운스 시간을 더 낮게 구성하면 오탐의 가능성이 증가합니다. 인터페이스 상태 업데이트가 발생하는 경우, 인터페이스를 실패로 표시하고 노드가 클러스터에서 제거되기 전에 노드는 지정되어 있는 밀리초 동안 대기합니다. 가동 중단 상태에서 가동 상태로 전환되는 EtherChannel의 경우(예: 스위치 다시 로드됨 또는 EtherChannel에서 스위치 활성화됨), 디바운스 시간이 더 길어 다른 클러스터 노드가 포트 번들링 시 더 빨랐다는 이유만으로 인터페이스가 클러스터 노드에서 실패한 것으로 표시되는 것을 방지할 수 있습니다.

단계 **7** 상태 검사에 실패한 후에 자동 다시 참가 클러스터 설정을 맞춤화합니다.

그림 ***21:*** 자동 재연결 설정 구성

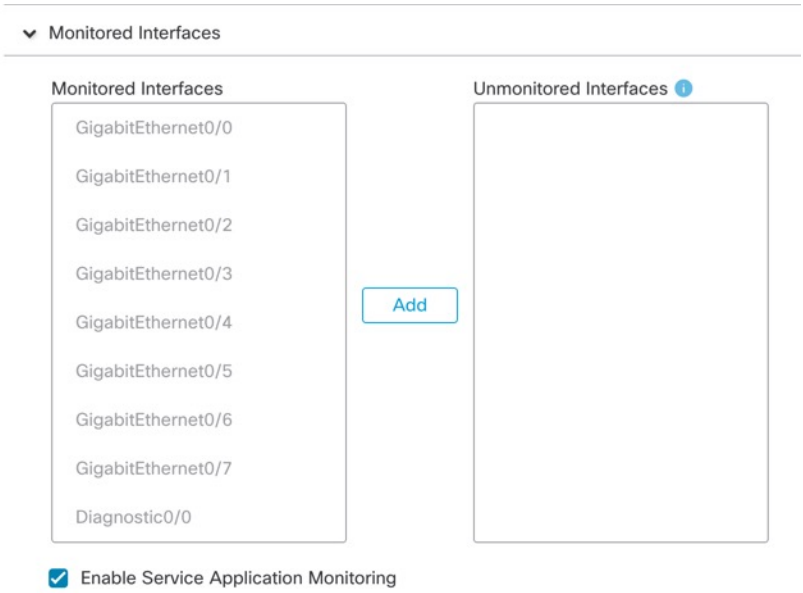


Cluster Interface(클러스터 인터페이스**)**, **Data Interface(**데이터 인터페이스**)** 및 **System(**시스템**)**에 대해 다음 값을 설정합니다(내부 장애에는 애플리케이션 동기화 시간 초과, 일관되지 않은 애플리케이션 상태 등이 포함됨).

- **Attempts(**시도 횟수**)** — 다시 참가 시도 횟수를 -1~65535 범위에서 설정합니다. **0**은 자동 다시 참가를 비활성화합니다. **Cluster Interface(**클러스터 인터페이스**)**의 기본값은 -1(무제한)입니다. **Data Interface(**데이터 인터페이스**)** 및 **System(**시스템**)**의 기본값은 3입니다.
- **Interval Between Attempts(**시도 간의 간격**)**—다시 참가 시도 간의 간격 기간(분)을 2~60분 사이로 정의합니다. 기본값은 5분입니다. 노드가 클러스터에 다시 조인하려고 시도하는 최대 총 시간은 마지막 장애 시간으로부터 14400분(10일)으로 제한됩니다.
- **Interval Variation(**간격 변동**)**—간격 기간이 증가하는지 여부를 정의합니다. 1~3 사이의 값 설정: **1**(변경 없음), **2**(2 x 이전 기간) 또는 **3**(3 x 이전 기간)입니다. 예를 들어, 간격 기간을 5분으로 설정하고 변수를 2로 설정하면 첫 번째 시도가 5분 후에 일어나고 두 번째 시도는 10분(2 x 5), 세 번째 시도는 20분(2 x 10) 후에 일어납니다. 기본값은 **Cluster Interface(**클러스터 인터페이스**)**의 경우 **1**이고 **Data Interface(**데이터 인터페이스**)** 및 **System(**시스템**)**의 경우 **2**입니다.

단계 **8** **Monitored Interfaces(**모니터링된 인터페이스**)** 또는 **Unmonitored Interfaces(**모니터링되지 않는 인터페이스**)** 창에서 인터페이스를 이동하여 모니터링되는 인터페이스를 구성합니다. 또한 **Enable Service Application Monitoring(**서비스 애플리케이션 모니터링 활성화**)**을 선택하거나 선택 취소하여 Snort 및 디스크 꽉 찬 프로세스의 모니터링을 활성화하거나 비활성화할 수 있습니다.

그림 ***22:*** 모니터링되는 인터페이스 구성



인터페이스 상태 검사에서는 링크 오류 여부를 모니터링합니다. 지정된 논리적 인터페이스에 대한 모든 물리적 포트가 특정 노드에서 오류가 발생했지만 다른 노드에 있는 동일한 논리적 인터페이스에서 활성 포트가 있는 경우 이 노드는 클러스터에서 제거됩니다. 노드에서 클러스터의 멤버를 제거하기 전까지 걸리는 시간은 인터페이스의 유형에 따라, 그리고 해당 노드가 설정된 노드인지 또는 클러스터에 참가하는지에 따라 달라집니다. 상태 검사는 모든 인터페이스와 Snort 및 디스크 풀 프로세스에 대해 기본적으로 활성화됩니다.

필수가 아닌 인터페이스에 대한 상태 모니터링을 비활성화할 수 있습니다.

토폴로지 변경 사항(예: 데이터 인터페이스 추가 또는 제거, 노드 또는 스위치의 인터페이스 활성화 또는 비활성화, 추가 스위치를 추가하여 VSS 또는 vPC 구성)이 발생할 경우 시스템 상태 검사 기능을 비활성화하고 비활성화된 인터페이스에 대한 인터페이스 모니터링도 비활성화해야 합니다. 토폴로지 변경이 완료되고 구성 변경 사항이 모든 노드와 동기화되면 시스템 상태 검사 기능 및 모니터링되는 인터페이스를 다시 활성화할 수 있습니다.

단계 **9** **Save(**저장**)**를 클릭합니다.

단계 **10** 구성 변경 사항을 구축합니다를 참조하십시오.

클러스터 노드 관리

•

클러스터링 비활성화

노드 삭제를 준비하거나 유지 보수를 위해 일시적으로 노드를 비활성화할 수 있습니다. 이 절차는 노드를 일시적으로 비활성화하기 위함이며, management center 디바이스 목록에 노드를 유지해야 합니다. 노드가 비활성 상태가 되면 모든 데이터 인터페이스가 종료됩니다.

참고 클러스터링을 비활성화하기 전에 노드의 전원을 끄지 마십시오.

프로시저

단계 **1** 비활성화하려는 유닛에 대해 **Devices(**디바이스**)** > **Device Management(**디바이스 관리**)**를 선택하고 추가 ( )를 클릭하고 **Disable Clustering(**클러스터링 비활성화**)**을 선택합니다.

단계 **2** 노드에서 클러스터링을 비활성화하고자 함을 확인합니다.

노드가 **Devices(**디바이스**)** > **Device Management(**디바이스 관리**)** 목록에서 그 이름 옆에 **(Disabled(**비활성화 됨**))**로 표시됩니다.

단계 **3** 클러스터링을 다시 활성화하려면 클러스터 재참가, 66 페이지의 내용을 참조하십시오.

클러스터 재참가

예를 들어 인터페이스 오류 등으로 노드가 클러스터에서 제거되거나 수동으로 클러스터링을 비활성화한 경우 클러스터를 수동으로 다시 참가시킬 수 있습니다. 클러스터 다시 조인을 시도하기 전에 오류가 해결되었는지 확인하십시오. 노드가 클러스터에서 제거되는 이유에 대한 자세한 내용은 클러스터 다시 참가, 84 페이지의 내용을 참조하십시오.

프로시저

단계 **1** 다시 활성화하려는 유닛에 대해 **Devices(**디바이스**)** > **Device Management(**디바이스 관리**)** 를 선택하고 추가 ( )를 클릭하고 **Enable Clustering(**클러스터링 다시 활성화**)**을 선택합니다.

단계 **2** 노드에서 클러스터링을 활성화하고자 함을 확인합니다.

클러스터 노드 조정

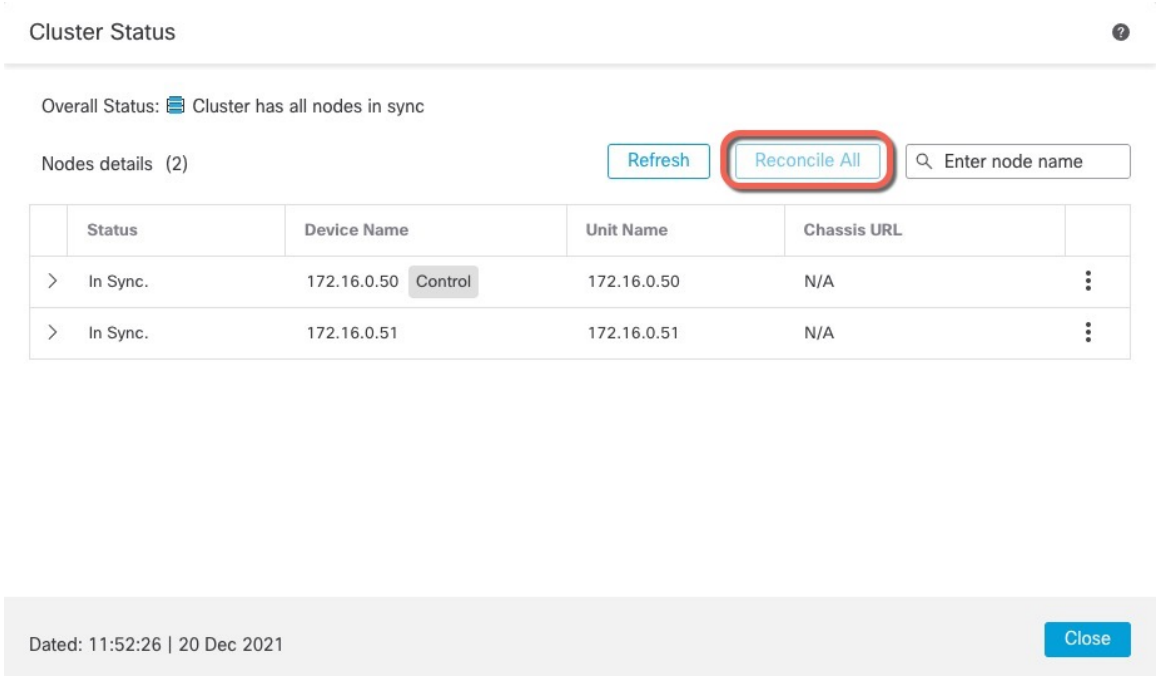
클러스터 노드 등록에 실패하면 디바이스에서 management center에 대해 클러스터 멤버십을 다시 조정합니다. 예를 들어, management center이 특정 프로세스 중이거나 네트워크에 문제가 있는 경우, 데이터 노드 등록에 실패할 수 있습니다.

프로시저

단계 **1** 클러스터에 대해 **Devices(**디바이스**)** > **Device Management(**디바이스 관리**)** 추가 ( )를 선택한 다음 **Cluster Live Status(**클러스터 라이브 상태**)**를 선택하여 **Cluster Status(**클러스터 상태**)** 대화 상자를 엽니다.

단계 **2** **Reconcile All(**모두 조정**)**을 클릭합니다.

그림 ***23:*** 모두 조정



클러스터 상대에 대한 자세한 내용은 클러스터 모니터링, 68 페이지을 참고하십시오.

클러스터 또는 노드를 삭제(등록 취소)하고 새 **Management Center**에 등록

management center에서 클러스터를 등록 취소할 수 있습니다. 그래도 클러스터는 그대로 유지됩니다. 클러스터를 새 management center에 추가하려는 경우 클러스터를 등록 취소할 수 있습니다.

클러스터에서 노드를 분리하지 않고 management center에서 노드를 등록 취소할 수도 있습니다. 노드는 management center에 표시되지 않지만 여전히 클러스터의 일부이며 트래픽을 계속 전달하며 제어 노드가 될 수도 있습니다. 현재 제어 노드는 등록 취소할 수 없습니다. management center에서 더 이상 연결할 수 없지만 관리 연결 문제를 해결하는 동안 클러스터의 일부로 계속 유지하려는 경우 노드를 등록 취소할 수 있습니다.

클러스터 등록 취소:

- management center와 클러스터 간 모든 통신이 단절됩니다.

- **Device Management(**디바이스 관리**)** 페이지에서 클러스터를 제거합니다.
- 디바이스가 NTP를 사용하여 management center에서 시간을 수신하도록 클러스터의 플랫폼 설정 정책이 구성된 경우 디바이스를 로컬 시간 관리로 되돌립니다.
- 구성을 그대로 유지하므로 클러스터가 트래픽을 계속 처리합니다.

  NAT 및 VPN, ACL 및 인터페이스 구성과 같은 정책은 그대로 유지됩니다.

클러스터를 동일하거나 다른 에 다시 등록하면 설정이 제거되므로 클러스터는 이 시점에서 트래픽 처리를 중지합니다. 클러스터 구성은 그대로 유지되므로 클러스터 전체를 추가할 수 있습니다.management center 등록 시 액세스 제어 정책을 선택할 수 있지만, 등록 후에 다른 정책을 다시 적용하고 구성을 구축해야만 트래픽을 다시 처리할 수 있습니다.

시작하기 전에

이 절차에서는 노드 중 하나에 대한 CLI 액세스가 필요합니다.

프로시저

단계 **1** **Devices(**디바이스**)** > **Device Management(**디바이스 관리**)**를 선택하고 클러스터 또는 노드로 추가 ( ) 을 클릭하고 **Delete(**삭제**)**를 선택합니다.

단계 **2** 클러스터 또는 노드를 Delete(삭제)하라는 프롬프트가 표시됩니다. **Yes(**예**)**를 클릭합니다.

단계 **3** 클러스터 멤버 중 하나를 새 디바이스로 추가하여 클러스터를 새(또는 동일한) management center에 등록할 수 있습니다.

클러스터 노드 중 하나만 디바이스로 추가하면 나머지 클러스터 노드가 검색됩니다.

a) 단일 클러스터 노드의 CLI에 연결하고 **configure manager add** 명령을 사용하여 새 management center를 식별합니다.
b) **Devices(**디바이스**)** > **Device Management(**디바이스 관리**)**를 선택한 다음 **Add Device(**디바이스 추가**)**를 클릭합니다.

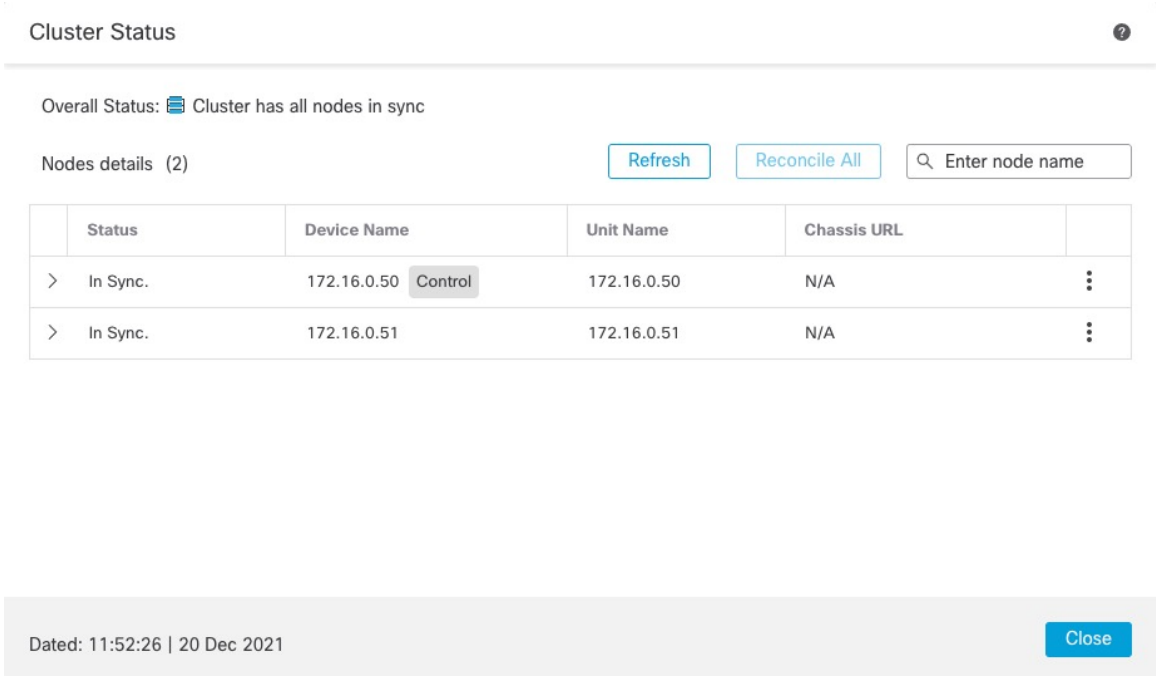
단계 **4** 삭제된 노드를 다시 추가하려면 클러스터 노드 조정, 66 페이지의 내용을 참조하십시오.

클러스터 모니터링

management center과 threat defense CLI에서 클러스터를 모니터링할 수 있습니다.

- **Cluster Status(**클러스터 상태**)**대화 상자는 **Devices(**디바이스**)** > **Device Management** > 추가 ( ) 아이콘 또는 **Devices(**디바이스**)** > **Device Management(**디바이스 관리**)** > **Cluster(**클러스터**)** 페이지 > **General(**일반**)** 영역 > **Cluster Live Status(**클러스터 라이브 상태**)** 링크에서 제공됩니다.

그림 ***24:*** 클러스터 상태



제어 노드에는 역할을 식별하는 그래픽 표시기가 있습니다.

클러스터 멤버 상태에는 다음 상태가 포함됩니다.

- 동기화 중 - 노드가 management center에 등록되었습니다.
- 등록 보류 중 - 유닛이 클러스터의 일부이지만 아직 management center에 등록되지 않았습니다. 노드 등록에 실패하는 경우, **Reconcile(**조정**)All(**모두**)**을 클릭하여 등록을 다시 시도할 수 있습니다.
- 클러스터링이 비활성화됨 - 노드가 management center에 등록되었지만, 클러스터의 비활성 멤버입니다. 클러스터링 구성은 나중에 다시 활성화하려는 경우에도 그대로 유지됩니다. 또는 클러스터에서 노드를 삭제할 수 있습니다.
- 클러스터 참가 중... - 노드가 새시의 클러스터에 참가 중이지만 아직 참가가 완료되지 않았습니다. 참가가 끝나면 management center로 등록합니다.

각 노드에 대해 요약 또는 기록을 볼 수 있습니다.

그림 ***25:*** 노드 요약

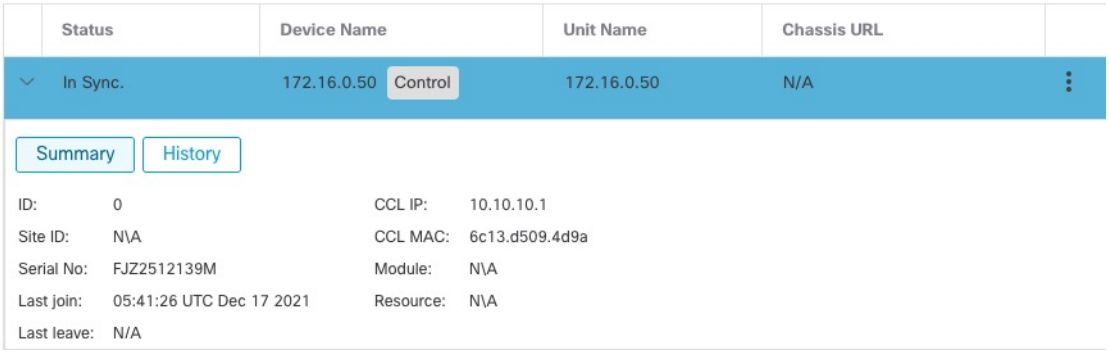
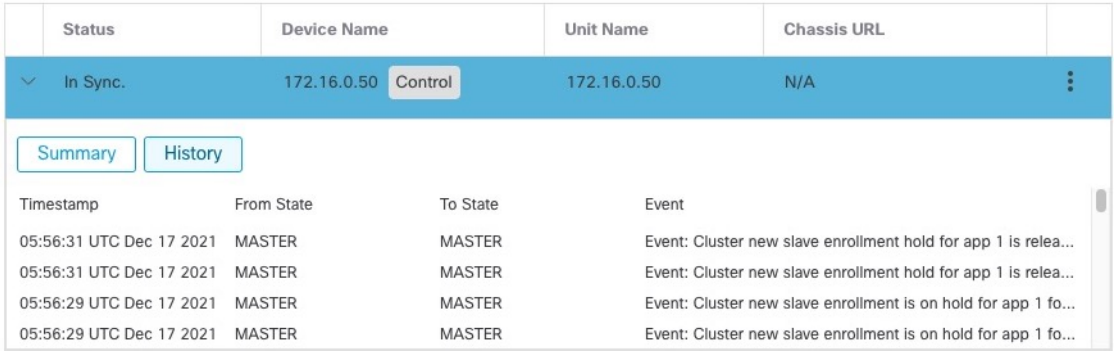


그림 ***26:*** 노드 기록



- 시스템 (⚙) > **Tasks(**작업**)** 페이지로 이동합니다.

  Tasks(작업**)** 페이지는 각 노드 등록에 대한 클러스터 등록 작업의 업데이트를 보여줍니다.

- **Devices(**디바이스**)** > **Device Management(**디바이스 관리**)** > *cluster_name*.

  디바이스 목록 페이지에서 클러스터를 확장하는 경우, IP 주소 옆에 해당 역할과 함께 표시되는 제어 노드를 포함하여 모든 멤버 노드를 볼 수 있습니다. 아직 등록 중인 로드는 로딩 아이콘이 표시됩니다.

- **show cluster** {**access-list** [*acl_name*] | **conn** [**count**] | **cpu** [**usage**] | **history** | **interface-mode** | **memory** | **resource usage** | **service-policy** | **traffic** | **xlate count**}

  전체 클러스터에 대한 집계된 데이터 또는 다른 정보를 보려면 **show cluster** 명령을 사용합니다.

- **show cluster info** [**auto-join** | **clients** | **conn-distribution** | **flow-mobility counters** | **goid** [*options*] | **health** | **incompatible-config** | **loadbalance** | **old-members** | **packet-distribution** | **trace** [*options*] | **transport** { **asp** | **cp**}]

  클러스터 정보를 보려면 **show cluster info** 명령을 사용합니다.

클러스터 상태 모니터 대시보드

클러스터 상태 모니터

threat defense가 클러스터의 제어 노드인 경우 management center는 디바이스 메트릭 데이터 컬렉터에서 다양한 메트릭을 주기적으로 수집합니다. 클러스터 상태 모니터는 다음 구성 요소로 이루어집니다.

- 대시보드 개요 - 클러스터 토폴로지, 클러스터 통계 및 메트릭 차트에 대한 정보를 표시합니다.
  - 토폴로지 섹션에는 클러스터의 라이브 상태, 개별 Threat Defense의 상태, Threat Defense 노드 유형(제어 노드 또는 데이터 노드) 및 디바이스의 상태가 표시됩니다. 디바이스의 상태는 *Disabled(*비활성화됨*)*(디바이스가 클러스터에서 나갈 때), *Added out of box*(퍼블릭 클라우드 클러스터에서 management center에 속하지 않는 추가 노드) 또는 *Normal*(노드의 이상적인 상태)일 수 있습니다.
  - 클러스터 통계 섹션에는 CPU 사용량, 메모리 사용량, 입력 속도, 출력 속도, 활성 연결 및 NAT 변환과 관련된 클러스터의 현재 메트릭이 표시됩니다.

  참고 CPU 및 메모리 메트릭은 데이터 플레인 및 Snort 사용량의 개별 평균을 표시합니다.

  - CPU Usage(CPU 사용량), Memory Usage(메모리 사용량), Throughput(처리량) 및 Connections(연결)와 같은 메트릭 차트는 지정된 기간 동안의 클러스터 통계를 도식적으로 표시합니다.
- 부하 분포 대시보드 - 클러스터 노드 전체의 부하 분포를 다음 두 가지 위젯으로 표시합니다:
  - Distribution(배포) 위젯은 클러스터 노드 전체에서 시간 범위의 평균 패킷 및 연결 분포를 표시합니다. 이 데이터는 노드에서 부하가 분산되는 방식을 나타냅니다. 이 위젯을 사용하면 부하 분포의 이상을 쉽게 식별하고 수정할 수 있습니다.
  - Node Statistics(노드 통계) 위젯은 노드 레벨 메트릭을 테이블 형식으로 표시합니다. 클러스터 노드 전체에서 CPU 사용량, 메모리 사용량, 입력 속도, 출력 속도, 활성 연결 및 NAT 변환에 대한 메트릭 데이터를 표시합니다. 이 테이블 보기를 사용하면 데이터의 상관관계를 파악하고 불일치를 쉽게 식별할 수 있습니다.
- Member Performance(멤버 성능) 대시보드 - 클러스터 노드의 현재 메트릭을 표시합니다. 선택기를 사용하여 노드를 필터링하고 특정 노드의 세부 정보를 볼 수 있습니다. 메트릭 데이터에는 CPU 사용량, 메모리 사용량, 입력 속도, 출력 속도, 활성 연결 및 NAT 변환이 포함됩니다.
- CCL 대시보드 - 클러스터 제어 링크 데이터, 즉 입력 및 출력 속도를 그래픽으로 표시합니다.
- 문제 해결 및 링크 - 자주 사용하는 문제 해결 주제 및 절차에 대한 편리한 링크를 제공합니다.
- 시간 범위 - 다양한 클러스터 메트릭 대시보드 및 위젯에 표시되는 정보를 제한하기 위한 조정 가능한 시간 창입니다.

- 사용자 지정 대시보드 - 클러스터 전체 메트릭 및 노드 레벨 메트릭 모두에 대한 데이터를 표시합니다. 그러나 노드 선택은 Threat Defense 메트릭에만 적용되며 노드가 속한 전체 클러스터에는 적용되지 않습니다.

클러스터 상태 보기

이 절차를 수행하려면 관리자, 유지 보수 또는 보안 분석가 사용자여야 합니다.

클러스터 상태 모니터는 클러스터와 해당 노드의 상태에 대한 자세한 보기를 제공합니다. 이 클러스터 상태 모니터는 대시보드 어레이에서 클러스터의 상태 및 추세를 제공합니다.

시작하기 전에

- management center에서 하나 이상의 디바이스에서 클러스터를 생성했는지 확인합니다.

프로시저

단계 **1** 시스템 ( ) > **Health(**상태**)** > **Monitor(**모니터**)**을(를) 선택합니다.

Monitoring(모니터링) 탐색 창을 사용하여 노드별 상태 모니터에 액세스합니다.

단계 **2** 디바이스 목록에서 **Expand(**확장**)** ( ) 및 **Collapse(**축소**)** ( )를 클릭하여 관리되는 클러스터 디바이스 목록을 확장하고 축소합니다.

단계 **3** 클러스터 상태 통계를 보려면 클러스터 이름을 클릭합니다. 클러스터 모니터는 기본적으로 사전 정의된 여러 대시보드에서 상태 및 성능 메트릭을 보고합니다. 메트릭 대시보드에는 다음이 포함됩니다.

- 개요 - 노드, CPU, 메모리, 입출력 속도, 연결 통계, NAT 변환 정보 등 미리 정의된 다른 대시보드의 주요 메트릭을 강조 표시합니다.
- Load Distribution(로드 분포) - 클러스터 노드 전체의 트래픽 및 패킷 분포입니다.
- Member Performance(멤버 성능) - CPU 사용량, 메모리 사용량, 입력 처리량, 출력 처리량, 활성 연결 및 NAT 변환에 대한 노드 레벨 통계.
- CCL - 인터페이스 상태 및 집계 트래픽 통계

레이블을 클릭하여 다양한 메트릭 대시보드를 탐색할 수 있습니다. 지원되는 클러스터 메트릭의 전체 목록은 Cisco Secure Firewall Threat Defense 상태 메트릭을 참고하십시오.

단계 **4** 오른쪽 상단의 드롭다운에서 시간 범위를 설정할 수 있습니다. 시간 범위는 지난 시간처럼 짧은 기간(기본값) 또는 지난 주처럼 긴 기간을 반영할 수 있습니다. 드롭다운에서 **Custom(**사용자 지정**)**을 선택하여 사용자 지정 시작 및 종료 날짜를 설정합니다.

새로 고침 아이콘을 클릭하여 자동 새로 고침을 5분으로 설정하거나 자동 새로 고침을 해제합니다.

단계 **5** 선택한 시간 범위와 관련하여 추세 그래프에서 구축 오버레이를 보려면 구축 아이콘을 클릭합니다.

구축 아이콘은 선택한 시간 범위 동안의 구축 수를 나타냅니다. 세로 줄은 구축 시작 및 종료 시간을 나타냅니다. 다수의 구축이 있는 경우 여러 대역/라인이 나타납니다. 점선 위에 있는 아이콘을 클릭하여 구축 세부 사항을 확인합니다.

단계 **6** (노드별 상태 모니터의 경우) 페이지 상단에서 디바이스 이름의 바로 오른쪽에 있는 알림에서 노드의 **Health Alerts(**상태 알림**)**를 확인합니다.

Health Alerts(상태 알림**)** 위에 포인터를 올려놓으면 노드의 상태 요약이 표시됩니다. 팝업 윈도우에는 상위 5개 상태 알림의 요약이 잘려서 표시됩니다. 팝업을 클릭하여 상태 알림 요약의 세부사항 보기를 엽니다.

단계 **7** (노드별 상태 모니터의 경우) 디바이스 모니터는 기본적으로 사전 정의된 여러 대시보드에서 상태 및 성능 메트릭을 보고합니다. 메트릭 대시보드에는 다음이 포함됩니다.

- Overview(개요) - CPU, 메모리, 인터페이스, 연결 통계 등 사전 정의된 다른 대시보드의 주요 메트릭을 강조 표시합니다. 디스크 사용량 및 중요 프로세스 정보도 포함됩니다.
- CPU - CPU 사용률(프로세스 및 물리적 코어별 CPU 사용률 포함)
- 메모리-데이터 플레인 및 Snort 메모리 사용량을 포함한 디바이스 메모리 사용량입니다.
- Interfaces(인터페이스) - 인터페이스 상태 및 집계 트래픽 통계
- Connections(연결) - 연결 통계(예: 엘리펀트 플로우, 활성 연결, 최대 연결 등) 및 NAT 변환 수.
- Snort - Snort 프로세스와 관련된 통계
- ASP 삭제 - 여러 이유로 인해 삭제된 패킷과 관련된 통계입니다.

레이블을 클릭하여 다양한 메트릭 대시보드를 탐색할 수 있습니다. 지원되는 디바이스 메트릭의 전체 목록은 Cisco Secure Firewall Threat Defense 상태 메트릭을 참고하십시오.

단계 **8** 사용 가능한 메트릭 그룹에서 고유한 변수 집합을 작성하여 사용자 지정 대시보드를 생성하려면 상태 모니터의 오른쪽 상단 모서리에 있는 더하기 기호(+)를 클릭합니다.

클러스터 전체 대시보드의 경우 Cluster metric group(클러스터 메트릭 그룹)을 선택한 다음 메트릭을 선택합니다.

클러스터 메트릭

클러스터 상태 모니터는 클러스터 및 해당 노드와 관련된 통계와 로드 분포, 성능 및 CCL 트래픽 통계의 집계를 추적합니다.

표 *7:* 클러스터 메트릭

메트릭	설명	형식
CPU	클러스터의 노드에 있는 CPU 메트릭의 평균입니다(데이터 플레인 및 snort에 대해 개별적으로).	백분율

메트릭	설명	형식
메모리	클러스터의 노드에 있는 메모리 메트릭의 평균입니다(데이터 플레인 및 snort에 대해 개별적으로).	백분율
데이터 처리량	클러스터에 대한 수신 및 발신 데이터 트래픽 통계입니다.	바이트
CCL 처리량	클러스터에 대한 수신 및 발신 CCL 트래픽 통계입니다.	바이트
연결	클러스터의 활성 연결 수입니다.	숫자
NAT 변환	클러스터에 대한 NAT 변환 수.	숫자
배포	초당 클러스터의 연결 분포 수입니다.	숫자
패킷	초당 클러스터의 패킷 배포 수입니다.	숫자

클러스터 문제 해결

CCL Ping 툴을 사용하여 클러스터 제어 링크가 올바르게 작동하는지 확인할 수 있습니다. 디바이스 및 클러스터에 제공되는 다음과 같은 툴을 사용할 수도 있습니다.

- 문제 해결 파일 - 노드가 클러스터에 조인하지 못하면 문제 해결 파일이 자동으로 생성됩니다. 또한 **Devices(**디바이스**)** > **Device Management(**디바이스 관리**)** > **Cluster(**클러스터**)** > **General(**일반**)** 영역에서 문제 해결 파일을 생성하고 다운로드할 수 있습니다.

  추가 (⋮)를 클릭하고 **Troubleshoot Files(**문제 해결 파일**)**를 선택하여 **Device Management(**디바이스 관리**)** 페이지에서 파일을 생성할 수도 있습니다.

- CLI 출력 - **Devices(**디바이스**)** > **Device Management(**디바이스 관리**)** > **Cluster(**클러스터**)** > **General(**일반**)** 영역에서 클러스터 문제를 해결하는 데 도움이 될 수 있는 사전 정의된 CLI 출력 집합을 볼 수 있습니다. 다음 명령은 클러스터에 대해 자동으로 실행됩니다.
  - **show running-config cluster**
  - **show cluster info**
  - **show cluster info health**
  - **show cluster info transport cp**
  - **show version**
  - **show asp drop**
  - **show counters**
  - **show arp**

- **show int ip brief**
- **show blocks**
- **CPU** 상세정보 표시
- **show interface** *ccl_interface*
- **ping** *ccl_ip* 크기 *ccl_mtu* 반복 **2**

Command(명령) 필드에 임의의 **show** 명령을 입력할 수도 있습니다.

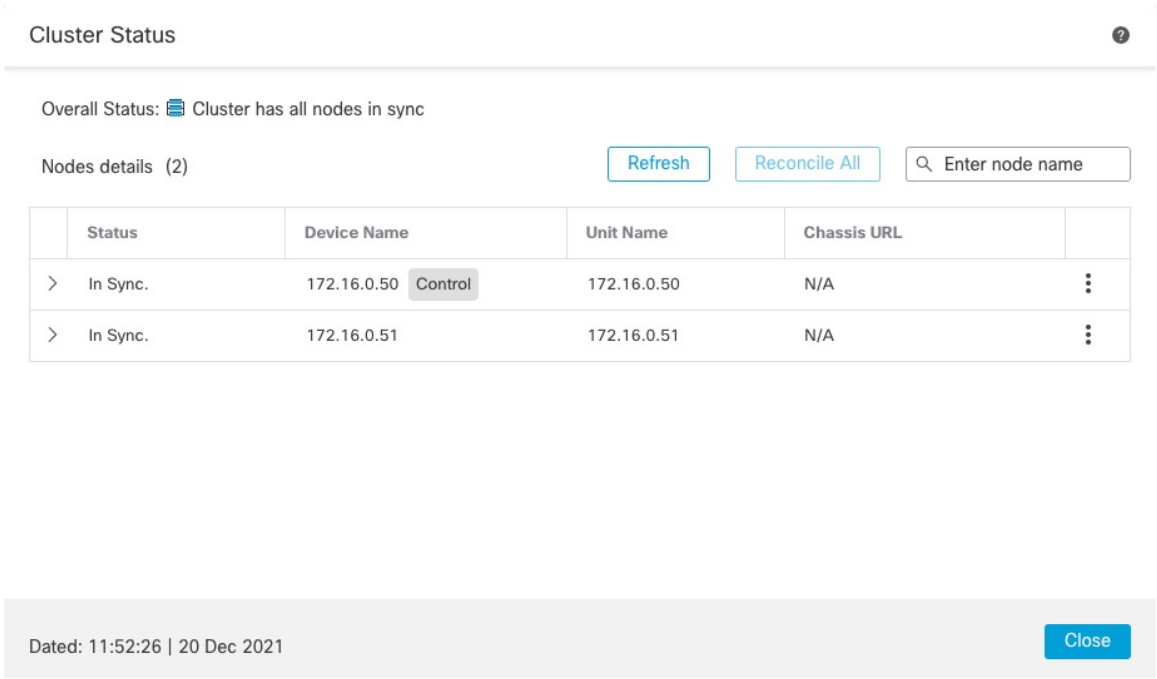
클러스터 제어 링크에서 **ping** 수행

ping을 수행하여 모든 클러스터 노드가 클러스터 제어 링크를 통해 서로 연결 가능한지 확인할 수 있습니다. 노드가 클러스터에 조인하지 못하는 주요 원인 중 하나는 잘못된 클러스터 제어 링크 구성입니다. 예를 들어, 클러스터 제어 링크 MTU가 연결 스위치 MTU보다 높게 설정될 수 있습니다.

프로시저

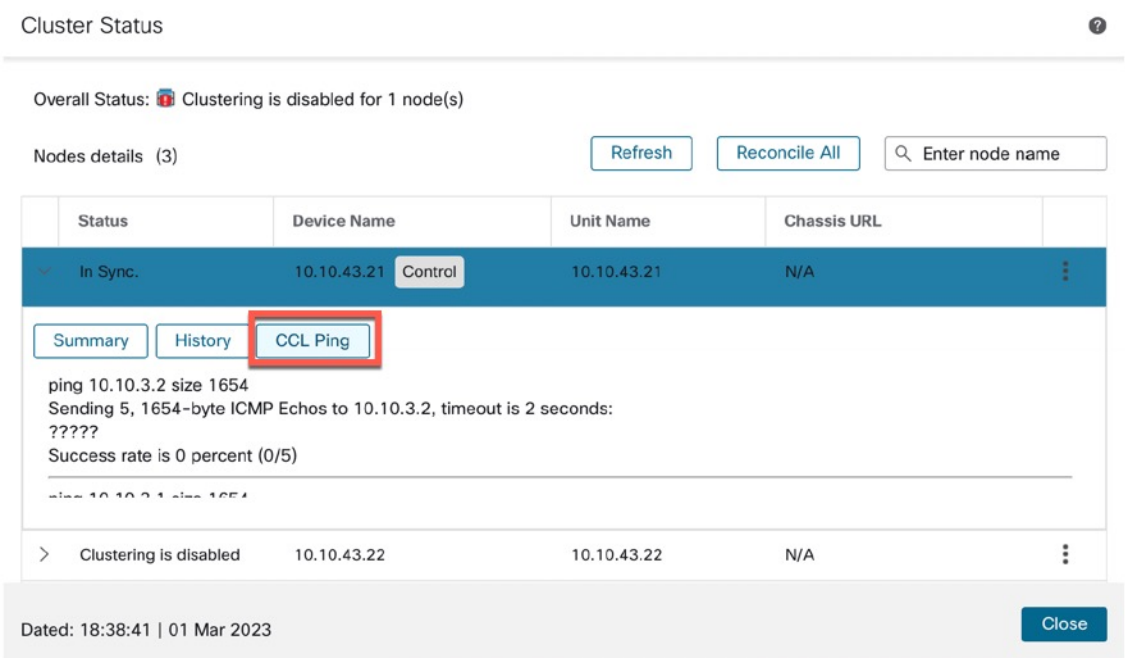
단계 **1** **Devices(**디바이스**)** > **Device Management(**디바이스 관리**)**를 선택한 후 클러스터에 옆에 있는 추가 (⋮) 아이콘을 클릭하고 > **Cluster Live Status(**> 클러스터 라이브 상태**)**를 선택합니다.

그림 ***27:*** 클러스터 상태



단계 **2** 노드 중 하나를 확장하고 **CCL Ping**을 클릭합니다.

그림 ***28: CCL Ping***



노드는 최대 MTU와 일치하는 패킷 크기를 사용하여 클러스터 제어 링크에 대한 ping을 두 번째 노드로 전송합니다.

클러스터 업그레이드

threat defense virtual 클러스터를 업그레이드하려면 다음 단계를 수행합니다.

프로시저

단계 **1** 대상 이미지 버전을 클라우드 이미지 스토리지에 업로드합니다.

단계 **2** 업데이트된 대상 이미지 버전으로 클러스터의 클라우드 인스턴스 템플릿을 업데이트합니다.

a) 대상 이미지 버전으로 인스턴스 템플릿의 사본을 생성합니다.

b) 새로 생성한 템플릿을 클러스터 인스턴스 그룹에 연결합니다.

단계 **3** management center에 대상 이미지 버전 업그레이드 패키지를 업로드합니다.

단계 **4** 업그레이드할 클러스터에서 준비도 확인을 수행합니다.

단계 **5** 준비도를 확인한 후 업그레이드 패키지 설치를 시작합니다.

단계 **6** management center에서는 한 번에 하나씩 클러스터 노드를 업그레이드합니다.

단계 **7** 클러스터를 성공적으로 업그레이드하면 management center에 알림이 표시됩니다.

업그레이드 후에 인스턴스의 일련 번호와 UUID는 변경되지 않습니다.

참고

- Management Center에서 클러스터 업그레이드를 시작하는 경우, Threat Defense Virtual 디바이스가 사후 업그레이드 재부팅 프로세스 중에 실수로 종료되거나 Auto Scaling 그룹에 의해 교체되지 않도록 합니다. 이를 방지하려면 AWS 콘솔로 이동하여 **Auto scaling group(Auto Scaling** 그룹**) -> Advanced configurations(**고급 구성**)**를 클릭하고 Health Check(상태 확인) 및 Replace Unhealthy(비정상 교체) 프로세스를 일시 중단합니다. 업그레이드가 완료되면 **Advanced configurations(**고급 구성**)**로 다시 이동한 후 일시 중단된 프로세스를 모두 제거하여 비정상 인스턴스를 탐지합니다.
- AWS에 구축된 클러스터를 주 릴리스에서 패치 릴리스로 업그레이드한 다음 클러스터를 확장하면 새 노드는 패치 릴리스 대신 주 릴리스 버전을 제공합니다. 그런 다음 각 노드를 Management Center에서 패치 릴리스로 수동으로 업그레이드해야 합니다.

  또는 패치가 적용되어 있고 Day 0 구성이 없는 독립형 Threat Defense Virtual 인스턴스의 스냅샷에서 AMI(Amazon Machine Image)를 생성할 수도 있습니다. 클러스터 구축 템플릿에서 이 AMI를 사용합니다. 클러스터를 확장할 때 표시되는 새 노드에는 패치 릴리스가 적용됩니다.

클러스터링에 대한 참조

이 섹션에는 클러스터링이 작동하는 방식에 대한 자세한 정보가 포함되어 있습니다.

Threat Defense 기능 및 클러스터링

일부 threat defense 기능은 클러스터링이 지원되지 않으며, 일부 기능은 기본 유닛에서만 지원됩니다. 기타 기능의 경우 올바르게 사용하는 데 필요한 주의 사항이 있을 수 있습니다.

지원되지 않는 기능 및 클러스터링

이러한 기능은 클러스터링을 사용하도록 설정한 경우 구성할 수 없으며 명령이 거부됩니다.

참고 클러스터링으로도 지원되지 않는 FlexConfig 기능(예: WCCP 검사)을 보려면 ASA 일반 운영 설정 가이드를 참조하십시오. FlexConfig를 사용하면 management center GUI에 없는 여러 ASA 기능을 설정할 수 있습니다.

- 원격 액세스 VPN(SSL VPN 및 IPsec VPN)
- DHCP 클라이언트, 서버, 프록시 DHCP 릴레이가 지원됩니다.
- Virtual Tunnel Interface(VTI)
- 고가용성
- 통합 라우팅 및 브리징

- FMC UCAPL/CC 모드

클러스터링을 위한 중앙 집중식 기능

다음 기능은 제어 노드에서만 지원되며 클러스터에 확장되지 않습니다.

참고 중앙 집중식 기능의 트래픽은 클러스터 제어 링크를 통해 멤버 노드에서 제어 노드로 전달됩니다.

리밸런싱 기능을 사용할 경우, 중앙 집중식 기능의 트래픽은 트래픽이 중앙 집중식 기능으로 분류되기 전에 비 제어 노드로 리밸런싱될 수 있습니다. 이렇게 되면 해당 트래픽은 제어 노드로 다시 전송됩니다.

중앙 집중식 기능의 경우 제어 노드에 오류가 발생하면 모든 연결이 취소되며 새 제어 노드에서 연결을 다시 설정해야 합니다.

참고 클러스터링으로도 집중되는 FlexConfig 기능(예: RADIUS 검사)을 보려면 ASA 일반 운영 설정 가이드를 참조하십시오. FlexConfig를 사용하면 management center GUI에 없는 여러 ASA 기능을 설정할 수 있습니다.

- 다음과 같은 애플리케이션 감시:
  - DCERPC
  - ESMTP
  - NetBIOS
  - PPTP
  - RSH
  - SQLNET
  - SUNRPC
  - TFTP
  - XDMCP
- 고정 경로 모니터링

Cisco TrustSec 및 클러스터링

제어 노드에서만 보안 그룹 태그(SGT) 정보를 학습합니다. 그런 다음 제어 노드에서는 SGT를 데이터 노드에 제공하며, 데이터 노드에서는 보안 정책을 기준으로 SGT의 일치 여부를 결정할 수 있습니다.

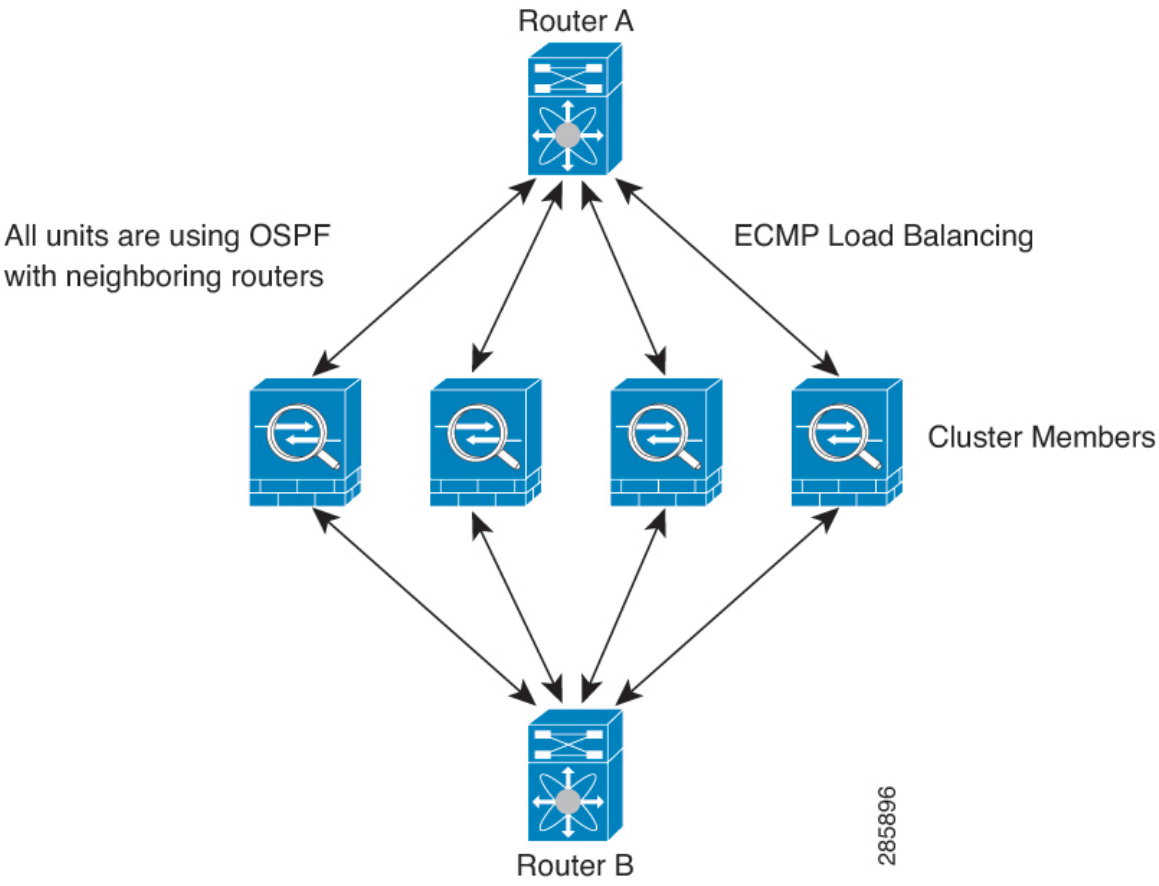
연결 설정 및 클러스터링

연결 제한은 클러스터 전체에서 시행됩니다. 각 노드에는 브로드캐스트 메시지를 기반으로 한 클러스터 전체의 카운터 값이 표시됩니다. 효율성을 고려하여 클러스터 전체에 구성된 연결 제한이 제한 수에 정확하게 적용되지 않을 수 있습니다. 각 노드는 언제든지 클러스터 전체 카운터 값을 과대 평가하거나 과소 평가할 수 있습니다. 그러나 로드 밸런싱된 클러스터에서는 시간이 지남에 따라 정보가 업데이트됩니다.

동적 라우팅 및 클러스터링

개별 인터페이스 모드의 경우 각 노드에서는 라우팅 프로토콜을 독립형 라우터로 실행하며, 경로에 대한 정보 학습은 각 노드에서 개별적으로 수행합니다.

그림 ***29:*** 개별 인터페이스 모드의 동적 라우팅



위 다이어그램에서 라우터 A는 라우터 B에 각각 노드를 통한 4개의 Equal-Cost 경로가 있다는 정보를 파악합니다. ECMP는 4개 경로 간의 트래픽을 로드 밸런싱하는 데 사용됩니다. 각각의 노드는 외부 라우터와 통신할 경우 다른 라우터 ID를 선택합니다.

라우터 ID에 대한 클러스터 풀을 구성하여 노드마다 개별 라우터 ID를 보유하도록 해야 합니다.

FTP 및 클러스터링

- 다른 클러스터 멤버가 FTP 데이터 채널 및 제어 채널의 흐름을 소유한 경우, 데이터 채널 소유자 유닛에서는 유휴 시간 제한 업데이트를 제어 채널 소유자에게 주기적으로 전송하고 유휴 시간 제한 값을 업데이트합니다. 그러나 제어 흐름 소유자가 다시 로드되고 제어 흐름이 다시 호스팅된 경우, 부모/자식 흐름 관계가 더 이상 유지되지 않으며 제어 흐름 유휴 시간 제한도 업데이트되지 않습니다.

NAT 및 클러스터링

NAT 용도에 대해서는 다음 제한 사항을 참고하십시오.

NAT는 클러스터의 전체 처리량에 영향을 미칠 수 있습니다. 로드 밸런싱 알고리즘은 IP 주소와 포트를 기반으로 할 뿐만 아니라 NAT로 인해 인바운드 및 아웃바운드 패킷의 IP 주소 및/또는 포트가 서로 달라질 수 있으므로, 인바운드 및 아웃바운드 NAT 패킷을 클러스터의 다른 threat defense에 전송할 수 있습니다. 패킷이 NAT 소유자가 아닌 threat defense에 전달되면 해당 패킷은 클러스터 제어 링크를 통해 소유자에게 전달되며 이때 클러스터 제어 링크에 매우 많은 양의 트래픽이 발생합니다. 보안 및 정책 확인 결과에 따라 NAT 소유자가 패킷에 대해 연결을 생성하지 않을 수 있으므로 수신 노드는 소유자에 대한 전달 플로우를 생성하지 않습니다.

클러스터링에 NAT를 계속 사용하려면 다음 지침을 숙지하십시오.

- 프록시 ARP 없음 — 개별 인터페이스에서 프록시 ARP 응답은 매핑된 주소에 전송되지 않습니다. 이렇게 되면 인접한 라우터가 클러스터에 더 이상 존재하지 않을 수 있는 ASA와 피어 관계를 유지하지 못하게 됩니다. 업스트림 라우터에는 기본 클러스터 IP 주소를 나타내는 매핑된 주소에 대한 고정 경로 또는 PBR(Object Tracking 포함)이 필요합니다.

- 포트 블록 할당이 있는 PAT - 이 기능에 대한 다음 지침을 참조하십시오.

  - 호스트당 최대 제한은 클러스터 전체 제한이 아니며 각 노드에서 개별적으로 적용됩니다. 호스트당 최대 제한이 1로 구성된 3-노드 클러스터에서 호스트의 트래픽이 3개 노드 모두에 로드 밸런싱되는 경우 각 노드에 하나씩 3개의 블록이 할당될 수 있습니다.

  - 백업 풀의 백업 노드에서 생성된 포트 블록은 호스트당 최대 제한을 적용할 때 고려되지 않습니다.

  - 완전히 새로운 IP 범위로 PAT 풀을 수정하는 즉석 PAT 규칙 수정을 수행할 경우, 새 풀이 작동하게 되는 동안 여전히 전환 중이던 xlate 백업 요청에 대해 xlate 백업 생성이 실패하게 됩니다. 이러한 동작은 포트 블록 할당 기능과 관련이 없으며, 풀이 분산되고 트래픽이 클러스터 노드 전체에서 부하 분산되는 클러스터 구축 과정에서만 발생하는 일시적인 PAT 풀 문제입니다.

  - 클러스터에서 작업할 때는 단순히 블록 할당 크기를 변경할 수 없습니다. 새 크기는 클러스터에서 각 디바이스를 다시 로드한 후에만 적용됩니다. 각 디바이스를 다시 로드하지 않으려면 모든 블록 할당 규칙을 삭제하고 해당 규칙과 관련된 모든 xlate를 지우는 것이 좋습니다. 그런 다음 블록 크기를 변경하고 블록 할당 규칙을 다시 생성할 수 있습니다.

- 동적 PAT에 대한 NAT 풀 주소 분산 - PAT 풀을 구성하면 클러스터는 풀의 각 IP 주소를 포트 블록으로 나눕니다. 기본적으로 각 블록은 512포트이지만 포트 블록 할당 규칙을 구성하는 경우에

는 블록 설정이 대신 사용됩니다. 이러한 블록은 클러스터의 노드 간에 균등하게 분산되므로 각 노드에는 PAT 풀의 각 IP 주소에 대해 하나 이상의 블록이 있습니다. 따라서 예상되는 PAT 처리된 연결 수에 충분한 경우 클러스터의 PAT 풀에 IP 주소를 하나만 포함할 수 있습니다. PAT 풀 NAT 규칙에 예약된 포트 1~1023을 포함하도록 옵션을 구성하지 않는 한 포트 블록은 1024~65535 포트 범위를 포함합니다.

- 여러 규칙에서 PAT 풀 재사용 - 여러 규칙에서 동일한 PAT 풀을 사용하려면 규칙에서 인터페이스 선택에 주의해야 합니다. 모든 규칙에서 특정 인터페이스를 사용하거나 또는 모든 규칙에서 "any(임의의)"를 사용해야 합니다. 규칙 전체에서 특정 인터페이스와 "any(임의의)"를 혼합할 수 없거나, 시스템에서 클러스터의 오른쪽 노드에 대한 반환 트래픽을 일치시키지 못할 수 있습니다. 규칙 당 고유한 PAT 풀을 사용하는 것은 가장 신뢰할 수 있는 옵션입니다.
- 라운드 로빈 없음 — 클러스터링에서는 PAT 풀을 위한 라운드 로빈을 지원하지 않습니다.
- 확장 PAT 없음 - 클러스터링에서 확장 PAT가 지원되지 않습니다.
- 제어 노드에 의해 관리되는 동적 NAT xlate — 제어 노드에서는 xlate 테이블을 유지하고 데이터 노드에 복제합니다. 동적 NAT가 필요한 연결이 데이터 노드에 전달되고 xlate가 테이블에 없을 경우, 제어 노드에서 xlate를 요청합니다. 데이터 노드에서는 이 연결을 소유합니다.
- 오래된 xlates - 연결 소유자의 xlate 유휴 시간이 업데이트되지 않습니다. 따라서 유휴 시간이 유휴 시간 제한을 초과할 수 있습니다. refcnt가 0인 구성된 시간 초과 값보다 큰 유휴 타이머 값은 오래된 xlate를 나타냅니다.
- 다음을 검사할 수 있는 고정 PAT 없음
  - FTP
  - RSH
  - SQLNET
  - TFTP
  - XDMCP
  - SIP
- 10,000개가 넘는 매우 많은 NAT 규칙이 있는 경우 디바이스 CLI에서 **asp rule-engine transactional-commit nat** 명령을 사용하여 트랜잭션 커밋 모델을 활성화해야 합니다. 그렇지 않으면 노드가 클러스터에 조인하지 못할 수 있습니다.

SIP 검사 및 클러스터링

로드 밸런싱으로 인해 모든 노드에서 제어 플로우를 만들 수 있지만 하위 데이터 플로우는 동일한 노드에 상주해야 합니다.

SNMP 및 클러스터링

SNMP 폴링에는 기본 클러스터 IP 주소가 아닌 로컬 주소를 항상 사용해야 합니다. SNMP 에이전트에서 기본 클러스터 IP 주소를 폴링하면서 새 제어 노드가 선택된 경우, 새 제어 노드에 대한 폴링이 이루어지지 않습니다.

클러스터링과 함께 SNMPv3를 사용할 때 초기 클러스터 형성 후 새 클러스터 노드를 추가하면 SNMPv3 사용자가 새 노드에 복제되지 않습니다. 사용자를 제거하고 다시 추가한 다음 사용자가 새 노드에 복제하도록 강제로 구성을 재구축해야 합니다.

시스템 로그 및 클러스터링

- 클러스터의 각 노드에서는 고유한 syslog 메시지를 생성합니다. 각 노드에서 syslog 메시지 헤더 필드에 동일하거나 다른 디바이스 ID를 사용하도록 로깅을 구성할 수 있습니다. 예를 들어, 호스트 이름 구성은 클러스터의 모든 노드에 의해 복제 및 공유됩니다. 호스트 이름을 디바이스 ID로 사용하도록 로깅을 구성할 경우, 모든 노드에서는 단일 노드에서 생성된 것처럼 보이는 syslog 메시지를 생성합니다. 클러스터 부트스트랩 구성에 할당된 로컬-노드 이름을 디바이스 ID로 사용하도록 로깅을 구성할 경우, syslog 메시지는 다른 노드에서 생성된 것처럼 보입니다.

VPN 및 클러스터링

사이트 간 VPN은 중앙 집중식 기능이며, 마스터 노드에서만 VPN 연결을 지원합니다.

참고 원격 액세스 VPN은 클러스터링으로 지원되지 않습니다.

VPN 기능은 마스터 노드에만 제한되며 클러스터 고가용성 기능을 사용하지 않습니다. 제어 노드에 오류가 발생할 경우, 모든 기존 VPN 연결이 손실되며 VPN 사용자에게는 서비스 중단 메시지가 표시됩니다. 새 제어 노드가 선택되면 VPN 연결을 다시 설정해야 합니다.

PBR 또는 ECMP를 사용할 경우 개별 인터페이스에 연결하려면 항상 로컬 주소가 아닌 기본 클러스터 IP 주소에 연결해야 합니다.

VPN 관련 키 및 인증서는 모든 노드에 복제됩니다.

성능 확장 요소

클러스터에 여러 유닛을 결합할 경우 총 클러스터 성능을 대략 최대 결합 처리량의 약 80%로 예측할 수 있습니다.

예를 들어 모델이 단독으로 실행될 때 약 10Gbps의 트래픽을 처리할 수 있는 경우, 8개 유닛으로 구성된 클러스터의 경우 최대 통합 처리량은 80Gbps(8개 유닛 x 10Gbps)의 약 80%인 64Gbps가 됩니다.

제어 노드 선택

클러스터의 노드는 클러스터 제어 링크로 통신을 수행하여 다음과 같은 방식으로 제어 노드를 선택합니다.

1. 노드에 클러스터링을 사용할 경우(또는 이미 사용 설정된 클러스터링을 처음 시작할 경우), 선택 요청이 3초마다 전송됩니다.
2. 다른 노드의 우선순위가 더 높을 경우 해당 노드가 선택 요청에 응답하게 됩니다. 우선순위는 1에서 100까지 설정되며 1이 가장 높은 우선순위입니다.
3. 45초 후에 우선순위가 더 높은 다른 노드에서 응답을 받지 못한 노드는 제어 노드가 됩니다.

참고 가장 우선순위가 높은 노드가 공동으로 여러 개인 경우, 클러스터 노드 이름과 일련 번호를 사용하여 제어 노드를 결정합니다.

4. 노드가 우선순위가 더 높은 클러스터에 참가한다고 해서 해당 노드가 자동으로 제어 노드가 되는 것은 아닙니다. 기존 제어 노드는 응답이 중지되지 않는 한 항상 제어 노드로 유지되며 응답이 중지될 때에 새 제어 노드가 선택됩니다.
5. 제어 노드가 일시적으로 여러 개 있는 "스플릿 브레인" 시나리오에서는 우선 순위가 가장 높은 노드가 역할을 유지하는 반면 다른 노드는 데이터 노드 역할로 돌아갑니다.

참고 노드를 수동으로 강제 변경하여 제어 노드가 되도록 할 수 있습니다. 중앙 집중식 기능의 경우 제어 노드를 강제로 변경하면 모든 연결이 취소되며 새 제어 노드에서 연결을 다시 설정해야 합니다.

클러스터 내의 고가용성

클러스터링에서는 노드 및 인터페이스의 상태를 모니터링하고 노드 간의 연결 상태를 복제하여 고가용성을 제공합니다.

노드 상태 모니터링

각 노드는 클러스터 제어 링크를 통해 브로드 캐스트 heartbeat 패킷을 주기적으로 전송합니다. 제어 노드가 구성 가능한 시간 초과 기간 내에 데이터 유닛에서 heartbeat 패킷 또는 기타 패킷을 수신하지 않는 경우, 제어 노드는 클러스터에서 데이터 노드를 제거합니다. 데이터 노드가 제어 노드에서 패킷을 수신하지 않으면 나머지 노드에서 새 제어 노드가 선택됩니다.

네트워크 장애로 인해 노드가 실제로 장애가 발생한 것이 아니라 클러스터 제어 링크를 통해 노드가 서로 연결할 수 없는 경우, 클러스터는 격리된 데이터 노드가 자체 제어 노드를 선택하는 "스플릿 브레인" 시나리오로 전환될 수 있습니다. 예를 들어 두 클러스터 위치 간에 라우터가 실패하면 위치 1의 원래 제어 노드가 클러스터에서 위치 2 데이터 노드를 제거합니다. 한편, 위치 2의 노드는 자체 제어 노드를 선택하고 자체 클러스터를 구성합니다. 이 시나리오에서는 비대칭 트래픽이 실패할 수 있습니다. 클러스터 제어 링크가 복원되면 우선 순위가 더 높은 제어 노드가 제어 노드의 역할을 유지합니다.

인터페이스 모니터링

각 노드에서는 사용 중인 모든 명명된 하드웨어 인터페이스의 링크 상태를 모니터링하며 상태 변경 사항을 제어 노드에 보고합니다.

모든 물리적 인터페이스가 모니터링됩니다. 명명된 인터페이스만 모니터링할 수 있습니다. 선택적으로 인터페이스별 모니터링을 비활성화할 수 있습니다.

노드의 모니터링된 인터페이스에 장애가 발생하면 클러스터에서 해당 노드가 제거됩니다. 노드는 500밀리초 후에 제거됩니다.

실패 이후 상태

제어 노드에 장애가 발생할 경우, 우선순위가 가장 높은(숫자가 가장 낮은) 클러스터의 다른 멤버가 제어 노드가 됩니다.

Threat Defense는 실패 이벤트에 따라 클러스터에 다시 참가하려고 시도합니다.

참고 Threat Defense가 비활성화되고 클러스터에 자동으로 다시 조인하지 못할 경우, 모든 데이터 인터페이스가 종료되며 관리 인터페이스에서만 트래픽을 주고받을 수 있습니다.

클러스터 다시 참가

클러스터 멤버가 클러스터에서 제거된 후 해당 멤버가 클러스터에 다시 참가할 수 있는 방법은 처음에 제거된 이유에 따라 결정됩니다.

- 최초 가입 시 오류가 발생한 클러스터 제어—클러스터 제어 링크의 문제를 해결한 다음 클러스터링을 다시 활성화하여 수동으로 클러스터를 다시 가입시켜야 합니다.
- 클러스터 가입 후 클러스터 제어 링크 장애 — FTD에서는 자동으로 5분마다 무기한으로 다시 가입하려고 시도합니다.
- 데이터 인터페이스 오류 — threat defense에서는 5분에 다시 참가를 시도하며 그다음에는 10분, 마지막으로 20분에 참가를 시도합니다. 20분 후에도 참가가 이루어지지 않을 경우 threat defense에서는 클러스터링을 비활성화합니다. 데이터 인터페이스의 문제를 해결한 다음 수동으로 클러스터링을 활성화해야 합니다.
- 노드 오류 — 노드 상태 검사 오류로 인해 클러스터에서 노드가 제거된 경우, 클러스터에 다시 참가할 수 있을지 여부는 오류의 원인에 따라 결정됩니다. 예를 들어, 일시적인 정전이 발생한 경우 클러스터 제어 링크가 작동 상태이면 전원을 다시 가동할 때 노드가 클러스터에 다시 참가할 수 있습니다. threat defense 애플리케이션은 5초마다 클러스터에 다시 참가하려고 시도합니다.
- 내부 오류 — 내부 장애 포함: 애플리케이션 동기화 시간 초과, 일치하지 않는 애플리케이션 상태 등이 있습니다. 문제를 해결한 후 클러스터링을 다시 사용하도록 설정하여 클러스터에 수동으로 다시 참가해야 합니다.
- 실패한 구성 구축-FMC에서 새 구성을 구축하는 경우 일부 클러스터 멤버에서는 구축이 실패하지만 다른 클러스터 멤버에서는 성공할 경우 실패한 노드는 클러스터에서 제거됩니다. 문제를 해결한 후 클러스터링을 다시 사용하도록 설정하여 클러스터에 수동으로 다시 참가해야 합니다.

제어 노드에서 구축이 실패하면 구축이 롤백되고 멤버가 제거되지 않습니다. 모든 데이터 노드에서 구축이 실패하면 구축이 롤백되고 멤버가 제거되지 않습니다.

데이터 경로 연결 상태 복제

모든 연결마다 클러스터 내에 하나의 소유자 및 최소 하나의 백업 소유자가 있습니다. 백업 소유자는 장애 발생 시 연결을 인계받는 대신 TCP/UDP 상태 정보를 저장하므로, 장애가 발생할 경우 연결이 새로운 소유자에게 원활하게 전송될 수 있습니다. 백업 소유자는 일반적으로 관리자이기도 합니다.

일부 트래픽의 경우 TCP 또는 UDP 레이어 상위에 대한 상태 정보가 필요합니다. 클러스터링 지원에 대해 알아보거나 이러한 종류의 트래픽에 대한 지원이 부족한 경우 다음 표를 참조하십시오.

표 ***8:*** 클러스터 전반에 걸쳐 복제된 기능

트래픽	상태 지원	참고
가동 시간	예	시스템 가동 시간을 추적합니다.
ARP 테이블	예	—
MAC 주소 테이블	예	—
사용자 ID	예	—
IPv6 네이버 데이터베이스	예	—
동적 라우팅	예	—
SNMP 엔진 ID	아니요	—

클러스터에서 연결을 관리하는 방법

클러스터의 여러 노드에 대한 연결을 로드 밸런싱할 수 있습니다. 연결 역할은 정상적인 작동이 이루어지고 있고 가용성이 높은 상황에서 연결을 처리하는 방법을 결정합니다.

연결 역할

각 연결에 대해 정의된 다음 역할을 참조하십시오.

- 소유자 - 일반적으로 연결을 가장 처음 수신하는 노드입니다. 소유자 유닛에서는 TCP 상태를 유지하고 패킷을 처리합니다. 연결이 하나인 경우 소유자 유닛도 1개뿐입니다. 원래 소유자가 실패하고 새 노드가 연결에서 패킷을 수신하면, 관리자는 해당 노드로부터 새 소유자를 선택합니다.
- 백업 소유자 - 장애가 발생할 경우 연결이 새로운 소유자에게 원활하게 전송될 수 있도록 소유자로부터 수신한 TCP/UDP 상태 정보를 저장하는 노드입니다. 백업 소유자는 장애 발생 시 연결을 승계할 수 없습니다. 소유자를 사용할 수 없는 경우, 연결에서 (로드 밸런싱을 기준으로) 패킷을 받을 첫 번째 노드가 백업 소유자에 관련 상태 정보를 문의하면 해당 백업 소유자가 새로운 소유자가 될 수 있습니다.

관리자(아래 설명 참조)는 소유자와 같은 노드가 아니라면 백업 소유자로도 사용됩니다. 소유자가 자신을 디렉터로 선택하면 별도의 백업 소유자가 선택됩니다.

Firepower 9300의 클러스터링(섀시 하나에 클러스터 노드가 3개까지 포함될 수 있음)에서 백업 소유자가 소유자와 같은 섀시에 있으면 섀시 장애로부터 플로우를 보호하기 위해 다른 섀시에서 추가 백업 소유자가 선택됩니다.

- 관리자 - 전달자의 소유자 조회 요청을 처리하는 노드입니다. 소유자가 새 연결을 수신할 경우, 소유자 노드에서는 소스/대상 IP 주소와 포트의 해시를 기준으로 관리자를 선택하며 관리자에 메시지를 전송하여 새 연결을 등록합니다(아래에서 ICMP 해시 세부 정보 참조). 패킷이 소유자가 아닌 다른 노드에 전달될 경우, 해당 노드는 관리자에 어떤 노드가 소유자인지 조회하여 패킷이 전달될 수 있도록 합니다. 연결이 하나인 경우 관리자 유닛도 1개뿐입니다. 관리자가 실패하면 소유자는 새 관리자를 선택합니다.

  관리자는 소유자와 같은 노드가 아니면 백업 소유자로도 사용됩니다(위의 설명 참조). 소유자가 자신을 디렉터로 선택하면 별도의 백업 소유자가 선택됩니다.

  ICMP/ICMPv6 해시 세부 정보:

  - 에코 패킷의 경우 소스 포트는 ICMP 식별자이고, 대상 포트는 0입니다.
  - 응답 패킷의 경우 소스 포트는 0이고, 대상 포트는 ICMP 식별자입니다.
  - 기타 패킷의 경우 소스 및 대상 포트가 모두 0입니다.

- 전달자 — 패킷을 소유자에 전달하는 노드입니다. 소유하지 않은 연결 패킷이 전달자 유닛에 수신될 경우, 전달자 유닛에서는 소유자 유닛의 관리자를 조회한 다음 이러한 연결을 수신하는 기타 모든 패킷의 소유자에 대한 흐름을 설정합니다. 관리자 유닛은 전달자가 될 수도 있습니다. 전달자 유닛에서 SYN-ACK 패킷을 수신할 경우, 패킷의 SYN 쿠키에서 소유자를 직접 파생할 수 있으므로 관리자 유닛에 조회하지 않아도 됩니다. (TCP 시퀀스 임의 설정을 비활성화한 경우 SYN 쿠키는 사용되지 않으며, 책임자에게 쿼리해야 합니다.) DNS 및 ICMP 같이 짧은 흐름의 경우 쿼리 대신 전달자가 책임자에게 패킷을 즉시 전송하고 책임자가 소유자에게 전송합니다. 하나의 연결에 여러 개의 전달자 유닛이 있을 수 있습니다. 가장 효율적인 처리량 목표를 실현하려면 전달자가 없고 연결의 모든 패킷이 소유자 유닛에 전송되는 우수한 로드 밸런싱 방법을 사용합니다.

참고 클러스터링을 사용할 때는 TCP 시퀀스 임의 설정을 비활성화하지 않는 것이 좋습니다. SYN/ACK 패킷이 삭제될 수 있으므로 일부 TCP 세션이 설정되지 않을 가능성이 적습니다.

- 프래그먼트 소유자 - 프래그먼트화된 패킷의 경우 프래그먼트를 수신하는 클러스터 노드가 프래그먼트 소스 IP 주소, 대상 IP 주소 및 패킷 ID의 해시를 사용하여 프래그먼트 소유자를 결정합니다. 그런 다음 모든 프래그먼트가 클러스터 제어 링크를 통해 프래그먼트 소유자에게 전달됩니다. 첫 번째 프래그먼트만 스위치 로드 밸런싱 해시에 사용되는 5 튜플을 포함하기 때문에 프래그먼트는 다른 클러스터 노드로 로드 밸런싱될 수 있습니다. 다른 프래그먼트는 소스 및 대상 포트를 포함하지 않으며 다른 클러스터 노드에 로드 밸런싱될 수 있습니다. 프래그먼트 소유자는 패킷을 일시적으로 리어셈블하므로 소스/대상 IP 주소 및 포트의 해시를 기반으로 디렉터를

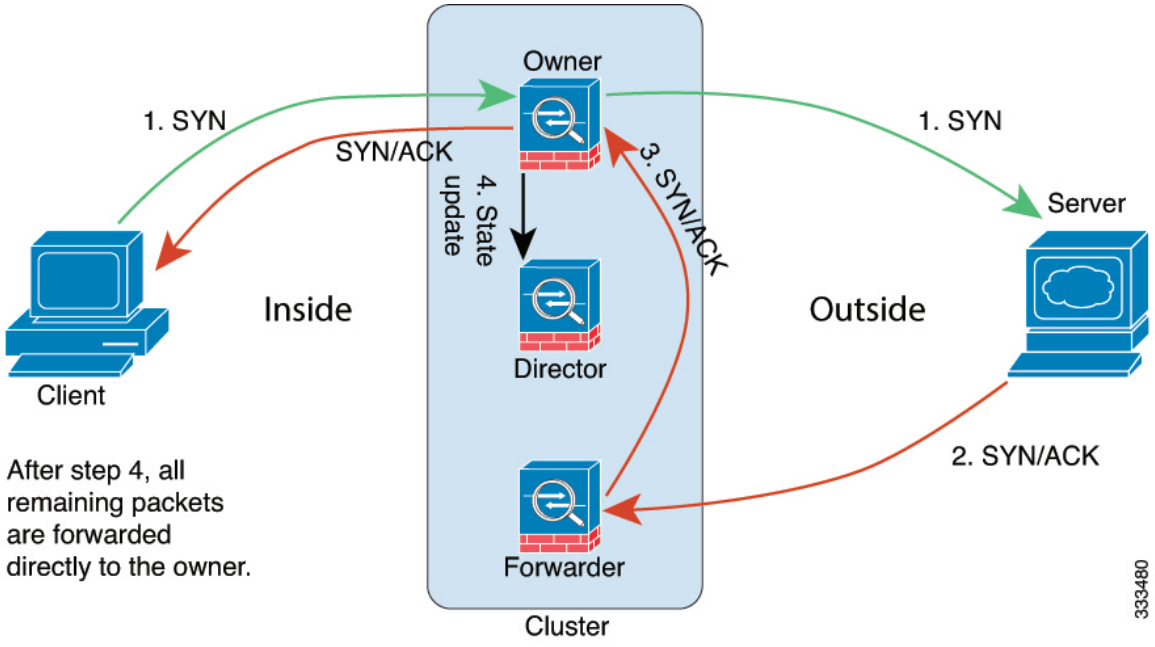
확인할 수 있습니다. 새 연결인 경우 프래그먼트 소유자가 연결 소유자로 등록됩니다. 기존 연결인 경우 프래그먼트 소유자는 클러스터 제어 링크를 통해 모든 프래그먼트를 제공된 연결 소유자에게 전달합니다. 그러면 연결 소유자가 모든 프래그먼트를 리어셈블합니다.

새 연결 소유권

로드 밸런싱을 통해 클러스터의 노드에 새 연결이 전송될 경우, 해당 노드에서는 연결의 양방향을 모두 소유합니다. 다른 노드에 연결 패킷이 전송될 경우, 해당 패킷은 클러스터 제어 링크를 통해 소유자 노드에 전달됩니다. 다른 노드에 반대 방향의 흐름이 전송될 경우, 이는 원래 노드로 다시 리디렉션됩니다.

TCP에 대한 샘플 데이터 플로우

다음 예에는 새 연결을 설정하는 방법이 나와 있습니다.



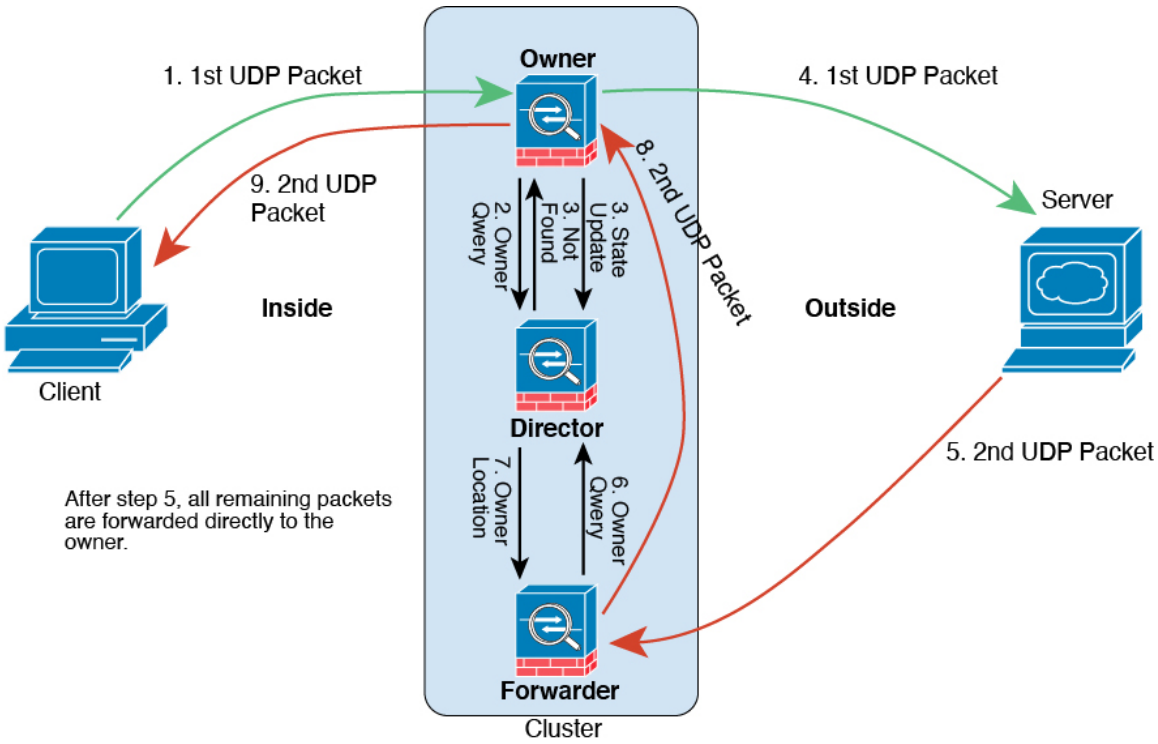
1. SYN 패킷은 클라이언트에서 시작되고 threat defense에 전달(로드 밸런싱 방법을 기준으로)되며, 이 유닛이 소유자 유닛이 됩니다. 소유자 유닛에서는 흐름을 생성하고, 소유자 정보를 SYN 쿠키로 인코딩하며, 패킷을 서버에 전달합니다.
2. SYN-ACK 패킷은 서버에서 시작되고 다른 threat defense에 전달(로드 밸런싱 방법을 기준으로)됩니다. 이 threat defense는 전달자 유닛입니다.
3. 전달자 유닛에서는 연결을 소유하지 않으므로 SYN 쿠키에서 소유자 정보를 디코딩하고, 소유자에 대한 전달 흐름을 생성하며, SYN-ACK를 소유자 유닛에 전달합니다.
4. 소유자 유닛에서는 관리자 유닛에 상태 업데이트를 보내고, SYN-ACK를 클라이언트에 전달합니다.

5. 관리자 유닛에서는 소유자 유닛을 통해 상태 업데이트를 수신하고, 소유자에 대한 흐름을 생성하며, TCP 상태 정보는 물론 소유자를 기록합니다. 관리자 유닛은 연결의 백업 소유자 역할을 수행합니다.
6. 전달자 유닛에 전달된 모든 후속 패킷은 소유자 유닛에 전달됩니다.
7. 패킷이 추가 노드에 전달된 경우, 관리자에 쿼리하고 플로우를 설정합니다.
8. 플로우 결과의 상태가 변경되면 소유자 유닛과 관리자 유닛의 상태도 업데이트됩니다.

ICMP 및 **UDP**의 샘플 데이터 플로우

다음 예에는 새 연결을 설정하는 방법이 나와 있습니다.

1. 그림 ***30: ICMP*** 및 ***UDP*** 데이터 플로우



첫 번째 UDP 패킷은 클라이언트에서 시작되고 (로드 밸런싱 방법을 기준으로) threat defense에 전달됩니다.

2. 첫 번째 패킷을 수신한 노드는 소스/대상 IP 주소 및 포트의 해시를 기반으로 선택된 관리자 노드에 쿼리합니다.
3. 관리자는 기존 플로우를 찾지 못하고 관리자 플로우를 생성하며 이전 노드로 패킷을 다시 전달합니다. 즉, 관리자가 이 플로우의 소유자를 선택했습니다.
4. 소유자가 플로우를 생성하고 관리자에게 상태 업데이트를 보내고 서버에 패킷을 전달합니다.
5. 두 번째 UDP 패킷은 서버에서 시작되어 전달자에게 전달됩니다.

6. 전달자는 관리자에게 소유권 정보를 쿼리합니다. DNS와 같이 짧은 플로우의 경우 쿼리하는 대신 전달자가 관리자에게 패킷을 즉시 전송하고 관리자가 소유자에게 전송합니다.
7. 관리자는 전달자에게 소유권 정보를 회신합니다.
8. 전달자는 전달 플로우를 생성하여 소유자 정보를 기록하고 소유자에게 패킷을 전달합니다.
9. 소유자는 패킷을 클라이언트에 전달합니다.

퍼블릭 클라우드의 **Threat Defense Virtual** 클러스터링 기록

기능	최소 **Management Center**	최소 **Threat Defense**	세부 사항
클러스터 제어 링크 ping 도구.	7.4.1	Any(모든)	ping을 수행하여 모든 클러스터 노드가 클러스터 제어 링크를 통해 서로 연결 가능한지 확인할 수 있습니다. 노드가 클러스터에 조인하지 못하는 주요 원인 중 하나는 잘못된 클러스터 제어 링크 구성입니다. 예를 들어, 클러스터 제어 링크 MTU가 연결 스위치 MTU보다 높게 설정될 수 있습니다. 신규/수정된 화면: **Devices(**디바이스**)** > **Device Management(**디바이스 관리**)** > 추가 (⋮) > **Cluster Live Status(**클러스터 라이브 상태**)** 기타 버전 제한: Management Center 버전 7.3.x 또는 7.4.0에서는 지원되지 않습니다.
Device(디바이스) 및 Cluster(클러스터) 페이지에서 지원되는 문제 해결 파일 생성 및 다운로드.	7.4.1	7.4.1	Device(디바이스) 페이지에서 각 디바이스에 대한 문제 해결 파일을 생성하고 다운로드할 수 있으며, Cluster(클러스터) 페이지에서 모든 클러스터 노드에 대한 문제 해결 파일을 생성하고 다운로드할 수 있습니다. 클러스터의 경우, 모든 파일을 단일 압축 파일로 다운로드할 수 있습니다. 또한 클러스터 노드의 클러스터에 대한 클러스터 로그를 포함할 수 있습니다. **Devices(**디바이스**)** > **Device Management(**디바이스 관리**)** > 추가 (⋮) > **Troubleshoot Files(**문제 해결 파일**)** 메뉴에서 파일 생성을 트리거하는 방법도 있습니다. 신규/수정된 화면: • **Devices(**디바이스**)** > **Device Management(**디바이스 관리**)** > **Device(**디바이스**)** > **General(**일반**)** • **Devices(**디바이스**)** > **Device Management(**디바이스 관리**)** > **Cluster(**클러스터**)** > **General(**일반**)**

기능	최소 **Management Center**	최소 **Threat Defense**	세부 사항
디바이스 또는 디바이스 클러스터에 대한 CLI 출력 조회.	7.4.1	Any(모든)	디바이스 또는 클러스터의 문제를 해결하는 데 도움이 되는 사전 정의된 CLI 출력 집합을 볼 수 있습니다. 또한 **show** 명령을 입력하여 출력을 확인할 수 있습니다. 신규/수정된 화면: **Devices(**디바이스**)** > **Device Management(**디바이스 관리**)** > **Cluster(**클러스터**)** > **General(**일반**)**
클러스터 상태 모니터링 설정	7.3.0	Any(모든)	이제 클러스터 상태 모니터링 설정을 편집할 수 있습니다. 신규/수정된 화면: **Devices(**디바이스**)** > **Device Management(**디바이스 관리**)** > **Cluster(**클러스터**)** > **Cluster Health Monitor Settings(**클러스터 상태 모니터링 설정**)** 참고 이전에 FlexConfig를 사용하여 이러한 설정을 구성한 경우 구축하기 전에 FlexConfig 구성을 제거해야 합니다. 그렇지 않으면 FlexConfig 구성이 Management Center 구성을 덮어씁니다.
클러스터 상태 모니터 대시보드	7.3.0	Any(모든)	이제 클러스터 상태 모니터 대시보드에서 클러스터 상태를 볼 수 있습니다. 신규/수정된 화면: 시스템 (⚙) > **Health(**상태**)** > **Monitor(**모니터**)**
Azure에서 threat defense virtual에 대한 클러스터링	7.3.0	7.3.0	이제 Azure 게이트웨이 로드 밸런서 또는 외부 로드 밸런서에 대해 Azure의 threat defense virtual에서 최대 16개의 노드에 대해 클러스터링을 구성할 수 있습니다. 신규/수정된 화면: • **Devices(**디바이스**)** > **Device Management(**디바이스 관리**)** > **Add Cluster(**클러스터 추가**)** • **Devices(**디바이스**)** > **Device Management(**디바이스 관리**)** > **More(**더 보기**)** 메뉴 • **Devices(**디바이스**)** > **Device Management(**디바이스 관리**)** > **Cluster(**클러스터**)** 지원되는 플랫폼: Azure의 Threat Defense Virtual

기능	최소 **Management Center**	최소 **Threat Defense**	세부 사항
퍼블릭 클라우드(Amazon Web Services 및 Google Cloud Platform)에서의 Threat Defense Virtual 클러스터링	7.2.0	7.2.0	threat defense virtual는 퍼블릭 클라우드(AWS 및 GCP)에서 최대 16개의 노드에 대한 개별 인터페이스 클러스터링을 지원합니다. 신규/수정된 화면: • **Devices(**디바이스**)** > **Device Management(**디바이스 관리**)** > **Add Device(**디바이스추가**)** • **Devices(**디바이스**)** > **Device Management(**디바이스 관리**)** > **More(**더 보기**)** 메뉴 • **Devices(**디바이스**)** > **Device Management(**디바이스 관리**)** > **Cluster(**클러스터**)** 지원되는 플랫폼: AWS 및 GCP의 Threat Defense Virtual