

# 퍼블릭 클라우드에서 **ASA** 가상용 클러스터 구축

초판: 2023년 12월 13일

## 퍼블릭 클라우드에서 **ASA** 가상용 클러스터 구축

클러스터링을 사용하면 여러 개의 ASA 가상을 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 클라우드를 사용하여 퍼블릭 클라우드에서 ASA 가상 클러스터를 구축할 수 있습니다.

- ASA 9.19 이상이 설치된 AWS(Amazon Web Services)
- ASA 9.20(2) 이상이 설치된 Microsoft Azure

현재는 라우팅 방화벽 모드만 지원됩니다.



참고 클러스터링을 사용할 경우 일부 기능이 지원되지 않습니다. [클러스터링으로 지원되지 않는 기능, 53 페이지](#)의 내용을 참조하십시오.

## 퍼블릭 클라우드에서 **ASA** 가상 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 ASA 가상에서 브로드캐스트/멀티캐스트 메시지를 전송하도록 합니다.
- Load Balancer(로드 밸런서) - 외부 로드 밸런싱의 경우 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. ASA 가상은 Geneve

인터페이스 단일 암 프록시를 사용하여 분산형 데이터 플레인(게이트웨이 로드 밸런서 엔드포인트)이 있는 게이트웨이 로드 밸런서 중앙 집중식 제어 평면을 지원합니다.

- 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, ASA 가상 오류가 발생하면 문제를 초래할 수 있습니다. 경로는 계속 사용할 수 있으며 오류가 발생한 ASA 가상에 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 ASA 가상을 구성해야 합니다.



참고 레이어 2 스패 EtherChannel은 로드 밸런싱에 지원되지 않습니다.

## 클러스터 노드

클러스터 노드는 보안 정책 및 트래픽 흐름을 공유하기 위해 서로 연동됩니다. 이 섹션에서는 각 노드 역할의 특성을 설명합니다.

### 부트스트랩 컨피그레이션

각 디바이스에서 클러스터 이름, 클러스터 제어 링크 인터페이스, 기타 클러스터 설정 등을 비롯한 최소 부트스트랩 컨피그레이션을 구성합니다. 클러스터링을 활성화한 첫 번째 노드가 일반적으로 제어 노드가 됩니다. 후속 노드에서 클러스터링을 사용하도록 설정할 경우, 해당 노드는 클러스터에 슬레이브로 참가합니다.

### 제어 및 데이터 노드 역할

클러스터의 멤버 중 하나는 제어 노드입니다. 여러 클러스터 노드가 동시에 온라인 상태가 되면 부트스트랩 구성의 우선 순위 설정에 따라 제어 노드가 결정됩니다. 우선 순위는 1에서 100까지 1이 가장 높은 우선 순위입니다. 다른 모든 멤버는 데이터 노드입니다. 일반적으로 클러스터를 처음 생성할 때 추가하는 첫 번째 노드는 현재까지 클러스터의 유일한 노드이기 때문에 제어 노드가 됩니다.

제어 노드에서만 모든 구성(부트스트랩 구성 제외)을 수행해야 하며, 그 후 이러한 구성은 데이터 노드에 복제됩니다. 인터페이스와 같은 물리적 자산의 경우 제어 노드의 구성은 모든 데이터 노드에 미러링됩니다. 예를 들어 Ethernet 1/2를 내부 인터페이스로 구성하고 Ethernet 1/1을 외부 인터페이스로 구성하는 경우, 이러한 인터페이스는 데이터 노드에서도 내부 및 외부 인터페이스로 사용됩니다.

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

## 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.

## 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 ASA 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 ASA 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

## 클러스터 제어 링크 오류

유닛의 클러스터 제어 링크 라인 프로토콜이 작동되지 않을 경우, 클러스터링을 사용할 수 없게 되며 데이터 인터페이스가 종료됩니다. 클러스터 제어 링크를 해결한 후 클러스터링을 다시 사용하도록 설정하여 클러스터에 수동으로 다시 참가해야 합니다.



**참고** ASA 가상이 비활성화되면 모든 데이터 인터페이스가 종료되며 관리 전용 인터페이스에서만 트래픽을 주고받을 수 있습니다. 관리 인터페이스에서는 DHCP 또는 클러스터 IP 풀에서 유닛으로 전송된 IP 주소를 사용하여 가동 상태를 유지합니다. 클러스터 IP 풀을 사용하는 경우 다시 로드한 후에도 유닛이 클러스터 내에서 비활성 상태일 경우, 관리 인터페이스에서는 제어 노드와 동일한 기본 IP 주소를 사용하므로 관리 인터페이스에 액세스할 수 없습니다. 추가 구성을 위해서는 콘솔 포트(가능한 경우)를 사용해야 합니다.

## 구성 복제

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

## ASA 가상클러스터 관리

ASA 가상 클러스터링을 사용하는 데 따른 여러 장점 중 하나는 관리하기가 쉽다는 점입니다. 이 섹션에서는 클러스터를 관리하는 방법에 대해 설명합니다.

### 관리 네트워크

모든 노드를 단일한 관리 네트워크에 연결할 것을 권장합니다. 이 네트워크는 클러스터 제어 링크와 분리되어 있습니다.

### 관리 인터페이스

관리에 Management 0/0 인터페이스를 사용합니다.



**참고** 관리 인터페이스에 대해서는 동적 라우팅을 활성화할 수 없습니다. 고정 경로를 사용해야 합니다.

관리 IP 주소로 고정 주소 또는 DHCP를 사용할 수 있습니다.



정적 주소 지정을 사용하는 경우 항상 현재 제어 노드에 속하는 클러스터의 고정 주소인 주 클러스터 IP 주소를 사용할 수 있습니다. 각 인터페이스에는 주소의 범위를 구성하여 현재 제어 노드를 비롯한 각 노드에서 해당 범위의 로컬 주소를 사용할 수 있도록 합니다. 기본 클러스터 IP 주소에서는 주소에 대한 일관된 관리 액세스를 제공합니다. 제어 노드가 변경될 경우 기본 클러스터 IP 주소는 새 제어 노드로 이동되므로 클러스터는 지속적으로 원활하게 관리됩니다. 로컬 IP 주소는 라우팅에 사용되며 트러블슈팅에도 도움이 됩니다. 예를 들어, 현재 제어 노드에 항상 연결되어 있는 기본 클러스터 IP 주소에 연결하여 클러스터를 관리할 수 있습니다. 로컬 IP 주소에 연결하여 개별 멤버를 관리할 수 있습니다. TFTP 또는 시스템 로그 같은 아웃바운드 관리 트래픽의 경우, 제어 노드를 비롯한 각 노드에서는 로컬 IP 주소를 사용하여 서버에 연결합니다.

DHCP를 사용하는 경우에는 로컬 주소의 풀을 사용하거나 기본 클러스터 IP 주소가 없습니다.

## 제어 노드 관리 대 데이터 노드 관리

모든 관리 및 모니터링은 제어 노드에서 수행할 수 있습니다. 제어 노드에서 런타임 통계, 리소스 사용량 또는 모든 노드의 기타 모니터링 정보를 확인할 수 있습니다. 또한 클러스터 내의 모든 노드에 명령을 배포하고, 데이터 노드의 콘솔 메시지를 제어 노드로 복제할 수 있습니다.

필요한 경우 데이터 노드를 직접 모니터링할 수 있습니다. 제어 노드에서도 사용 가능하지만 데이터 노드에서 파일 관리를 수행할 수 있습니다(구성 백업 및 이미지 업데이트 포함). 제어 노드에서는 다음 기능을 사용할 수 없습니다.

- 노드당 클러스터별 통계 모니터링.
- 노드당 Syslog 모니터링(콘솔 복제가 활성화된 경우 콘솔로 전송되는 syslog 제외).
- SNMP
- NetFlow

## 암호화 키 복제

제어 노드에서 암호화 키를 생성할 경우, 해당 키는 모든 데이터 노드에 복제됩니다. 기본 클러스터 IP 주소에 대한 SSH 세션이 있는 경우 제어 노드에 오류가 발생하면 연결이 끊어집니다. 새 제어 노드에서는 SSH 연결에 동일한 키를 사용하므로, 새 제어 노드에 다시 연결할 때 캐시된 SSH 호스트 키를 업데이트하지 않아도 됩니다.

## ASDM 연결 인증서 IP 주소 불일치

기본적으로, 자체 서명된 인증서는 로컬 IP 주소를 기준으로 ASDM 연결에 사용됩니다. ASDM을 사용하여 기본 클러스터 IP 주소를 연결할 경우, 인증서에서는 기본 클러스터 IP 주소가 아닌 로컬 IP 주소를 사용하므로 IP 주소가 일치하지 않는다는 경고 메시지가 표시됩니다. 이 메시지를 무시하고 ASDM 연결을 설정할 수 있습니다. 그러나 이러한 유형의 경고를 방지하려면 기본 클러스터 IP 주소 및 IP 주소 풀의 모든 로컬 IP 주소가 포함된 인증서를 등록하면 됩니다. 그런 다음 이 인증서를 각 클러스터 멤버에 사용할 수 있습니다. 자세한 내용은 <https://www.cisco.com/c/en/us/td/docs/security/asdm/identity-cert/cert-install.html>를 참조하십시오.

## ASA 가상 클러스터링용 라이선스

각 클러스터 노드에는 동일한 모델 라이선스가 필요합니다. 모든 노드에 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.



**참고** ASA 가상을 등록 해제하여 라이선스가 없는 경우, ASA 가상을 다시 로드하면 심각한 속도 제한 상태로 전환됩니다. 라이선스가 없는 성능이 낮은 클러스터 노드는 전체 클러스터의 성능에 부정적인 영향을 미칩니다. 모든 클러스터 노드를 라이선스로 유지하거나 라이선스가 없는 노드를 제거해야 합니다.

## ASA 가상 클러스터링의 요구 사항 및 사전 요건

### 모델 요구 사항

- ASAv30, ASAv50, ASAv100
- 다음 퍼블릭 클라우드 서비스:
  - ASA 9.19 이상이 설치된 AWS(Amazon Web Services)
  - ASA 9.20(2) 이상이 설치된 Microsoft Azure
- 최대 16개 노드

[ASA 가상 시작 가이드](#)에서 ASA 가상에 대한 일반 요구 사항도 참조하십시오.

### 하드웨어 및 소프트웨어 요건

#### 클러스터의 모든 노드:

- 동일한 성능 계층이어야 합니다. 모든 노드에 대해 동일한 수의 CPU 및 메모리를 사용하는 것이 좋습니다. 그렇지 않으면 성능이 가장 낮은 노드와 일치하도록 모든 노드에서 제한됩니다.
- 이미지 업그레이드 시간을 제외하고는 동일한 소프트웨어를 실행해야 합니다. 무중단 업그레이드가 지원됩니다. 소프트웨어 버전이 일치하지 않으면 성능이 저하될 수 있으므로 동일한 유지 관리 기간에 모든 노드를 업그레이드해야 합니다.
- 단일 가용 영역 구축이 지원됩니다.
- 클러스터 제어 링크 인터페이스는 동일한 서브넷에 있어야 하므로 클러스터를 동일한 서브넷에 구축해야 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. 소스 인터페이스 MTU를 네트워크 MTU + 54바이트로 설정해야 합니다.

다음 표는 권장되는 클러스터 제어 링크 MTU 및 데이터 인터페이스 MTU를 보여줍니다.

표 1: 권장 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          |

## ASA 가상 클러스터링 가이드라인

### 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, ASA 가상 또는 스위치의 인터페이스 활성화 또는 비활성화, 추가 스위치를 추가하여 이중화 스위치 시스템 구성)이 발생할 경우 상태 검사 기능을 비활성화하고 비활성화된 인터페이스에 대한 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.
- 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.
- 각 유지 보수 기간이 완료된 후 전역 구축을 수행합니다.
- Auto Scale 그룹(AWS) 또는 확장 세트(Azure)에서 한 번에 둘 이상의 디바이스를 제거하지 않아야 합니다. 또한 Auto Scale 그룹(AWS) 또는 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 Day 0 구성 스크립트를 사용하여 요구 사항에 따라 IP 주소를 변경하고, 맞춤형 인터페이스 이름을 제공하고, CCL-Link 인터페이스의 시퀀스를 변경할 수 있습니다.

### 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.

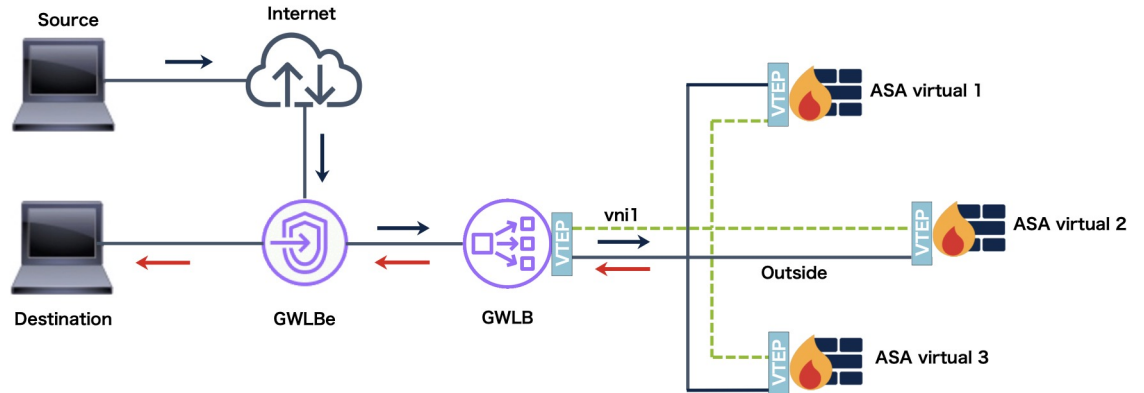
## AWS 게이트웨이 로드 밸런서 및 Geneve 단일 암 프록시



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. ASA 가상은 분산형 데이터 플레인(게이트웨이 로드 밸런서 엔드포인트)이 있는 게이트웨이 로드 밸런서 중앙 집중식 제어 평면을 지원합니다. 다음 그림에는 게이트웨이 로드 밸런서 엔드포인트에서 게이트웨이 로드 밸런서로 전달되는 트래픽이 나와 있습니다. 게이트웨이 로드 밸런서는 여러 ASA 가상 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.

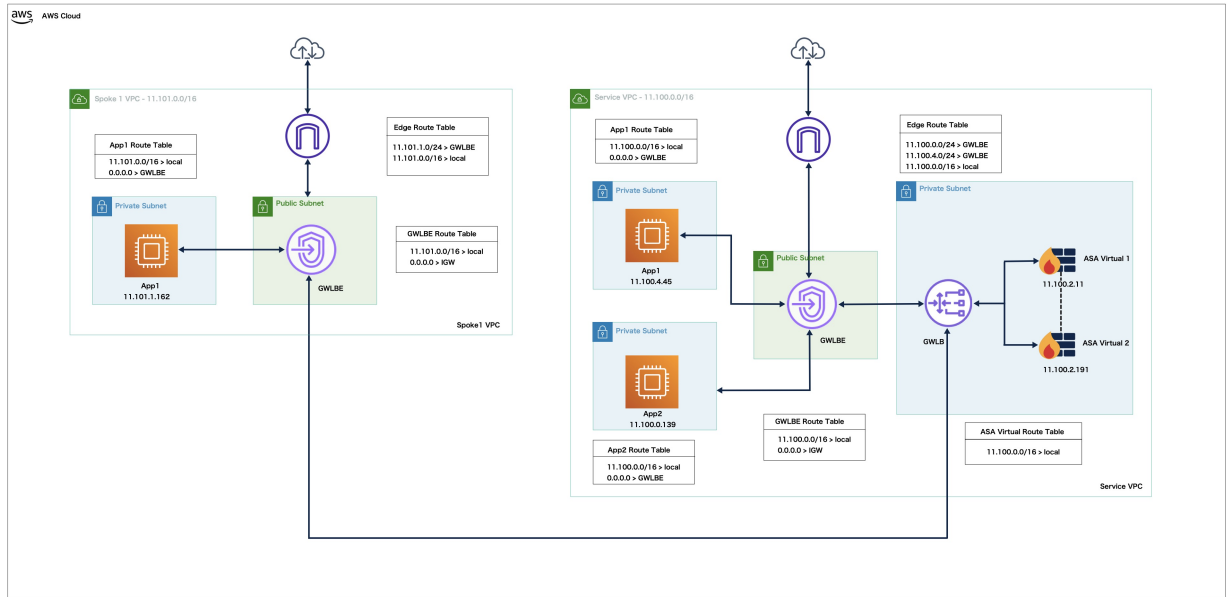
그림 1: Geneve 단일 암 프록시



### 샘플 토폴로지

아래 토폴로지에는 인바운드 및 아웃바운드 트래픽 플로우가 모두 나와 있습니다. GWLB에 연결된 클러스터에는 ASA 가상 인스턴스 3개가 있습니다.

AWP에서 ASA 가상 클러스터를 구축하기 위한 End-to-End 프로세스



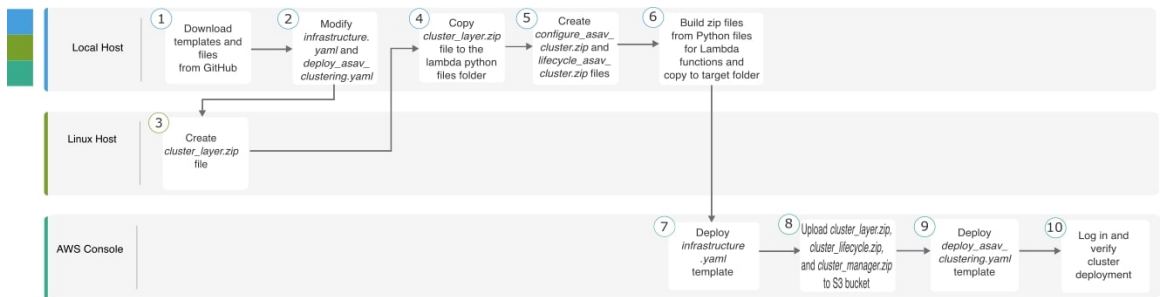
인터넷의 인바운드 트래픽은 GWLB 엔드포인트로 이동한 다음 이 엔드포인트에서 GWLB로 트래픽을 전송합니다. 그런 다음 트래픽은 ASA 가상 클러스터로 전달됩니다. 클러스터의 ASA 가상 인스턴스에서 검사된 트래픽은 애플리케이션 VM, App1로 전달됩니다.

App1의 아웃바운드 트래픽은 GWLB 엔드포인트로 전송된 다음 인터넷으로 전송됩니다.

AWP에서 ASA 가상 클러스터를 구축하기 위한 End-to-End 프로세스

템플릿 기반 구축

다음 순서도에는 AWS에서 ASA 가상 클러스터를 템플릿 기반으로 구축하는 워크플로우가 나와 있습니다.

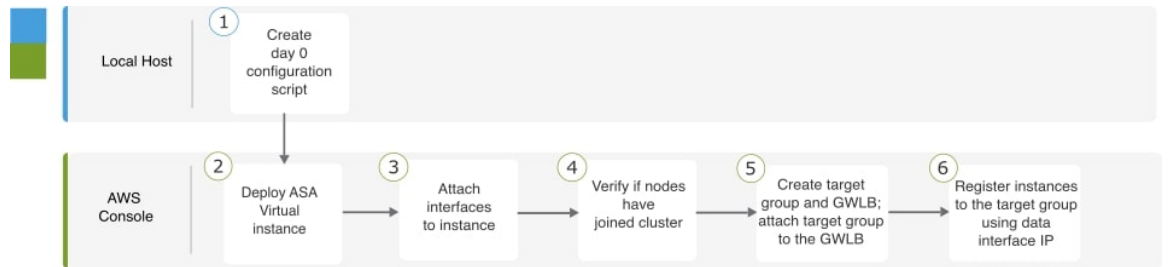


|   | 업무 환경  | 단계   |
|---|--------|--|
| 1 | 로컬 호스트 | GitHub에서 템플릿 및 파일을 다운로드합니다.                                      |
| 2 | 로컬 호스트 | infrastructure.yaml 및 deploying_asav_clustering.yaml 템플릿을 수정합니다. |

|    | 업무 환경     | 단계  |
|----|-----------|---|
| 3  | Linux 호스트 | <i>cluster_layer.zip</i> 파일을 생성합니다.   |
| 4  | 로컬 호스트    | <i>cluster_layer.zip</i> 파일을 Lambda python files 폴더에 복사합니다.   |
| 5  | 로컬 호스트    | <i>configure_asav_cluster.zip</i> 및 <i>Lifecycle_asav_cluster.zip</i> 파일을 생성합니다.                      |
| 6  | 로컬 호스트    | Lambda 함수용 Python 파일에서 zip 파일을 빌드하고 대상 폴더에 복사합니다.   |
| 7  | AWS 콘솔    | <i>infrastructure.yaml</i> 템플릿을 구축합니다.  |
| 8  | AWS 콘솔    | <i>cluster_layer.zip</i> , <i>cluster_lifecycle.zip</i> 및 <i>cluster_manager.zip</i> 을 S3 버킷에 업로드합니다. |
| 9  | AWS 콘솔    | <i>deploy_asav_clustering.yaml</i> 템플릿을 구축합니다.  |
| 10 | AWS 콘솔    | 로그인하여 클러스터 구축을 확인합니다.   |

수동 구축

다음 순서도에는 AWS에서 ASA 가상 클러스터를 수동으로 구축하는 워크플로우가 나와 있습니다.



|   | 업무 환경  | 단계                                       |
|---|--------|--|
| 1 | 로컬 호스트 | AWS에 대한 Day0 구성 생성                       |
| 2 | AWS 콘솔 | ASA 가상 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. |

|   | 업무 환경  | 단계                                     |
|---|--------|--|
| 6 | AWS 콘솔 | 데이터 인터페이스 IP를 사용하여 대상 그룹에 인스턴스를 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\_asav\_clustering.yaml* 템플릿의 InstanceType(인스턴스 유형) 매개변수에 허용되는 값 아래에서 찾을 수 있습니다.

## AWS에서 GWLB를 사용하는 ASA 가상 클러스터링을 위한 대상 페일오버 구성

AWS의 ASA 가상 클러스터링은 GWLB(게이트웨이 로드 밸런서)를 사용하여 검사를 위해 네트워크 패킷의 균형을 유지하고 지정된 ASA 가상 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.



**참고** GWLB가 SSH, SCP, CURL 또는 기타 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(Elastic Load Balancing) 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 [TargetGroupAttribute](#)를 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 ASA 가상에 대한 대상 페일 오버 활성화](#)를 참조하십시오.

## CloudFormation 템플릿을 사용하여 AWS에서 스택 구축

사용자 지정된 CloudFormation 템플릿을 사용하여 AWS에 스택을 구축합니다.

시작하기 전에

- Python 3이 설치된 Linux 컴퓨터가 필요합니다.

### 프로시저

단계 1 템플릿을 준비합니다.

- 로컬 폴더에 github 리포지토리를 복제합니다. <https://github.com/CiscoDevNet/cisco-asav/tree/master/cluster/aws>을 참조하십시오.
- 필수 매개변수를 사용하여 **infrastructure.yaml** 및 **deploy\_asav\_clustering.yaml**를 수정합니다.
- 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
zip
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.

- d) 결과 **cluster\_layer.zip** 파일을 **lambda python files** 폴더에 복사합니다.  
 e) **configure\_asav\_cluster.zip** 및 **Lifecycle\_asav\_cluster.zip** 파일 생성

**make.py** 파일은 복제된 리포지토리의 최상위 디렉토리에 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.

- AWS 콘솔에서 **CloudFormation**으로 이동하여 **Create stack**(스택 생성)을 클릭합니다. **With new resources (standard)**(새 리소스 포함(표준))를 선택합니다.
- Upload a template file**(템플릿 파일 업로드)을 선택하고 **Choose file**(파일 선택)을 클릭한 후 대상 폴더에서 **Infrastructure.yaml**을 선택합니다.
- Next**(다음)를 클릭하고 필수 정보를 제공합니다.
- Next**(다음), **Create stack**(스택 생성)을 차례로 클릭합니다.
- 구축이 완료되면 **Outputs**(출력)로 이동하여 **S3 BucketName**을 확인합니다.

그림 2: *Infrastructure.yaml*의 출력

| Outputs (13)                                |   |  |             |  |
|---|---|--|-------------|--|
| <input type="text" value="Search outputs"/> |   |  |             |  |
| Key   | Value   | Description  | Export name |  |
| AZ  | sa-east-1a  | Availability zone                                    | -           |  |
| BucketName                                  | ran-cls-infra-s3bucketcluster-kckr7518u00l  | Name of the Amazon S3 bucket                         | -           |  |
| BucketUrl                                   | <a href="http://ran-cls-infra-s3bucketcluster-kckr7518u00l.s3-website-sa-east-1.amazonaws.com">http://ran-cls-infra-s3bucketcluster-kckr7518u00l.s3-website-sa-east-1.amazonaws.com</a> | URL of S3 Bucket Static Website                      | -           |  |
| CCLSubnetId                                 | subnet-050feb347e57eba99  | CCL subnet ID  | -           |  |
| EIPforNATgw                                 | 52.67.246.95  | EIP reserved for NAT GW                              | -           |  |
| InInterfaceSGId                             | sg-0333e92f36b2aa0bf  | Security Group ID for Instances Inside Interface     | -           |  |
| InsideSubnetIds                             | subnet-047c0a2beffb5a70f  | Inside subnet ID                                     | -           |  |
| InstanceSGId                                | sg-0c0c6bfb5ba5f1c10  | Security Group ID for Instances Management Interface | -           |  |
| LambdaSecurityGroupId                       | sg-01771b0d3012a40c5  | Security Group ID for Lambda Functions               | -           |  |
| LambdaSubnetIds                             | subnet-0fb24785c687d50e4,subnet-0f1996a02ffaa2e62   | List of lambda subnet IDs (comma seperated)          | -           |  |
| MgmtSubnetIds                               | subnet-02d4a757b95a9a5b9  | Mangement subnet ID                                  | -           |  |
| UseGWLB                                     | Yes   | Use Gateway Load Balancer                            | -           |  |
| VpcName                                     | vpc-003b592ad2518d03d   | Name of the VPC created                              | -           |  |

단계 3 cluster\_layer.zip, cluster\_lifecycle.zip 및 cluster\_manager.zip을 *infrastructure.yaml*에서 생성한 S3 버킷에 업로드합니다.

단계 4 *deploy\_asav\_clustering.yaml*을 구축합니다.

- CloudFormation로 이동하고 **Create stack**(스택 생성)을 클릭합니다. **With new resources (standard)**(새 리소스 포함(표준))를 선택합니다.
- Upload a template file**(템플릿 파일 업로드)을 선택하고 **Choose file**(파일 선택)을 클릭한 후 대상 폴더에서 *deploy\_asav\_clustering.yaml*을 선택합니다.
- Next**(다음)를 클릭하고 필수 정보를 제공합니다.
- Next**(다음), **Create stack**(스택 생성)을 차례로 클릭합니다.

그림 3: 구축된 리소스

| Resources (21)                                |  |                                    |                   |               |  |
|---|--|------------------------------------|-------------------|---------------|--|
| <input type="text" value="Search resources"/> |  |                                    |                   |               |  |
| Logical ID ▲                                  | Physical ID ▼  | Type ▼                             | Status ▼          | Status reason |  |
| ASAvGroup                                     | ran-cls-1 <a href="#">↗</a>  | AWS::AutoScaling::AutoScalingGroup | ✔ CREATE_COMPLETE | -             |  |
| ASAvLaunchTemplate                            | lt-056fd20764270c893 <a href="#">↗</a>   | AWS::EC2::LaunchTemplate           | ✔ CREATE_COMPLETE | -             |  |
| CLSMangerTopic                                | arn:aws:sns:sa-east-1:797661843114:ran-cls-1-cluster-manager-topic <a href="#">↗</a>                   | AWS::SNS::Topic                    | ✔ CREATE_COMPLETE | -             |  |
| ClusterManager                                | ran-cls-1-manager-lambda <a href="#">↗</a>   | AWS::Lambda::Function              | ✔ CREATE_COMPLETE | -             |  |
| ClusterManagerLogGrp                          | /aws/lambda/ran-cls-1-manager-lambda <a href="#">↗</a>   | AWS::Logs::LogGroup                | ✔ CREATE_COMPLETE | -             |  |
| ClusterManagerSNS1                            | arn:aws:sns:sa-east-1:797661843114:ran-cls-1-cluster-manager-topic:e13bfc0-d698-4215-88a5-278474e22c32 | AWS::SNS::Subscription             | ✔ CREATE_COMPLETE | -             |  |
| ClusterManagerSNS1Permission                  | ran-cls-ClusterManagerSNS1Permission-S6BQAE05OG6U  | AWS::Lambda::Permission            | ✔ CREATE_COMPLETE | -             |  |
| InstanceEvent                                 | ran-cls-1-notify-instance-event <a href="#">↗</a>  | AWS::Events::Rule                  | ✔ CREATE_COMPLETE | -             |  |
| InstanceEventInvokeLambdaPermission           | ran-cls-InstanceEventInvokeLambdaPermission-1XP521Q4G2DY6  | AWS::Lambda::Permission            | ✔ CREATE_COMPLETE | -             |  |

상태가 **CREATE\_IN\_PROGRESS**에서 **CREATE COMPLETE**로 변경되어 구축이 성공했음을 나타냅니다.

단계 5 노트 중 하나에 로그인하고 **show cluster info** 명령을 입력하여 클러스터 구축을 확인합니다.

**show cluster info**

```
Cluster oneclicktest-cluster: On
Interface mode: individual
Cluster Member Limit : 16
This is "200" in state CONTROL_NODE
ID : 0
Version : 9.19.1
Serial No.: 9AU42EN5D1E
CCL IP : 1.1.1.200
CCL MAC : 4201.0a0a.0fc7
Module : ASAv
Resource : 4 cores / 8192 MB RAM
Last join : 15:26:22 UTC Jul 17 2022
Last leave: N/A
Other members in the cluster:
Unit "204" in state DATA_NODE
ID : 1
Version : 9.19.1
Serial No.: 9AJ9N46947R
CCL IP : 1.1.1.204
CCL MAC : 4201.0a0a.0fcb
Module : ASAv
Resource : 4 cores / 8192 MB RAM
```

Last join : 16:57:42 UTC Jul 17 2022  
 Last leave: 16:03:25 UTC Jul 17 2022

## AWS에서 수동으로 클러스터 구축

클러스터를 수동으로 구축하려면 Day0 구성을 준비하고 각 노드를 구축합니다.

### AWS에 대한 Day0 구성 생성

아래의 명령을 사용하여 각 클러스터 노드의 부트스트랩 구성을 제공합니다.

게이트웨이 로드 밸런서 예

다음 실행 구성 예는 U-turn 트래픽용 Geneve 인터페이스 1개 및 클러스터 제어 링크용 VXLAN 인터페이스 1개가 있는 게이트웨이 로드 밸런서에 대한 구성을 생성합니다.

```
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

int m0/0
management-only
nameif management
security-level 100
ip address dhcp setroute
no shut

interface TenGigabitEthernet0/0
nameif geneve-vtep-ifc
security-level 0
ip address dhcp
no shutdown

interface TenGigabitEthernet0/1
nve-only cluster
nameif ccl_link
security-level 0
ip address dhcp
no shutdown

interface vni1
description Clustering Interface
segment-id 1
vtep-nve 1

interface vni2
proxy single-arm
nameif ge
security-level 0
vtep-nve 2

object network ccl_link
range 10.1.90.4 10.1.90.254 //Mandatory user input, use same range on all nodes
```

```

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

cluster group asav-cluster // Mandatory user input, use same cluster name on all nodes
local-unit 1 //Value in bold here must be unique to each node
cluster-interface vni1 ip 1.1.1.1 255.255.255.0 //Value in bold here must be unique to each
node
priority 1
enable noconfirm

mtu geneve-vtep-ifc 1806
mtu ccl_link 1960
aaa authentication listener http geneve-vtep-ifc port 7575 //Use same port number on all
nodes
jumbo-frame reservation
wr mem

```




---

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

---

#### 비 기본 로드 밸런서 예

다음 예는 관리, 내부 및 외부 및 VXLAN 인터페이스가 있는 비 기본 로드 밸런서에 사용할 구성과 클러스터 제어 링크용 VXLAN 인터페이스를 생성합니다.

```

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 10.1.90.254 //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 asav-cluster //mandatory user input
local-unit 1 //mandatory user input
cluster-interface vni1 ip 10.1.1.1 255.255.255.0 //mandatory user input
priority 1
enable
    
```



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(GWLB)를 사용하는 경우 인터페이스 3개, 비 기본 로드 밸런서를 사용하는 경우 인터페이스 4개)로 클러스터 Day 0 구성을 사용하여 ASA 가상 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 3개 - 관리, 외부 및 클러스터 제어 링크.
- 비 기본 로드 밸런서 - 인터페이스 4개 - 관리, 내부, 외부 및 클러스터 제어 링크.

AWS에서 ASA 가상을 구축하는 방법에 대한 자세한 내용은 [AWS에서 ASA 가상 구축](#)을 참조하십시오.

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

**단계 3** ASA 가상 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 생성](#)을 참고하십시오.

## AWS에서 **ASA** 가상에 대한 대상 페일오버 활성화

**ASA** 가상의 데이터 인터페이스는 AWS에서 **GWLB**의 대상 그룹에 등록됩니다. **ASA** 가상 클러스터링에서 각 인스턴스는 대상 그룹과 연결됩니다. **GWLB**는 대상 그룹에서 대상 노드로 식별되거나 등록된 이 정상 인스턴스로 트래픽을 로드 밸런싱하고 전송합니다.

시작하기 전에

수동 방법으로 또는 CloudFormation 템플릿을 사용하여 AWS에 **ASA** 가상 스택을 구축해야 합니다.

CloudFormation 템플릿을 사용하여 클러스터를 구축하는 경우 클러스터 구축 파일 `deploy_asav_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** - **GWLB**가 실패하거나 등록 취소된 대상으로 네트워크 흐름을 계속 전송합니다.
- **rebalance** - **GWLB**는 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 **ASA** 가상 인스턴스 IP 주소가 등록된 대상 그룹을 선택하여 엽니다. 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 리밸런싱)** 슬라이더 버튼을 오른쪽으로 전환합니다. 이렇게 하면 대상 페일오버를 활성화하여 대상 페일오버 또는 등록 취소 시 기존 네트워크 패킷을 정상 대상 노드로 재조정하고 전달하도록 **GWLB**를 구성합니다.



## Azure에서 클러스터 구축

Azure 서비스 체인에서 ASA 가상은 인터넷과 고객 서비스 간의 패킷을 인터셉트할 수 있는 투명 게이트웨이 역할을 합니다. Azure에서 ASA 가상 인스턴스를 클러스터링하면 멀티 노드 ASAv를 단일 디바이스로 추상화하여 처리량을 확장할 수 있습니다.

ASAv는 인터넷을 연결하는 외부 인터페이스와 고객 서비스를 연결하는 내부 인터페이스라는 두 개의 논리 인터페이스로 구성됩니다. 이러한 인터페이스는 패어링된 프록시에서 VXLAN 세그먼트를 사용하여 ASAv의 단일 NIC(Network Interface Card)에서 정의됩니다.

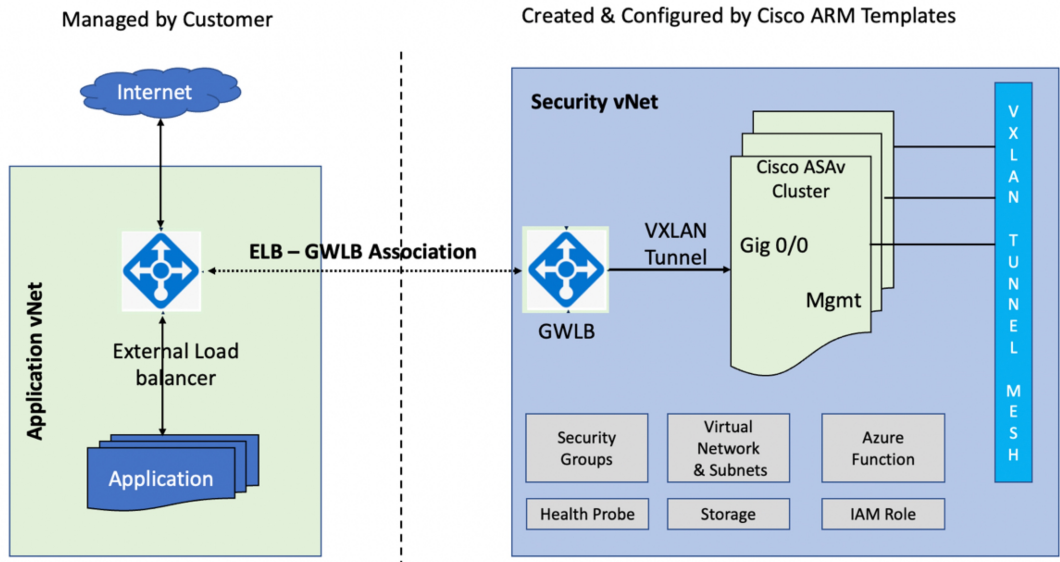
### Azure 게이트웨이 로드 밸런서 정보

Azure GWLB(Gateway Load Balancer)는 트래픽 검사를 위해 VXLAN 세그먼트를 통해 ASAv로 라우팅하여 인바운드 및 아웃바운드 트래픽의 균형을 유지하고 관리하는 데 도움이 됩니다. ASAv 클러스터 환경에서 Azure GLLB는 트래픽 로드와 따라 ASAv 노드의 처리량 레벨을 자동으로 확장합니다. GWLB는 경로를 수동으로 업데이트하지 않고도 네트워크 가상 어플라이언스에 대한 대칭 플로우 또는 일관된 경로를 보장할 수 있습니다. 이 기능을 사용하면 패킷이 양방향으로 동일한 네트워크 경로를 통과할 수 있습니다.

다음 그림에서는 외부 VXLAN 세그먼트의 공용 게이트웨이 로드 밸런서에서 Azure GWLB로 전달되는 트래픽을 보여줍니다. 게이트웨이 로드 밸런서는 주로 여러 ASAv간에 트래픽을 밸런싱하며, 이를 삭제하거나 내부 VXLAN 세그먼트에서 GWLB로 다시 전송하기 전에 트래픽을 검사합니다. 그런 다음 Azure GWLB는 퍼블릭 게이트웨이 로드 밸런서 및 대상으로 트래픽을 다시 전송합니다.

다음 그림은 Azure의 GWLB와 ASAv 간의 네트워크 흐름을 보여줍니다.

그림 4: GWLB를 사용하는 Azure에서 ASAv 클러스터링



## Azure의 클러스터 구축 정보

맞춤형 ARM(Azure Resource Manager) 템플릿을 사용하여 Azure GWLB 용 가상 시스템 확장 세트를 구축합니다.

클러스터 구축 후 Day0 구성을 사용하여 수동으로 또는 Azure 포털의 Function 앱을 통해 클러스터의 각 노드를 구성할 수 있습니다.

## Azure Resource Manager 템플릿을 사용하여 클러스터 구축

클러스터 노드(가상 머신 스케일 집합)를 구축하여 ARM(Azure Resource Manager, Azure 리소스 관리자) 템플릿을 사용하여 클러스터를 형성합니다.

시작하기 전에

- Azure 클러스터를 수동으로 생성하려면 day0 구성이 포함된 구성 텍스트 파일을 준비해야 합니다. [Azure에서 클러스터를 생성하기 위한 구성 파일 준비](#) 참조하십시오.

### 프로시저

단계 1 템플릿을 준비합니다.

- 로컬 폴더에 GitHub 리포지토리를 복제합니다. <https://github.com/CiscoDevNet/cisco-asav/tree/master/cluster/azure>의 내용을 참조하십시오.
- GWLB의 경우 필수 매개변수를 사용하여 `azure_asav_gwlb_cluster.json` 및 `asav-gwlb-cluster-config.txt`를 수정합니다.

단계 2 Azure 포털에 로그인합니다: <https://portal.azure.com>.

단계 3 **Resource group**(리소스 그룹)을 생성합니다.

[Home](#) > [Resource groups](#) >

### Create a resource group ...

Basics Tags Review + create

**Resource group** - A container that holds related resources for an Azure solution. The resource group can include all the resources for the solution, or only those resources that you want to manage as a group. You decide how you want to allocate resources to resource groups based on what makes the most sense for your organization. [Learn more](#) ↗

#### Project details

Subscription \* ⓘ

Resource group \* ⓘ

#### Resource details

Region \* ⓘ

단계 4 ASAv 클러스터에 **Management(관리)**, **Outside(외부)** 및 **CCL(Cluster Control Link, 클러스터 제어 링크)**의 서브넷 3개가 있는 가상 네트워크를 생성합니다. 이 포함된 가상 네트워크를 생성합니다.

[Home](#) > [Resource groups](#) > [asav-cluster-demo](#) > [Marketplace](#) > [Virtual network](#) >

## Create virtual network ...

**Basics** IP Addresses Security Tags Review + create

Azure Virtual Network (VNet) is the fundamental building block for your private network in Azure. VNet enables many types of Azure resources, such as Azure Virtual Machines (VM), to securely communicate with each other, the internet, and on-premises networks. VNet is similar to a traditional network that you'd operate in your own data center, but brings with it additional benefits of Azure's infrastructure such as scale, availability, and isolation. [Learn more about virtual network](#)

### Project details

Subscription \* ⓘ

Resource group \* ⓘ  [Create new](#)

### Instance details

Name \*  ✓

Region \*

[Review + create](#) [< Previous](#) [Next : IP Addresses >](#) [Download a template for automation](#)

단계 5 서브넷을 추가합니다.

Home > Resource groups > asav-cluster-demo > Marketplace > Virtual network >



## Create virtual network ...

Basics IP Addresses Security Tags Review + create

The virtual network's address space, specified as one or more address prefixes in CIDR notation (e.g. 192.168.1.0/24).

### IPv4 address space

10.0.0.0/16 10.0.0.0 - 10.0.255.255 (65536 addresses) 



 Address space '10.0.0.0/16 (10.0.0.0 - 10.0.255.255)' overlaps with address space '10.0.0.0/16 (10.0.0.0 - 10.0.255.255)' of virtual network 'waweb-eastu-4034838410-vnet'. Virtual networks with overlapping address space cannot be peered. If you intend to peer these virtual networks, change address space '10.0.0.0/16 (10.0.0.0 - 10.0.255.255)'. [Learn more](#) 

Add IPv6 address space 

The subnet's address range in CIDR notation (e.g. 192.168.1.0/24). It must be contained by the address space of the virtual network.

 Add subnet  Remove subnet

| <input type="checkbox"/> Subnet name | Subnet address range | NAT gateway |
|--------------------------------------|----------------------|-------------|
| <input type="checkbox"/> Management  | 10.0.0.0/24          | -           |
| <input type="checkbox"/> Data        | 10.0.1.0/24          | -           |
| <input type="checkbox"/> Ccl         | 10.0.2.0/24          | -           |

 A NAT gateway is recommended for outbound internet access from subnets. Edit the subnet to add a NAT gateway. [Learn more](#) 

[Review + create](#)

[< Previous](#)

[Next : Security >](#)

[Download a template for automation](#)

단계 6 맞춤형 템플릿을 구축합니다.

- Create**(생성) > > **Template deployment**(템플릿 구축)(맞춤형 템플릿을 사용하여 구축)을 클릭합니다.
- Build your own template in the editor(편집기에서 자체 템플릿 구축)를 클릭합니다.
- Load File**(파일 로드)을 클릭하고 파일을 업로드합니다.
- Save**(저장)를 클릭합니다.

단계 7 인스턴스 세부 정보를 구성합니다.

단계 8 필요한 값을 입력하고 **Review + create**(검토 + 생성)를 클릭합니다.

Home > Microsoft.VirtualNetwork-20230119131203 | Overview > asav-cluster-vnet > asav-cluster-demo > Marketplace > Template deployment (deploy using custom templates) >

### Custom deployment

Deploy from a custom template

#### Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \*

Resource group \*  [Create new](#)

#### Instance details

Region \*

Resource Name Prefix  ✓

Virtual Network Rg  ✓

Virtual Network Name  ✓

Mgmt Subnet  ✓

Data Interface Subnet  ✓

Gateway Load Balancer IP  ✓

Ccl Subnet  ✓

Internal Port Number  ✓

External Port Number  ✓

Internal Segment Id  ✓

External Segment Id  ✓

[Review + create](#) [< Previous](#) [Next : Review + create >](#)

단계 9 검증이 통과되면 **Create(생성)**를 클릭합니다.

## Custom deployment ...

Deploy from a custom template

✓ Validation Passed

If any Microsoft products are included in a Marketplace offering (e.g. Windows Server or SQL Server), such products are licensed by Microsoft and not by any third party.

### Basics

|                          |  |
|--------------------------|--|
| Subscription             | MSDN Dev/Test Pay-As-You-Go(Converted to EA)                                   |
| Resource group           | sumis-asav-clustering  |
| Region                   | East US  |
| Resource Name Prefix     | asacuster  |
| Virtual Network Rg       | asav-demo-clustering   |
| Virtual Network Name     | asav-clustering-vnet   |
| Mgmt Subnet              | Mgmt   |
| Data Interface Subnet    | Data   |
| Gateway Load Balancer IP | 172.23.2.4   |
| Ccl Subnet               | CCL  |
| Internal Port Number     | 2000   |
| External Port Number     | 2001   |
| Internal Segment Id      | 800  |
| External Segment Id      | 801  |
| Cluster Group Name       | asav-gwlb-cluster  |
| Image Id                 | /subscriptions/33d2517e-ca88-46aa-beb2-74ff1dd61b41/resourceGroups/su...       |
| Vm Size                  | Standard_D3_v2   |
| Asa Admin User Name      | cisco  |
| Asa Admin User Password  | *****  |
| Asav Node Count          | 4  |
| Asav Config File Url     | https://asavconfigs.blob.core.windows.net/asav-configfiles/asav-configurati... |

Create

< Previous

Next

단계 10 인스턴스를 실행한 후 노드 중 하나에 로그인하고 **show cluster info** 명령을 입력하여 클러스터 구축을 확인합니다.

```
> 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
```

다음에 수행할 작업

[Azure에서 클러스터 구성, 27 페이지.](#)

## Azure에서 클러스터 구성

Azure의 ASAv 노드에서 클러스터를 구성하려면 구성 파일 또는 Azure Function 앱을 사용하여 수동으로 구성할 수 있습니다. 기본 GWLB 와 함께 클러스터를 사용할 수 있습니다.

### Azure에서 클러스터를 생성하기 위한 구성 파일 준비

Azure 포털의 Function 앱 또는 구성 파일을 사용하여 ASA 가상 노드에서 클러스터를 수동으로 구성할 수 있습니다.

ASA 가상 노드에서 클러스터를 수동으로 구성하려면 `asav-gowlb-cluster-config.txt` 를 구성해야 합니다. 이 파일에서는 범위 개체, Day0, 클러스터 그룹 이름, 라이선싱 유형 등의 클러스터의 ASA 가상 노드에 구성된 파라미터를 정의해야 합니다.

이 섹션에서는 GWLB 를 사용하여 Azure에서 ASA 가상 노드를 설정하기 위한 클러스터 구성 파일 생성에 대해 설명합니다.

### 프로시저

**단계 1** Cisco GitHub 저장소 디렉토리 `asav-cluster/sample-config-file` 에서 `asav-gwlb-cluster-config.txt` 를 다운로드합니다.

**단계 2** Day0 구성을 준비하여 클러스터 생성을 위한 준비를 할 수 있습니다.

다음 Day0 구성 샘플을 통해 GWLB를 사용하여 Azure에서 클러스터를 생성하는 데 필요한 매개변수를 파악할 수 있습니다.

- **GWLB 클러스터 생성을 위한 샘플 Day0 구성**

다음은 GWLB 클러스터 생성에 사용되는 `asav-gwlb-cluster-config.txt` 파일에 필요한 샘플 Day0 구성입니다.

```
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 GigabitEthernet0/0
  nameif vxlan_tunnel
  security-level 0
  ip address dhcp
  no shutdown

interface GigabitEthernet0/1
  nve-only cluster
  nameif ccl_link
  security-level 0
  ip address dhcp
  no shutdown
```

```

interface vni1
  description ClusterInterface
  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 <CCLSubnetStartAddress> <CCLSubnetEndAddress>
  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>

mtu vxlan_tunnel 1454
mtu ccl_link 1374
cluster group <ClusterGroupName>
local-unit <Last Octet of CCL Interface IP>
cluster-interface vni1 ip 1.1.1.<Last Octet of CCL Interface IP> 255.255.255.0

priority 1
enable

```

위의 샘플 Day0 구성에서 캡슐화 유형이 **vxlan**으로 언급되어 있으면 GWLB 관련 설정이 활성화됩니다. **InternalPort** 및 **ExternalPort**는 vxlan 터널 인터페이스 설정에 사용되는 반면, **InternalSegId** 및 **ExternalSegId**는 내부 및 외부 인터페이스의 식별자로 사용됩니다.

**참고** Day0 구성에서 클러스터 제어 링크의 시작 주소(<CCLSubnetStartAddress>) 및 종료 주소를 지정합니다. 따라서 StartAddress는 항상 x.x.x.4로 시작해야 하며 EndAddress는 최적의 범위 내에 있어야 합니다. 많은 주소를 추가하면 성능에 영향을 줄 수 있으므로 필요한 주소 수(최대 16개)만 지정하는 것이 좋습니다.

예를 들어 CCL 서브넷이 192.168.3.0/24인 경우 StartAddress는 192.168.3.4가 되고 EndAddress는 192.168.3.30일 수 있습니다.

다음은 vni 인터페이스에 필요한 샘플 구성입니다.

```

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
```

단계 3 설정 파일을 Azure 스토리지에 업로드하고 이 위치의 경로(URL)를 적어 둡니다. 이 URL 경로는 ASA 가상 노드에서 클러스터를 수동으로 구성하는 데 필요합니다.

### 설정 파일을 수동으로 사용하여 클러스터 설정

구성 파일을 사용하여 Azure의 ASAv 노드에 클러스터를 수동으로 구성합니다.

시작하기 전에

구성 파일을 준비하고 파일이 업로드된 Azure 스토리지 위치를 기록해야 합니다. Azure에서 클러스터 구성 파일 준비를 참조하십시오.

#### 프로시저

단계 1 Azure 포털에 로그인합니다.

단계 2 Azure에 구축된 ASAv 인스턴스를 엽니다.

단계 3 Azure 스토리지 컨테이너에 업로드한 파일의 URL을 제공하여 다음 명령을 실행하여 클러스터 구성 파일을 ASAv 노드에 복사합니다.

**copy <Config File URL> running-config**

단계 4 다음 명령을 실행하여 ASAv 인스턴스에서 클러스터를 구성합니다.

```
cluster group <ClusterGroupName>
  local-unit <Last Octet of the Management Interface IP>
  cluster-interface vni1 ip 1.1.1.<Last Octet of the Management Interface IP>
255.255.255.0
  priority 1
  enable
```

단계 5 2~4단계를 반복하여 모든 ASAv 노드에서 클러스터를 설정합니다.

### Azure Function 앱을 사용하여 클러스터 구성

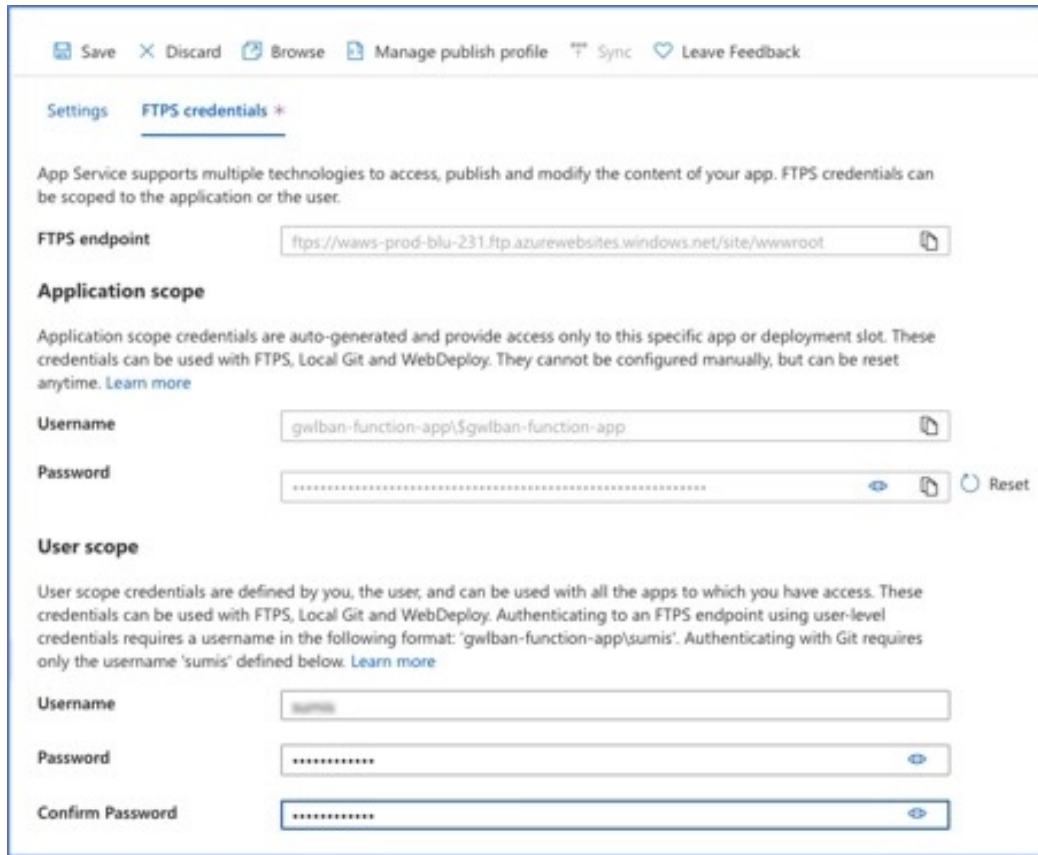
Azure Function 앱 서비스를 사용하여 Azure의 ASAv 노드에 클러스터를 구성합니다.

#### 프로시저

단계 1 Azure 포털에 로그인합니다.

단계 2 **Function App(Function 앱)**을 클릭합니다.

단계 3 **Deployment Center(구축 센터) > HTTPS credentials(HTTPS 자격 증명) > User scope(사용자 범위) > Configure Username and Password(사용자 이름 및 비밀번호 구성) >** 를 클릭한 다음 **Save(저장)**를 클릭합니다.



단계 4 로컬 터미널에서 다음 명령을 실행하여 Cluster\_Function.zip 파일을 Function 앱에 업로드합니다.

```
curl -X POST -u <Userscope_Username> --data-binary @"Cluster_Function.zip"
https://<Function_App_Name>.scm.azurewebsites.net/api/zipdeploy
```

그림 5: 카탈로그

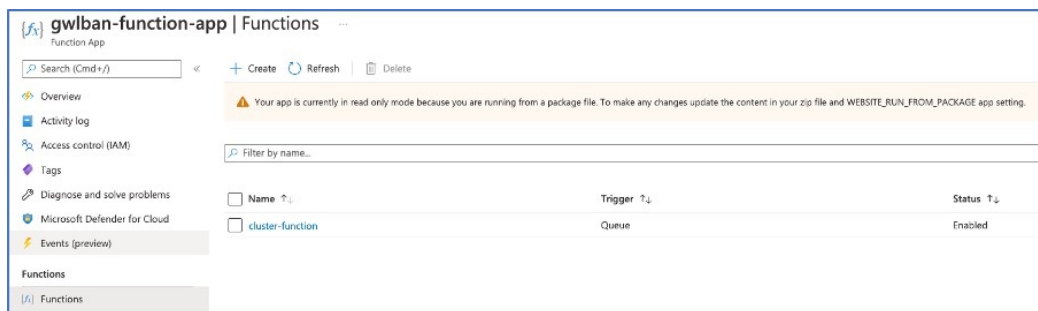


그림 6: 큐

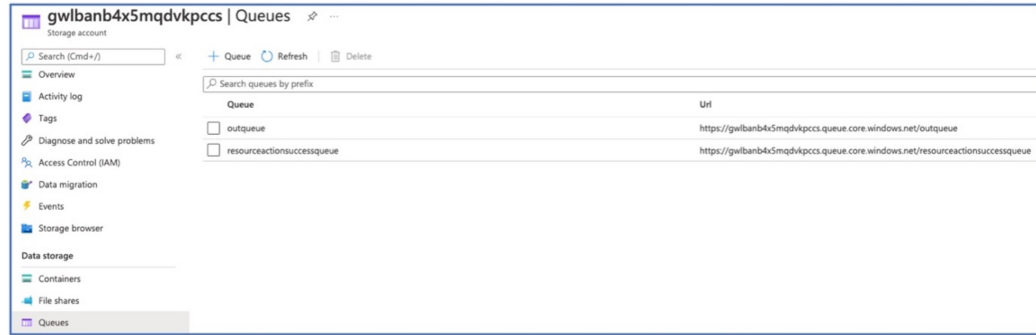
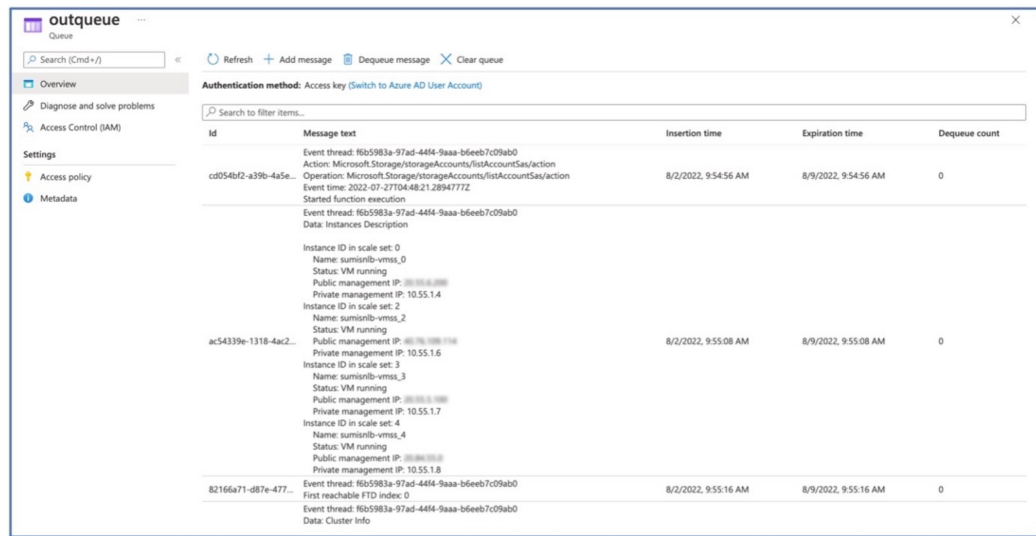


그림 7: Outqueue



함수가 함수 앱에 업로드됩니다. 함수가 시작되고 스토리지 계정의 outqueue에서 로그를 볼 수 있습니다.

Function 실행 후에는 모든 ASAv 노드에서 클러스터가 활성화됩니다.

## Azure에서 ASA 가상 클러스터 문제 해결

트래픽 문제

트래픽이 작동하지 않으면 다음을 확인합니다.

1. 로드 밸런서가 있는 ASA 가상 인스턴스의 상태 프로브 상태가 정상인지 확인하십시오.  
ASA 가상 인스턴스의 상태 프로브 상태가 unhealthy(비정상)인 경우 다음 작업을 수행합니다.
  1. ASA 가상에 구성되어 있는 정적 경로를 확인합니다.
  2. 기본 게이트웨이가 데이터 서브넷의 게이트웨이 IP인지 확인합니다.

3. ASA 가상 인스턴스가 상태 프로브 트래픽을 수신하는지 확인합니다.
4. ASA 가상 에 구성된 액세스 정책이 상태 프로브 트래픽을 허용하고 있는지 확인합니다.

### 클러스터 문제

클러스터가 형성되지 않은 경우 다음을 확인합니다.

- Network Virtualization Endpoint(NVE 전용) 클러스터 인터페이스의 IP 주소. 다른 노드의 NVE 전용 클러스터 인터페이스를 ping할 수 있는지 확인합니다.
- NVE 전용 클러스터 인터페이스의 IP 주소가 개체 그룹의 일부인지 확인합니다. NVE가 개체 그룹으로 구성되어 있는지 확인합니다.
- 클러스터 그룹의 클러스터 인터페이스에 올바른 VNI 인터페이스가 있는지 확인합니다. 이 VNI 인터페이스에는 해당 개체 그룹이 있는 NVE가 있습니다.
- 각 노드에는 자체 IP 인터페이스가 있으므로 노드가 서로 ping을 수행하여 클러스터의 노드 간의 연결성을 보장할 수 있는지 확인합니다.
- 템플릿 구축 중에 언급된 CCL 서브넷의 시작 및 종료 주소가 올바른지 확인합니다. 시작 주소는 서브넷에서 사용 가능한 첫 번째 IP 주소로 시작해야 합니다. 예를 들어 서브넷이 192.168.1.0/24 인 경우, 시작 주소는 192.168.1.4여야 합니다(시작 시 3개의 IP 주소는 Azure에서 예약됨).

### 역할 관련 문제

동일한 리소스 그룹에서 리소스를 다시 구축하는 동안 역할 관련 오류가 발생하는 경우 다음을 수행합니다.

특정 역할과 관련된 문제가 있는 경우 오류 메시지가 표시됩니다.

다음은 샘플 오류 메시지입니다.

```
"error(오류)": {
```

```
"code(코드)": "RoleAssignmentUpdateNotPermitted",
```

```
"message(메세지)": "테넌트 ID, 애플리케이션 ID, 보안 주체 ID, 범위는 업데이트할 수 없습니다."/}
```

터미널에서 다음 명령을 실행하여 다음 역할을 제거합니다.

- 스토리지 대기열 데이터 기여자 역할을 제거하는 명령:

```
az role assignment delete --resource-group <Resource Group Name> --role "Storage Queue Data Contributor"
```

- 기여자 역할을 제거하는 명령:

```
az role assignment delete --resource-group <Resource Group Name> --role "Contributor"
```

## 클러스터링 운영 맞춤화

Day 0 구성의 일부로 또는 클러스터를 구축한 후에 클러스터링 상태 모니터링, TCP 연결 복제 지연, 플로우 모빌리티 및 기타 최적화를 맞춤화할 수 있습니다.

다음 절차는 제어 노드에서 수행합니다.

### 기본 ASA 클러스터 파라미터 구성

제어 노드에서 클러스터 설정을 맞춤화할 수 있습니다.

#### 프로시저

단계 1 클러스터 구성 모드로 들어갑니다.

**cluster group** *name*

단계 2 (선택 사항) 데이터 노드에서 제어 노드로 콘솔 복제를 활성화합니다.

**console-replicate**

이 기능은 기본적으로 비활성화되어 있습니다. ASA에서는 중요한 특정 이벤트 발생 시 일부 메시지를 콘솔에 직접 출력합니다. 콘솔 복제를 활성화할 경우, 데이터 노드에서는 콘솔 메시지를 제어 노드에 전송하므로 클러스터의 콘솔 포트 하나만 모니터링하면 됩니다.

단계 3 클러스터링 이벤트의 최소 추적 레벨을 설정합니다

**trace-level** 레벨

원하는 대로 최소 레벨을 설정합니다.

- **critical**— 중요 이벤트(심각도=1)
- **warning**— 경고(심각도=2)
- **informational**— 정보 이벤트(심각도=3)
- **debug**— 디버깅 이벤트(심각도=4)

단계 4 플로우 소유자에서 디렉터 및 백업 소유자로서의 플로우 상태 새로 고침 메시지(`clu_keepalive` 및 `clu_update` 메시지)에 대한 `keepalive` 간격을 설정합니다.

**clu-keepalive-interval** *seconds*

- 초 - 15~55 기본값은 15입니다.

클러스터 제어 링크의 트래픽 양을 줄이기 위해 간격을 기본값보다 길게 설정할 수 있습니다.

## 상태 모니터링 및 자동 재참가 설정 구성

이 절차에서는 노드 및 인터페이스 상태 모니터링을 구성합니다.

필수가 아닌 인터페이스(예: 관리 인터페이스)에 대한 상태 모니터링을 비활성화할 수 있습니다. 상태 모니터링은 VLAN 하위 인터페이스에서 수행되지 않습니다. 클러스터 제어 링크의 모니터링을 구성할 수 없습니다. 이 링크는 항상 모니터링됩니다.

### 프로시저

단계 1 클러스터 구성 모드로 들어갑니다.

**cluster group** *name*

예제:

```
ciscoasa(config)# cluster group test
ciscoasa(cfg-cluster)#
```

단계 2 클러스터 노드 상태 검사 기능을 맞춤화합니다.

**health-check** [**holdtime** *Timeout*(시간 초과)]

노드 상태를 확인하기 위해 ASA 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.

- **holdtime** *Timeout*(시간 초과 - 노드 하트비트 상태 메시지 간의 시간 간격을 0.3초에서 45초 범위에서 지정합니다. 기본값은 3초입니다).

토폴로지 변경 사항(예: 데이터 인터페이스 추가 또는 제거, ASA 또는 스위치에서 인터페이스 활성화 또는 비활성화)이 발생할 경우 상태 검사 기능을 비활성화하고 비활성화된 인터페이스에 대한 인터페이스 모니터링도 비활성화해야 합니다(**no health-check monitor-interface**). 토폴로지 변경이 완료되고 구성 변경 사항이 모든 노드와 동기화되면 상태 검사 기능을 다시 사용할 수 있습니다.

예제:

```
ciscoasa(cfg-cluster)# health-check holdtime 5
```

단계 3 인터페이스에서 인터페이스 상태 검사를 비활성화합니다.

**no health-check monitor-interface** *interface\_id*

인터페이스 상태 검사에서는 링크 오류 여부를 모니터링합니다. ASA에서 클러스터의 멤버를 제거하기 전까지 걸리는 시간은 해당 노드가 설정된 멤버인지 또는 클러스터에 참가하는지에 따라 달라집니다. 상태 선택은 모든 인터페이스에 대해 기본적으로 활성화됩니다. 이 명령의 **no** 형식을 사용하여 이를 인터페이스별로 비활성화할 수 있습니다. 필수가 아닌 인터페이스(예: 관리 인터페이스)에 대한 상태 모니터링을 비활성화할 수 있습니다.

- **interface\_id**- 인터페이스의 모니터링을 비활성화합니다. 상태 모니터링은 VLAN 하위 인터페이스에서 수행되지 않습니다. 클러스터 제어 링크의 모니터링을 구성할 수 없습니다. 이 링크는 항상 모니터링됩니다.

토폴로지 변경 사항(예: 데이터 인터페이스 추가 또는 제거, ASA 또는 스위치에서 인터페이스 활성화 또는 비활성화)이 발생할 경우 상태 검사 기능을 비활성화(**no health-check**)하고 비활성화된 인터페이스에 대한 인터페이스 모니터링도 비활성화해야 합니다. 토폴로지 변경이 완료되고 구성 변경 사항이 모든 노드와 동기화되면 상태 검사 기능을 다시 사용할 수 있습니다.

예제:

```
ciscoasa(cfg-cluster)# no health-check monitor-interface management1/1
```

단계 4 상태 검사에 실패한 후에 자동 다시 참가 클러스터 설정을 맞춤화합니다.

**health-check {data-interface | cluster-interface | system} auto-rejoin [unlimited | auto\_rejoin\_max] auto\_rejoin\_interval auto\_rejoin\_interval\_variation**

- **system**— 내부 오류에 대한 자동 다시 참가 설정을 지정합니다. 내부 오류 포함: 애플리케이션 동기화 시간 초과, 일치하지 않는 애플리케이션 상태 등
- **unlimited** — (**cluster-interface**의 기본값) 다시 참가 시도 횟수를 제한하지 않습니다.
- **auto-rejoin-max** — 다시 참가 시도 횟수를 0~65535 사이로 설정합니다. 0은 자동 다시 참가를 비활성화합니다. **data-interface** 및 **system**에 대한 기본값은 3입니다.
- **auto\_rejoin\_interval** — 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.
- **auto\_rejoin\_interval\_variation** — 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 클러스터 인터페이스의 경우 **1**이며 데이터 인터페이스 및 시스템의 경우 **2**입니다.

예제:

```
ciscoasa(cfg-cluster)# health-check data-interface auto-rejoin 10 3 3
```

단계 5 ASA가 인터페이스를 실패 상태로 간주하고 노드가 클러스터에서 제거되기 전에 디바운스 시간을 구성합니다.

**health-check monitor-interface debounce-time** 밀리초

예제:

```
ciscoasa(cfg-cluster)# health-check monitor-interface debounce-time 300
```

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 ASA는 지정되어 있는 밀리초 동안 대기합니다.

단계 6 (선택 사항) 트래픽 로드 모니터링을 구성합니다.

**load-monitor [frequency 초] [intervals 간격]**

- **frequency seconds(초)**- 모니터링 메시지 사이의 시간(초)을 10~360초 사이로 설정합니다. 기본값은 20초입니다.
- **intervals interval(간격)**- ASA가 데이터를 유지 관리하는 간격의 수를 1~60 사이의 값으로 설정합니다. 기본값은 30입니다.

총 연결 수, CPU 및 메모리 사용량, 버퍼 삭제를 포함한 클러스터 멤버의 트래픽 로드를 모니터링할 수 있습니다. 로드가 너무 높은 경우 나머지 노드가 로드를 처리할 수 있는 경우 노드에서 클러스터링을 수동으로 비활성화하도록 선택하거나 외부 스위치의 로드 밸런싱을 조정할 수 있습니다. 이 기능은 기본적으로 활성화되어 있습니다. 트래픽 로드를 주기적으로 모니터링할 수 있습니다. 로드가 너무 높은 경우 노드에서 수동으로 클러스터링을 비활성화하도록 선택할 수 있습니다.

**show cluster info load-monitor** 명령을 사용하여 트래픽 로드를 확인합니다.

예제:

```
ciscoasa(cfg-cluster)# load-monitor frequency 50 intervals 25
ciscoasa(cfg-cluster)# show cluster info load-monitor
ID Unit Name
0 B
1 A_1
Information from all units with 50 second interval:
Unit Connections Buffer Drops Memory Used CPU Used
Average from last 1 interval:
0 0 0 14 25
1 0 0 16 20
Average from last 25 interval:
0 0 0 12 28
1 0 0 13 27
```

예

다음 예에서는 상태 확인 보류 시간을 0.3초로 구성하고, Management 0/0 인터페이스에서 모니터링을 비활성화하며, 데이터 인터페이스에 대한 자동 다시 참가를 2분에 시작하는 4회 시도도로 설정하고, 기간을 3 x 이전 간격으로 늘리며, 클러스터 제어 링크에 대한 자동 다시 참가를 2분마다 6회 시도도로 설정합니다.

```
ciscoasa(config)# cluster group test
ciscoasa(cfg-cluster)# health-check holdtime .3
ciscoasa(cfg-cluster)# no health-check monitor-interface management0/0
ciscoasa(cfg-cluster)# health-check data-interface auto-rejoin 4 2 3
```



```
ciscoasa(cfg-cluster)# health-check cluster-interface auto-rejoin 6 2 1
```

## 클러스터 노드 관리

클러스터를 배치한 후에는 구성을 변경하고 클러스터 노드를 관리할 수 있습니다.

### 비활성 노드 되기

클러스터의 멤버를 비활성화하려면, 클러스터링 구성은 그대로 유지한 상태로 노드의 클러스터링을 비활성화합니다.



**참고** 수동으로 또는 상태 확인 장애를 통해 ASA가 비활성화되면 모든 데이터 인터페이스가 종료되며, 관리 전용 인터페이스에서만 트래픽을 주고받을 수 있습니다. 트래픽 흐름을 다시 시작하려면 클러스터링을 다시 활성화합니다. 또는 클러스터에서 노드를 모두 제거할 수 있습니다. 관리 인터페이스에서는 클러스터 IP 풀에서 노드로 전송된 IP 주소를 사용하여 가동 상태를 유지합니다. 그러나 다시 로드해도 노드가 클러스터에서 여전히 비활성 상태인 경우(예를 들어 클러스터링이 비활성화된 구성을 저장한 경우)에는 관리 인터페이스가 비활성화됩니다. 추가 구성을 위해서는 콘솔 포트를 사용해야 합니다.

### 프로시저

**단계 1** 클러스터 구성 모드로 들어갑니다.

```
cluster group name
```

예제:

```
ciscoasa(config)# cluster group pod1
```

**단계 2** 클러스터링을 비활성화합니다.

```
no enable
```

이 노드가 제어 노드인 경우 새 제어 선택이 이루어지고 다른 멤버가 제어 노드가 됩니다.

클러스터 구성은 그대로 유지되므로 클러스터링을 나중에 다시 활성화할 수 있습니다.

### 제어 노드에서 데이터 노드 비활성화

로그인한 노드 이외의 멤버를 비활성화하려면 다음 단계를 수행합니다.



참고 ASA가 비활성화되면 모든 데이터 인터페이스가 종료되며 관리 전용 인터페이스에서만 트래픽을 주고받을 수 있습니다. 트래픽 흐름을 재개하려면 클러스터링을 다시 활성화합니다. 관리 인터페이스에서는 클러스터 IP 풀에서 노드로 전송된 IP 주소를 사용하여 가동 상태를 유지합니다. 그러나 다시 로드해도 노드가 클러스터에서 여전히 비활성 상태인 경우(예를 들어 클러스터링이 비활성화된 구성을 저장한 경우)에는 관리 인터페이스가 비활성화됩니다. 추가 구성을 위해서는 콘솔 포트를 사용해야 합니다.

## 프로시저

클러스터에서 노드를 제거합니다.

**cluster remove unit *node\_name***

부트스트랩 구성과 제어 노드에서 동기화한 마지막 구성도 그대로 유지되므로 나중에 구성이 유실되는 일 없이 노드를 다시 추가할 수 있습니다. 이 명령을 데이터 노드에 입력해서 제어 노드를 제거하면 새로운 제어 노드가 선택됩니다.

멤버 이름을 보려면 **cluster remove unit ?**을 입력하거나 **show cluster info** 명령을 입력합니다.

예제:

```
ciscoasa(config)# cluster remove unit ?
Current active units in the cluster:
asa2

ciscoasa(config)# cluster remove unit asa2
WARNING: Clustering will be disabled on unit asa2. To bring it back
to the cluster please logon to that unit and re-enable clustering
```

## 클러스터 재참가

노드가 클러스터에서 제거된 경우, 예를 들어 실패한 인터페이스의 경우 또는 멤버를 수동으로 비활성화한 경우, 클러스터를 수동으로 다시 조인해야 합니다.

## 프로시저

단계 1 콘솔에서 클러스터 구성 모드를 시작합니다.

**cluster group *name***

예제:

```
ciscoasa(config)# cluster group pod1
```

단계 2 클러스터링을 활성화합니다.

**enable**

## 클러스터 벗어나기

클러스터를 모두 벗어나려는 경우, 전체 클러스터 부트스트랩 컨피그레이션을 제거해야 합니다. 각 노드에 대한 현재 구성이 동일하므로(활성 유닛에서 동기화됨), 클러스터를 벗어날 경우 백업에서 사전 클러스터링 구성을 복원하거나, IP 주소 충돌을 피하기 위해 구성을 지우고 처음부터 다시 시작하게 됩니다.

### 프로시저

단계 1 데이터 노드의 경우 클러스터링을 비활성화합니다.

**cluster group *cluster\_name* no enable**

예제:

```
ciscoasa(config)# cluster group cluster1
ciscoasa(cfg-cluster)# no enable
```

데이터 노드에 클러스터링이 활성화되어 있는 동안에는 구성을 변경할 수 없습니다.

단계 2 클러스터 구성을 지웁니다.

**clear configure cluster**

ASA에서는 관리 인터페이스 및 클러스터 제어 링크를 비롯한 모든 인터페이스를 종료합니다.

단계 3 클러스터 인터페이스 모드를 비활성화합니다.

**no cluster interface-mode**

모드는 구성에 저장되지 않으며 수동으로 재설정해야 합니다.

단계 4 백업 구성이 있을 경우, 실행 중인 구성에 백업 구성을 복사합니다.

**copy *backup\_cfg* running-config**

예제:

```
ciscoasa(config)# copy backup_cluster.cfg running-config

Source filename [backup_cluster.cfg]?

Destination filename [running-config]?
ciscoasa(config)#
```

단계 5 시작에 구성을 저장합니다.

**write memory**

단계 6 백업 구성이 없는 경우 관리 액세스를 다시 구성합니다. 인터페이스 IP 주소를 변경하고 이를테면 올바른 호스트 이름을 복원해야 합니다.

## 제어 노드 변경



주의 제어 노드를 변경하는 가장 좋은 방법은 제어 노드의 클러스터링을 비활성화한 후 새 제어 노드가 선택될 때까지 기다렸다가 클러스터링을 다시 활성화하는 것입니다. 제어 노드가 될 정확한 노드를 지정해야 할 경우, 이 섹션의 절차를 참조하십시오. 그러나 중앙 집중식 기능의 경우, 이 절차를 통해 제어 노드를 강제로 변경하면 모든 연결이 끊어지며 새 제어 노드에서 연결을 다시 설정해야 합니다.

제어 노드를 변경하려면 다음 단계를 수행합니다.

### 프로시저

새 노드를 제어 노드로 설정합니다.

**cluster control-node unit***node\_name*

예제:

```
ciscoasa(config)# cluster control-node unit asa2
```

기본 클러스터 IP 주소에 다시 연결해야 합니다.

멤버 이름을 보려면 **cluster control-node unit ?**을 입력합니다. (현재 노드를 제외한 모든 이름을 보려는 경우), 또는 **show cluster info** 명령을 입력합니다.

## 클러스터 전체에서 명령 실행

클러스터의 모든 노드 또는 특정 노드에 명령을 보내려면 다음 단계를 수행합니다. 모든 노드에 **show** 명령을 보내면 모든 출력이 수집되고 해당 내용이 현재 노드의 콘솔에 표시됩니다. **capture** 및 **copy** 같은 다른 명령의 경우 클러스터 전체 실행을 활용할 수도 있습니다.

### 프로시저

모든 노드에 명령을 전송하거나, 노드 이름을 지정한 경우 특정 노드를 지정합니다.

**cluster exec [ unit node\_name]** 명령

예제:

```
ciscoasa# cluster exec show xlate
```

노드 이름을 보려면 **cluster exec unit ?**을 입력합니다. (현재 노드를 제외한 모든 이름을 보려는 경우), 또는 **show cluster info** 명령을 입력합니다.

예

클러스터에 있는 모든 노드의 동일한 캡처 파일을 TFTP 서버에 동시에 복사하려면 다음 명령을 제어 노드에 입력합니다.

```
ciscoasa# cluster exec copy /pcap capture: tftp://10.1.1.56/capture1.pcap
```

노드당 하나씩인 여러 PCAP 파일이 TFTP 서버에 복사됩니다. 목적지 캡처 파일의 이름 뒤에는 노드 이름이 자동으로 연결되며 capture1\_asa1.pcap, capture1\_asa2.pcap 같은 형식이 됩니다. 이 예에서 asa1 및 asa2는 클러스터 노드 이름입니다.

## 클러스터 모니터링

클러스터의 상태 및 연결을 모니터링하고 문제를 해결할 수 있습니다.

### 클러스터 상태 모니터링

클러스터 상태 모니터링에 대한 내용은 다음 명령을 참조하십시오.

- **show cluster info [health [details]]**

키워드가 없는 경우 **show cluster info** 명령을 사용하면 클러스터의 모든 멤버 상태가 표시됩니다.

**show cluster info health** 명령을 사용하면 인터페이스, 노드, 클러스터 전반의 현재 상태가 표시됩니다. **details** 키워드를 사용하면 숫자 하트비트 메시지 장애가 표시됩니다.

**show cluster info** 명령에 대한 내용은 다음 출력을 참조하십시오.

```
ciscoasa# show cluster info
Cluster stbu: On
  This is "C" in state DATA_NODE
    ID       : 0
    Site ID  : 1
              Version   : 9.4(1)
    Serial No.: P3000000025
    CCL IP   : 10.0.0.3
    CCL MAC  : 000b.fcf8.c192
    Last join : 17:08:59 UTC Sep 26 2011
    Last leave: N/A
Other members in the cluster:
  Unit "D" in state DATA_NODE
    ID       : 1
    Site ID  : 1
              Version   : 9.4(1)
    Serial No.: P3000000001
    CCL IP   : 10.0.0.4
```

```

CCL MAC   : 000b.fcf8.c162
Last join : 19:13:11 UTC Sep 23 2011
Last leave: N/A
Unit "A" in state CONTROL_NODE
  ID      : 2
  Site ID : 2
    Version : 9.4(1)
  Serial No.: JAB0815R0JY
  CCL IP   : 10.0.0.1
  CCL MAC  : 000f.f775.541e
  Last join : 19:13:20 UTC Sep 23 2011
  Last leave: N/A
Unit "B" in state DATA_NODE
  ID      : 3
  Site ID : 2
    Version : 9.4(1)
  Serial No.: P3000000191
  CCL IP   : 10.0.0.2
  CCL MAC  : 000b.fcf8.c61e
  Last join : 19:13:50 UTC Sep 23 2011
  Last leave: 19:13:36 UTC Sep 23 2011

```

#### • show cluster info auto-join

시간 지연 이후에 클러스터 노드가 자동으로 클러스터에 다시 참가하는지 여부 및 오류 상태(예: 라이선스 대기 중, 새시 상태 검사 오류 등)가 지워졌는지 여부를 표시합니다. 노드가 영구적으로 비활성화된 경우 또는 노드가 이미 클러스터에 있는 경우, 이 명령은 출력을 표시하지 않습니다.

**show cluster info auto-join** 명령에 대한 내용은 다음 출력을 참조하십시오.

```

ciscoasa(cfg-cluster)# show cluster info auto-join
Unit will try to join cluster in 253 seconds.
Quit reason: Received control message DISABLE

ciscoasa(cfg-cluster)# show cluster info auto-join
Unit will try to join cluster when quit reason is cleared.
Quit reason: Control node has application down that data node has up.

ciscoasa(cfg-cluster)# show cluster info auto-join
Unit will try to join cluster when quit reason is cleared.
Quit reason: Chassis-blade health check failed.

ciscoasa(cfg-cluster)# show cluster info auto-join
Unit will try to join cluster when quit reason is cleared.
Quit reason: Service chain application became down.

ciscoasa(cfg-cluster)# show cluster info auto-join
Unit will try to join cluster when quit reason is cleared.
Quit reason: Unit is kicked out from cluster because of Application health check failure.

ciscoasa(cfg-cluster)# show cluster info auto-join
Unit join is pending (waiting for the smart license entitlement: ent1)

ciscoasa(cfg-cluster)# show cluster info auto-join
Unit join is pending (waiting for the smart license export control flag)

```

#### • show cluster info transport {asp | cp [detail]}

다음에 대한 전송 관련 통계를 표시합니다.

- **asp** — 데이터 평면 전송 통계입니다.
- **cp** — 제어 평면 전송 통계입니다.

**detail** 키워드를 입력하는 경우, 클러스터의 신뢰할 수 있는 전송 프로토콜 사용량을 볼 수 있어 버퍼가 제어 평면에서 가득 찬 경우 패킷 삭제 문제를 식별할 수 있습니다. **show cluster info transport cp detail** 명령에 대한 내용은 다음 출력을 참조하십시오.

```
ciscoasa# show cluster info transport cp detail
Member ID to name mapping:
  0 - unit-1-1  2 - unit-4-1  3 - unit-2-1

Legend:
U    - unreliable messages
UE   - unreliable messages error
SN   - sequence number
ESN  - expecting sequence number
R    - reliable messages
RE   - reliable messages error
RDC  - reliable message deliveries confirmed
RA   - reliable ack packets received
RFR  - reliable fast retransmits
RTR  - reliable timer-based retransmits
RDP  - reliable message dropped
RDPR - reliable message drops reported
RI   - reliable message with old sequence number
RO   - reliable message with out of order sequence number
ROW  - reliable message with out of window sequence number
ROB  - out of order reliable messages buffered
RAS  - reliable ack packets sent

This unit as a sender
-----
      all      0      2      3
U    123301    3867966  3230662  3850381
UE   0         0         0         0
SN   1656a4ce  acb26fe  5f839f76  7b680831
R    733840    1042168  852285   867311
RE   0         0         0         0
RDC  699789    934969  740874   756490
RA   385525    281198  204021   205384
RFR  27626     56397   0         0
RTR  34051     107199  111411   110821
RDP  0         0         0         0
RDPR 0         0         0         0

This unit as a receiver of broadcast messages
-----
      0      2      3
U    111847    121862  120029
R    7503     665700  749288
ESN  5d75b4b3  6d81d23  365ddd50
RI   630      34278   40291
RO   0        582     850
ROW  0        566     850
ROB  0         16       0
RAS  1571     123289  142256

This unit as a receiver of unicast messages
-----
```

```

      0      2      3
U      1      3308122  4370233
R      513846  879979  1009492
ESN    4458903a 6d841a84 7b4e7fa7
RI     66024   108924  102114
RO     0       0       0
ROW    0       0       0
ROB    0       0       0
RAS    130258  218924  228303
    
```

Gated Tx Buffered Message Statistics

```

-----
current sequence number: 0

total:          0
current:        0
high watermark: 0

delivered:      0
deliver failures: 0

buffer full drops: 0
message truncate drops: 0

gate close ref count: 0

num of supported clients:45
    
```

MRT Tx of broadcast messages

```

=====
Message high watermark: 3%
Total messages buffered at high watermark: 5677
[Per-client message usage at high watermark]
-----
Client name                Total messages  Percentage
Cluster Redirect Client    4153            73%
Route Cluster Client        419             7%
RRI Cluster Client          1105            19%
    
```

```

Current MRT buffer usage: 0%
Total messages buffered in real-time: 1
[Per-client message usage in real-time]
Legend:
    
```

- F - MRT messages sending when buffer is full
- L - MRT messages sending when cluster node leave
- R - MRT messages sending in Rx thread

```

-----
Client name                Total messages  Percentage  F  L  R
VPN Clustering HA Client    1             100%      0  0  0
    
```

MRT Tx of unitcast messages(to member\_id:0)

```

=====
Message high watermark: 31%
Total messages buffered at high watermark: 4059
[Per-client message usage at high watermark]
-----
Client name                Total messages  Percentage
Cluster Redirect Client    3731            91%
RRI Cluster Client          328             8%
    
```

```

Current MRT buffer usage: 29%
Total messages buffered in real-time: 3924
[Per-client message usage in real-time]
Legend:
    
```



```

F - MRT messages sending when buffer is full
L - MRT messages sending when cluster node leave
R - MRT messages sending in Rx thread
-----
Client name                               Total messages  Percentage    F    L    R
Cluster Redirect Client                   3607            91%         0    0    0
RRI Cluster Client                        317             8%         0    0    0

MRT Tx of unitcast messages(to member_id:2)
=====
Message high watermark: 14%
Total messages buffered at high watermark: 578
[Per-client message usage at high watermark]
-----
Client name                               Total messages  Percentage
VPN Clustering HA Client                   578            100%

Current MRT buffer usage: 0%
Total messages buffered in real-time: 0

MRT Tx of unitcast messages(to member_id:3)
=====
Message high watermark: 12%
Total messages buffered at high watermark: 573
[Per-client message usage at high watermark]
-----
Client name                               Total messages  Percentage
VPN Clustering HA Client                   572            99%
Cluster VPN Unique ID Client               1              0%

Current MRT buffer usage: 0%
Total messages buffered in real-time: 0

```

• **show cluster history**

클러스터 노드가 참여하지 못한 이유 또는 노드가 클러스터에서 이탈한 이유에 대한 오류 메시지와 함께 클러스터 기록을 표시합니다.

## 클러스터 전체 패킷 캡처

클러스터의 패킷을 캡처하는 방법에 대한 내용은 다음 명령을 참조하십시오.

**cluster exec capture**

클러스터 전체의 문제를 해결하기 위해 **cluster exec capture** 명령을 사용하여 제어 노드에서 클러스터별 트래픽의 캡처를 활성화할 수 있습니다. 이 경우 클러스터의 모든 데이터 노드에서 캡처가 자동으로 활성화됩니다.

## 클러스터 리소스 모니터링

클러스터 리소스 모니터링에 대한 내용은 다음 명령을 참조하십시오.

**show cluster {cpu | memory | resource} [options]**

전체 클러스터에 대한 집계된 데이터를 표시합니다. 사용 가능한 옵션은 데이터 유형에 따라 달라집니다.

## 클러스터 트래픽 모니터링

클러스터 트래픽 모니터링에 대한 내용은 다음 명령을 참조하십시오.

- **show conn [detail], cluster exec show conn**

**show conn** 명령을 사용하면 흐름이 관리자, 백업 또는 전달자 흐름인지 보여 줍니다. 노드에 **cluster exec show conn** 명령을 사용하여 모든 연결을 볼 수 있습니다. 이 명령은 클러스터의 다른 ASA에 단일 플로우에 대한 트래픽이 어떤 방식으로 도착하는지를 표시할 수 있습니다. 클러스터의 처리량은 로드 밸런싱의 효율성과 구성에 따라 달라집니다. 이 명령을 사용하면 연결에 대한 트래픽 흐름이 클러스터를 통해 어떻게 이루어지는지 손쉽게 볼 수 있으며, 로드 밸런서가 이 흐름의 성능에 어떤 영향을 미치는지 파악하는 데 유용합니다.

**show conn detail** 명령을 사용하면 플로우 모빌리티의 영향을 받는 플로우도 표시됩니다.

다음은 **show conn detail** 명령의 샘플 출력입니다.

```
ciscoasa/ASA2/data node# show conn detail
12 in use, 13 most used
Cluster stub connections: 0 in use, 46 most used
Flags: A - awaiting inside ACK to SYN, a - awaiting outside ACK to SYN,
      B - initial SYN from outside, b - TCP state-bypass or nailed,
      C - CTIQBE media, c - cluster centralized,
      D - DNS, d - dump, E - outside back connection, e - semi-distributed,
      F - outside FIN, f - inside FIN,
      G - group, g - MGCP, H - H.323, h - H.225.0, I - inbound data,
      i - incomplete, J - GTP, j - GTP data, K - GTP t3-response
      k - Skinny media, L - LISP triggered flow owner mobility,
      M - SMTP data, m - SIP media, n - GUP
      O - outbound data, o - offloaded,
      P - inside back connection,
      Q - Diameter, q - SQL*Net data,
      R - outside acknowledged FIN,
      R - UDP SUNRPC, r - inside acknowledged FIN, S - awaiting inside SYN,
      s - awaiting outside SYN, T - SIP, t - SIP transient, U - up,
      V - VPN orphan, W - WAAS,
      w - secondary domain backup,
      X - inspected by service module,
      x - per session, Y - director stub flow, y - backup stub flow,
      Z - Scansafe redirection, z - forwarding stub flow
ESP outside: 10.1.227.1/53744 NP Identity Ifc: 10.1.226.1/30604, , flags c, idle 0s,
uptime
1m21s, timeout 30s, bytes 7544, cluster sent/rcvd bytes 0/0, owners (0,255) Traffic
received
at interface outside Locally received: 7544 (93 byte/s) Traffic received at interface
NP
Identity Ifc Locally received: 0 (0 byte/s) UDP outside: 10.1.227.1/500 NP Identity
Ifc:
10.1.226.1/500, flags -c, idle 1m22s, uptime 1m22s, timeout 2m0s, bytes 1580, cluster
sent/rcvd bytes 0/0, cluster sent/rcvd total bytes 0/0, owners (0,255) Traffic received
at
interface outside Locally received: 864 (10 byte/s) Traffic received at interface NP
Identity
Ifc Locally received: 716 (8 byte/s)
```

연결 흐름 문제를 해결하려면, 노드에 **cluster exec show conn** 명령을 입력하여 모든 노드에 대한 연결을 우선 확인해야 합니다. 디렉터(Y), 백업(y) 및 전달자(z) 플래그가 있는 흐름을 확인합니다. 다음 예는 세 ASA 모두에 대한 172.18.124.187:22와 192.168.103.131:44727 간의 SSH 연결을

보여 줍니다. ASA1에는 연결의 전달자임을 나타내는 z 플래그가 있고, ASA3에는 연결의 디렉터임을 나타내는 Y 플래그가 있으며, ASA2에는 특별한 플래그가 없어 소유자임을 나타냅니다. 아웃바운드 방향에서 이 연결의 패킷은 ASA2의 내부 인터페이스로 들어가 외부 인터페이스를 나갑니다. 인바운드 방향에서 이 연결의 패킷은 ASA1 및 ASA3의 외부 인터페이스로 들어가 클러스터 제어 링크를 통해 ASA2로 전달된 다음 ASA2의 내부 인터페이스를 나갑니다.

```
ciscoasa/ASA1/control node# cluster exec show conn
ASA1 (LOCAL):*****
18 in use, 22 most used
Cluster stub connections: 0 in use, 5 most used
TCP outside 172.18.124.187:22 inside 192.168.103.131:44727, idle 0:00:00, bytes
37240828, flags z

ASA2:*****
12 in use, 13 most used
Cluster stub connections: 0 in use, 46 most used
TCP outside 172.18.124.187:22 inside 192.168.103.131:44727, idle 0:00:00, bytes
37240828, flags UIO

ASA3:*****
10 in use, 12 most used
Cluster stub connections: 2 in use, 29 most used
TCP outside 172.18.124.187:22 inside 192.168.103.131:44727, idle 0:00:03, bytes 0,
flags Y
```

- **show cluster info [conn-distribution | packet-distribution | loadbalance | flow-mobility counters]**

**show cluster info conn-distribution** 및 **show cluster info packet-distribution** 명령을 사용하면 모든 클러스터 노드 전체의 트래픽 분포가 표시됩니다. 이러한 명령은 외부 로드 밸런서를 평가하고 조정하는 데 유용합니다.

**show cluster info loadbalance** 명령을 사용하면 연결 리밸런싱 통계가 표시됩니다.

**show cluster info flow-mobility counters** 명령을 사용하면 EID 이동과 플로우 소유자 이동 정보가 표시됩니다. **show cluster info flow-mobility counters** 명령에 대한 내용은 다음 출력을 참조하십시오.

```
ciscoasa# show cluster info flow-mobility counters
EID movement notification received : 4
EID movement notification processed : 4
Flow owner moving requested : 2
```

- **show cluster info load-monitor [details]**

**show cluster info load-monitor** 명령은 마지막 간격 동안의 클러스터 멤버에 대한 트래픽 로드를 표시하며, 구성된 총 간격 수(기본적으로 30개)에 대한 평균도 표시합니다. **details** 키워드를 사용하여 각 간격에서 각 측정값의 값을 확인합니다.

```
ciscoasa(cfg-cluster)# show cluster info load-monitor
ID Unit Name
0 B
1 A_1
Information from all units with 20 second interval:
Unit Connections Buffer Drops Memory Used CPU Used
```

```
Average from last 1 interval:
  0      0      0      14      25
  1      0      0      16      20
Average from last 30 interval:
  0      0      0      12      28
  1      0      0      13      27
```

```
ciscoasa(cfg-cluster)# show cluster info load-monitor details
```

```
ID  Unit Name
  0  B
  1  A_1
```

```
Information from all units with 20 second interval
```

```
Connection count captured over 30 intervals:
```

```
Unit ID 0
      0      0      0      0      0      0
      0      0      0      0      0      0
      0      0      0      0      0      0
      0      0      0      0      0      0
      0      0      0      0      0      0
Unit ID 1
      0      0      0      0      0      0
      0      0      0      0      0      0
      0      0      0      0      0      0
      0      0      0      0      0      0
      0      0      0      0      0      0
```

```
Buffer drops captured over 30 intervals:
```

```
Unit ID 0
      0      0      0      0      0      0
      0      0      0      0      0      0
      0      0      0      0      0      0
      0      0      0      0      0      0
      0      0      0      0      0      0
Unit ID 1
```

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |

Memory usage(%) captured over 30 intervals:

|           |    |    |    |    |    |
|-----------|----|----|----|----|----|
| Unit ID 0 |    |    |    |    |    |
| 25        | 25 | 30 | 30 | 30 | 35 |
| 25        | 25 | 35 | 30 | 30 | 30 |
| 25        | 25 | 30 | 25 | 25 | 35 |
| 30        | 30 | 30 | 25 | 25 | 25 |
| 25        | 20 | 30 | 30 | 30 | 30 |
| Unit ID 1 |    |    |    |    |    |
| 30        | 25 | 35 | 25 | 30 | 30 |
| 25        | 25 | 35 | 25 | 30 | 35 |
| 30        | 30 | 35 | 30 | 30 | 30 |
| 25        | 20 | 30 | 25 | 25 | 30 |
| 20        | 30 | 35 | 30 | 30 | 35 |

CPU usage(%) captured over 30 intervals:

|           |    |    |    |    |    |
|-----------|----|----|----|----|----|
| Unit ID 0 |    |    |    |    |    |
| 25        | 25 | 30 | 30 | 30 | 35 |
| 25        | 25 | 35 | 30 | 30 | 30 |
| 25        | 25 | 30 | 25 | 25 | 35 |
| 30        | 30 | 30 | 25 | 25 | 25 |
| 25        | 20 | 30 | 30 | 30 | 30 |
| Unit ID 1 |    |    |    |    |    |
| 30        | 25 | 35 | 25 | 30 | 30 |
| 25        | 25 | 35 | 25 | 30 | 35 |
| 30        | 30 | 35 | 30 | 30 | 30 |
| 25        | 20 | 30 | 25 | 25 | 30 |

20                    30                    35                    30                    30                    35

• **show cluster {access-list | conn | traffic | user-identity | xlate} [options]**

전체 클러스터에 대한 집계된 데이터를 표시합니다. 사용 가능한 옵션은 데이터 유형에 따라 달라집니다.

**show cluster access-list** 명령에 대한 내용은 다음 출력을 참조하십시오.

```
ciscoasa# show cluster access-list
hitcnt display order: cluster-wide aggregated result, unit-A, unit-B, unit-C, unit-D
access-list cached ACL log flows: total 0, denied 0 (deny-flow-max 4096) alert-interval
  300
access-list 101; 122 elements; name hash: 0xe7d586b5
access-list 101 line 1 extended permit tcp 192.168.143.0 255.255.255.0 any eq www
(hitcnt=0, 0, 0, 0, 0) 0x207a2b7d
access-list 101 line 2 extended permit tcp any 192.168.143.0 255.255.255.0 (hitcnt=0,
0, 0, 0, 0) 0xfe4f4947
access-list 101 line 3 extended permit tcp host 192.168.1.183 host 192.168.43.238
(hitcnt=1, 0, 0, 0, 1) 0x7b521307
access-list 101 line 4 extended permit tcp host 192.168.1.116 host 192.168.43.238
(hitcnt=0, 0, 0, 0, 0) 0x5795c069
access-list 101 line 5 extended permit tcp host 192.168.1.177 host 192.168.43.238
(hitcnt=1, 0, 0, 1, 0) 0x51bde7ee
access-list 101 line 6 extended permit tcp host 192.168.1.177 host 192.168.43.13
(hitcnt=0, 0, 0, 0, 0) 0x1e68697c
access-list 101 line 7 extended permit tcp host 192.168.1.177 host 192.168.43.132
(hitcnt=2, 0, 0, 1, 1) 0xc1ce5c49
access-list 101 line 8 extended permit tcp host 192.168.1.177 host 192.168.43.192
(hitcnt=3, 0, 1, 1, 1) 0xb6f59512
access-list 101 line 9 extended permit tcp host 192.168.1.177 host 192.168.43.44
(hitcnt=0, 0, 0, 0, 0) 0xdc104200
access-list 101 line 10 extended permit tcp host 192.168.1.112 host 192.168.43.44
(hitcnt=429, 109, 107, 109, 104)
0xce4f281d
access-list 101 line 11 extended permit tcp host 192.168.1.170 host 192.168.43.238
(hitcnt=3, 1, 0, 0, 2) 0x4143a818
access-list 101 line 12 extended permit tcp host 192.168.1.170 host 192.168.43.169
(hitcnt=2, 0, 1, 0, 1) 0xb18dfea4
access-list 101 line 13 extended permit tcp host 192.168.1.170 host 192.168.43.229
(hitcnt=1, 1, 0, 0, 0) 0x21557d71
access-list 101 line 14 extended permit tcp host 192.168.1.170 host 192.168.43.106
(hitcnt=0, 0, 0, 0, 0) 0x7316e016
access-list 101 line 15 extended permit tcp host 192.168.1.170 host 192.168.43.196
(hitcnt=0, 0, 0, 0, 0) 0x013fd5b8
access-list 101 line 16 extended permit tcp host 192.168.1.170 host 192.168.43.75
(hitcnt=0, 0, 0, 0, 0) 0x2c7dba0d
```

모든 노드에서 사용 중인 연결의 집계된 수를 표시하려면 다음을 입력합니다.

```
ciscoasa# show cluster conn count
Usage Summary In Cluster:*****
  200 in use (cluster-wide aggregated)
  cl2 (LOCAL):*****
  100 in use, 100 most used

  cl1:*****
  100 in use, 100 most used
```

- **show asp cluster counter**

이 명령은 데이터 경로 문제를 해결하는 데 유용합니다.

## 클러스터 라우팅 모니터링

클러스터 라우팅에 대한 내용은 다음 명령을 참조하십시오.

- **show route cluster**
- **debug route cluster**

라우팅에 대한 클러스터 정보를 표시합니다.

- **show lisp eid**

EID 및 사이트 ID를 보여주는 ASA EID 테이블을 표시합니다.

**cluster exec show lisp eid** 명령의 다음 출력을 참조하십시오.

```
ciscoasa# cluster exec show lisp eid
L1 (LOCAL):*****
  LISP EID      Site ID
  33.44.33.105      2
  33.44.33.201      2
  11.22.11.1        4
  11.22.11.2        4
L2:*****
  LISP EID      Site ID
  33.44.33.105      2
  33.44.33.201      2
  11.22.11.1        4
  11.22.11.2        4
```

- **show asp table classify domain inspect-lisp**

이 명령은 트러블슈팅에 유용합니다.

## 클러스터링의 로깅 구성

클러스터링의 로깅 구성에 대한 내용은 다음 명령을 참조하십시오.

**logging device-id**

클러스터의 각 노드에서는 시스템 로그 메시지를 독립적으로 생성합니다. **logging device-id** 명령을 사용하면 디바이스 ID가 동일하거나 다른 시스템 로그 메시지를 생성하여 클러스터의 동일한 또는 다른 노드에서 메시지가 표시되도록 할 수 있습니다.

## 클러스터 인터페이스 모니터링

클러스터 인터페이스 모니터링에 대한 내용은 다음 명령을 참조하십시오.

- **show cluster interface-mode**

클러스터 인터페이스 모드를 표시합니다.

## 클러스터링 디버깅

클러스터링 디버깅에 대해서는 다음 명령을 참조하십시오.

- **debug cluster [ccp | datapath | fsm | general | hc | license | rpc | transport]**

클러스터링에 대한 디버그 메시지가 표시됩니다.

- **debug cluster flow-mobility**

클러스터링 플로우 모빌리티와 관련된 이벤트를 표시합니다.

- **debug lisp eid-notify-intercept**

eid-notify 메시지가 차단된 경우 이벤트를 표시합니다.

- **show cluster info trace**

**show cluster info trace** 명령을 사용하면 추가적인 문제 해결을 위한 디버깅 정보가 표시됩니다.

**show cluster info trace** 명령에 대한 내용은 다음 출력을 참조하십시오.

```
ciscoasa# show cluster info trace
Feb 02 14:19:47.456 [DEBUG]Receive CCP message: CCP_MSG_LOAD_BALANCE
Feb 02 14:19:47.456 [DEBUG]Receive CCP message: CCP_MSG_LOAD_BALANCE
Feb 02 14:19:47.456 [DEBUG]Send CCP message to all: CCP_MSG_KEEPALIVE from 80-1 at
CONTROL_NODE
```

예를 들어 **local-unit** 이름이 동일한 두 노드가 제어 노드로 작동하고 있음을 보여주는 다음 메시지가 표시된다면, 두 노드의 **local-unit** 이름이 동일하거나(구성 확인), 노드에서 자체 브로드캐스트 메시지를 수신(네트워크 확인)하고 있음을 의미할 수 있습니다.

```
ciscoasa# show cluster info trace
May 23 07:27:23.113 [CRIT]Received datapath event 'multi control_nodes' with parameter
1.
May 23 07:27:23.113 [CRIT]Found both unit-9-1 and unit-9-1 as control_node units.
Control_node role retained by unit-9-1, unit-9-1 will leave then join as a Data_node
May 23 07:27:23.113 [DEBUG]Send event (DISABLE, RESTART | INTERNAL-EVENT, 5000 msecs,
Detected another Control_node, leave and re-join as Data_node) to FSM. Current state
CONTROL_NODE
May 23 07:27:23.113 [INFO]State machine changed from state CONTROL_NODE to DISABLED
```

## 클러스터링에 대한 참조

이 섹션에는 클러스터링이 작동하는 방식에 대한 자세한 정보가 포함되어 있습니다.



## ASA 기능 및 클러스터링

일부 ASA 기능은 ASA 클러스터링이 지원되지 않으며, 일부 기능은 제어 노드에서만 지원됩니다. 기타 기능의 경우 올바르게 사용하는 데 필요한 주의 사항이 있을 수 있습니다.

### 클러스터링으로 지원되지 않는 기능

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

- TLS 프록시를 사용하는 Unified Communication 기능
- 원격 액세스 VPN(SSL VPN 및 IPsec VPN)
- Virtual Tunnel Interface(VTI)
- 다음과 같은 애플리케이션 감시:
  - CTIQBE
  - H323, H225, RAS
  - IPsec 통과
  - MGCP
  - MMP
  - RTSP
  - SCCP(Skinny)
  - WAAS
  - WCCP
- 봇넷 트래픽 필터
- Auto Update Server
- DHCP 클라이언트, 서버, 프록시 DHCP 릴레이가 지원됩니다.
- VPN 로드 밸런싱
- Azure에서 페일오버
- 통합 라우팅 및 브리징
- FIPS mode(FIPS 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.

• 다음과 같은 애플리케이션 감시:

- DCERPC
  - ESMTTP
  - IM
  - NetBIOS
  - PPTP
  - RADIUS
  - RSH
  - SNMP
  - SQLNET
  - SUNRPC
  - TFTP
  - XDMCP
- 고정 경로 모니터링
  - 네트워크 액세스에 대한 인증 및 권한 부여. 어카운팅이 분산됨
  - 필터링 서비스
  - Site-Site VPN
  - 멀티캐스트 라우팅

## 개별 노드에 적용되는 기능

이러한 기능은 전체 클러스터 또는 제어 노드가 아닌 각 ASA 노드에 적용됩니다.

- QoS — QoS 정책은 구성 복제의 일부로 클러스터 전체와 동기화됩니다. 그러나 정책은 각 노드에서 독립적으로 시행됩니다. 예를 들어, 출력에 대한 정책 시행을 구성할 경우 특정 ASA에 있

는 트래픽에서 적응 속도 및 적응 버스트 값이 시행됩니다. 3개 노드로 구성되고 트래픽이 균일하게 분산된 클러스터의 경우, 적응 속도는 클러스터 속도의 3배가 됩니다.

- 위협 탐지 — 위협 탐지는 각 노드에서 독립적으로 작동됩니다. 예를 들어, 상위 통계는 노드별로 적용됩니다. 포트 검사 탐지 기능의 경우, 검사 트래픽이 모든 노드 간에 로드 밸런싱되고 한 노드에 모든 트래픽이 표시되지 않으므로 이 기능은 작동하지 않습니다.

## 네트워크 액세스 및 클러스터링용 AAA

네트워크 액세스용 AAA는 인증, 권한 부여, 어카운팅이라는 세 가지 구성 요소로 이루어져 있습니다. 인증 및 권한 부여는 클러스터 데이터 노드에 대한 데이터 구조의 복제를 통해 클러스터링 제어 노드에서 중앙 집중식 기능으로 구현됩니다. 제어 노드가 선택된 경우, 새 제어 노드에서는 설정된 인증 완료 사용자 및 관련 인증 작업을 중단 없이 계속 가동하는 데 필요한 모든 정보를 보유하게 됩니다. 사용자 인증의 유효 및 절대 시간 제한은 제어 노드가 변경될 경우 유지됩니다.

어카운팅은 클러스터에서 분산된 기능으로 구현됩니다. 어카운팅은 플로우 기준으로 수행되므로, 플로우에 대한 어카운팅이 구성되면 플로우를 소유한 클러스터 노드에서는 어카운팅 시작 및 중지 메시지를 AAA 서버에 보냅니다.

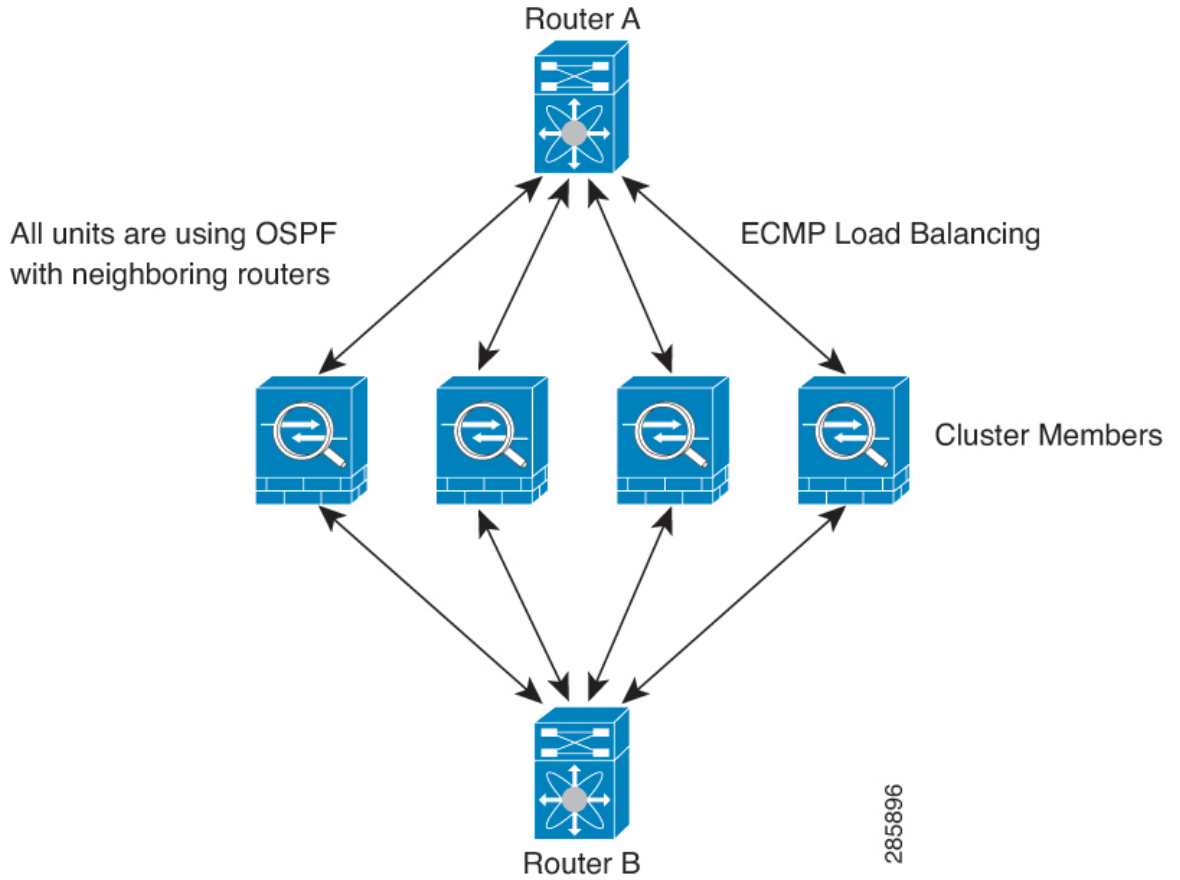
## 연결 설정 및 클러스터링

연결 제한은 클러스터 전체에서 시행됩니다(**set connection conn-max**, **set connection embryonic-conn-max**, **set connection per-client-embryonic-max** 및 **set connection per-client-max** 명령). 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 8: 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.

EIGRP는 개별 인터페이스 모드에서 클러스터 피어와 네이버 관계를 형성하지 않습니다.



참고 클러스터가 이중화를 위해 동일한 라우터의 여러 위치에 인접하는 경우 비대칭 라우팅으로 인해 트래픽이 너무 많이 손실될 수 있습니다. 비대칭 라우팅을 피하려면 모든 노드 인터페이스를 동일한 트래픽 영역으로 그룹화하십시오.

### FTP 및 클러스터링

- 다른 클러스터 멤버가 FTP 데이터 채널 및 제어 채널의 흐름을 소유한 경우, 데이터 채널 소유자 유닛에서는 유희 시간 제한 업데이트를 제어 채널 소유자에게 주기적으로 전송하고 유희 시간 제한 값을 업데이트합니다. 그러나 제어 흐름 소유자가 다시 로드되고 제어 흐름이 다시 호스팅

된 경우, 부모/자식 흐름 관계가 더 이상 유지되지 않으며 제어 흐름 유효 시간 제한도 업데이트 되지 않습니다.

- FTP 액세스용 AAA를 사용할 경우 제어 노드에서는 제어 채널 플로우를 중앙 집중화합니다.

## ICMP 검사 및 클러스터링

클러스터를 통과하는 ICMP 및 ICMP 오류 패킷의 플로우는 ICMP/ICMP 오류 검사의 활성화 여부에 따라 달라집니다. ICMP 검사를 사용하지 않으면 ICMP는 단방향 플로우이며 관리자 플로우 지원이 없습니다. ICMP 검사를 사용하면 ICMP 플로우가 양방향이며, 관리자/백업 플로우에 의해 백업됩니다. 검사된 ICMP 플로우의 한 가지 차이점은 전달된 패킷의 관리자 처리에 있습니다. 관리자는 패킷을 전달자에게 반환하는 대신 ICMP 에코 응답 패킷을 플로우 소유자에게 전달합니다.

## 멀티캐스트 라우팅 및 클러스터링

개별 인터페이스 모드에서 유닛은 멀티캐스트와 별개로 작동하지 않습니다. 모든 데이터 및 라우팅 패킷은 제어 유닛을 통해 처리되고 전달되므로, 패킷 복제가 방지됩니다.

## NAT 및 클러스터링

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 ASA에 전송할 수 있습니다. 패킷이 NAT 소유자가 아닌 ASA에 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. 스패 EtherChannel의 경우에는 하나의 IP 주소만 클러스터 인터페이스에 연결되므로 이것이 문제가 되지 않습니다.
- 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 경우에는 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 유효 시간이 업데이트되지 않습니다. 따라서 유효 시간이 유효 시간 제한을 초과할 수 있습니다. refcnt가 0인 구성된 시간 초과 값보다 큰 유효 타이머 값은 오래된 xlate를 나타냅니다.
- Per-session PAT 기능 — 클러스터링에만 해당되는 것은 아니지만, 세Per-session PAT기능을 사용하면 PAT의 확장성이 개선되며 클러스터링을 수행할 때 각 데이터 노드에서 고유한 PAT 연결을 소유할 수 있게 됩니다. 이와 달리 multi-session PAT 연결은 제어 노드에 전달해야 하며 제어 노드에서 해당 연결을 소유하게 됩니다. 기본적으로 모든 TCP 트래픽 및 UDP DNS 트래픽은 세션 단위 PAT xlate를 사용하며, 여기서 ICMP 및 기타 모든 UDP 트래픽은 멀티 세션을 사용합니다. TCP 및 UDP에 대해 이러한 기본값을 변경하도록 세션 단위 NAT 규칙을 구성할 수 있지만, ICMP에 대해서는 세션 단위 PAT를 구성할 수 없습니다. H.323, SIP, Skinny 등과 같이 다중 세션 PAT가 도움이 되는 트래픽의 경우 연결된 TCP 포트에 대해 세션 단위 PAT를 비활성화할 수 있습니다(이러한 H.323 및 SIP에 대한 UDP 포트는 기본적으로 이미 다중 세션임). 세션당 PAT에 대한 자세한 내용은 방화벽 설정 가이드를 참조하십시오.
- 다음을 검사할 수 있는 고정 PAT 없음
  - FTP
  - PPTP
  - 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.

## SCTP 및 클러스터링

로드 밸런싱으로 인해 모든 노드에서 SCTP 연결을 만들 수 있습니다. 멀티호밍 연결은 동일한 노드에 있어야 합니다.

## SIP 검사 및 클러스터링

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

TLS 프록시 구성은 지원되지 않습니다.

## SNMP 및 클러스터링

SNMP 에이전트에서는 로컬 IP 주소로 각각의 개별 ASA를 폴링합니다. 클러스터의 통합 데이터는 폴링할 수 없습니다.

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 새 노드로 복제하거나 데이터 노드에서 직접 복제하려면 제어 노드에서 다시 추가해야 합니다.

## STUN 및 클러스터링

STUN 검사는 편환이 복제될 때 장애 조치 및 클러스터 모드에서 지원됩니다. 그러나 트랜잭션 ID는 노드 간에 복제되지 않습니다. STUN 요청을 수신한 후 노드가 실패하고 다른 노드가 STUN 응답을 수신한 경우, STUN 응답은 삭제됩니다.

## 시스템 로그 및 클러스터링

- Syslog - 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.

- NetFlow — 클러스터의 각 노드에는 고유한 NetFlow 스트림이 있습니다. NetFlow 컬렉터에서는 각각의 ASA를 별도의 NetFlow 내보내기 장치로만 처리할 수 있습니다.

## 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.

## 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 표시 됩니다. 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 선택 합니다.

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 노드끼리 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.

상태 모니터링을 활성화하면 모든 물리적 인터페이스가 기본적으로 모니터링됩니다. 선택적으로 인터페이스별 모니터링을 비활성화할 수 있습니다. 명명된 인터페이스만 모니터링될 수 있습니다.

노드의 모니터링된 인터페이스에 장애가 발생하면 클러스터에서 해당 노드가 제거됩니다. ASA에서 클러스터의 멤버를 제거하기 전까지 걸리는 시간은 해당 노드가 설정된 멤버인지 또는 클러스터에 참가하는지에 따라 달라집니다. ASA에서는 노드가 클러스터에 참가하는 처음 90초 동안에는 인터페이스를 모니터링하지 않습니다. 이 시간 동안에는 인터페이스 상태가 변경되어도 ASA가 클러스터에서 제거되지 않습니다. 노드 상태와 관계없이 500ms 후에 노드가 제거됩니다.

## 실패 이후 상태

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

ASA는 실패 이벤트에 따라 클러스터에 다시 참가하려고 시도합니다.



**참고** ASA가 비활성화되고 클러스터에 자동으로 다시 조인하지 못할 경우, 모든 데이터 인터페이스가 종료되며 관리 전용 인터페이스에서만 트래픽을 주고받을 수 있습니다. 관리 인터페이스에서는 클러스터 IP 풀에서 노드로 전송된 IP 주소를 사용하여 가동 상태를 유지합니다. 그러나 다시 로드해도 노드가 클러스터에서 여전히 비활성 상태인 경우 관리 인터페이스가 비활성화됩니다. 추가 구성을 위해서는 콘솔 포트를 사용해야 합니다.

## 클러스터 다시 참가

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

- 처음 참가 시 클러스터 제어 링크 장애 - 클러스터 제어 링크의 문제를 해결한 후에는 CLI에서 **cluster group name**을 입력한 다음 **enable**을 입력하여 클러스터링을 다시 활성화함으로써 클러스터에 수동으로 다시 참가해야 합니다.
- 클러스터 참가 후 클러스터 제어 링크 장애 — ASA에서는 자동으로 5분마다 무기한으로 다시 참가하려고 시도합니다. 이 동작은 구성 가능합니다.
- 데이터 인터페이스 장애 — ASA에서는 자동으로 5분, 10분, 마지막으로 20분 후에 다시 참가하도록 시도합니다. 20분 후에도 참가가 이루어지지 않을 경우 ASA에서는 클러스터링을 비활성화합니다. 데이터 인터페이스 문제를 해결한 후에는 CLI에서 **cluster group name**을 입력한 다음 **enable**을 입력하여 클러스터링을 수동으로 활성화해야 합니다. 이 동작은 구성 가능합니다.
- 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 활성 상태이고 **enable** 명령이 계속 활성화되어 있으면 전원을 다시 가동할 때 노드가 클러스터에 다시 가입할 수 있습니다. ASA에서는 5초마다 클러스터에 다시 참가하도록 시도합니다.
- 내부 오류 — 내부 장애 포함: 애플리케이션 동기화 시간 초과, 일치하지 않는 애플리케이션 상태 등이 있습니다. 노드는 5분, 10분, 20분 간격으로 자동으로 클러스터에 다시 참가하려고 시도합니다. 이 동작은 구성 가능합니다.

## 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 2: 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     | AAA 규칙(uauth)을 포함하고합니다.          |
| IPv6 네이버 데이터베이스                           | 예     | —                                |
| 동적 라우팅                                    | 예     | —                                |
| SNMP 엔진 ID                                | 아니요   | —                                |
| Firepower 4100/9300에 대한 분산 VPN(사이트 대 사이트) | 예     | 백업 세션이 활성 세션이 되며 새 백업 세션이 생성됩니다. |

## 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 관리자로 선택하면 별도의 백업 소유자가 선택됩니다.

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

사이트 간 클러스터링에 대한 관리자 지역화를 활성화하는 경우에는 두 가지 백업 소유자 역할, 즉 로컬 백업 및 글로벌 백업이 있습니다. 소유자는 항상 자신과 동일한 사이트의 로컬 백업을

선택합니다(사이트 ID 기반). 글로벌 백업은 어느 사이트에든 있을 수 있으며, 로컬 백업과 동일한 노드일 수도 있습니다. 소유자는 연결 상태 정보를 두 백업에 모두 전송합니다.

사이트 이중화를 활성화하는 경우 백업 소유자가 소유자와 같은 사이트에 있으면 사이트 장애로부터 플로우를 보호하기 위해 다른 사이트에서 추가 백업 소유자가 선택됩니다. 새시 백업 및 사이트 백업은 서로 독립적이므로 경우에 따라서는 플로우에 새시 백업과 사이트 백업이 모두 포함됩니다.

- 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.

사이트 간 클러스터링에 대한 관리자 지역화를 활성화하는 경우에는 두 가지 관리자 역할, 즉 로컬 관리자와 전역 관리자가 있습니다. 소유자는 항상 자신과 동일한 사이트의 로컬 관리자를 선택합니다(사이트 ID 기반). 전역 관리자는 어느 사이트에든 있을 수 있으며, 로컬 관리자와 동일한 노드일 수도 있습니다. 원래 소유자가 실패하면 로컬 관리자가 동일한 사이트에서 새로운 연결 소유자를 선택합니다.

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. 관리자 지역화를 활성화하면, 전달자는 항상 로컬 관리자를 쿼리합니다. 전달자는 로컬 관리자가 소유자를 모르는 경우에만 전역 관리자를 쿼리합니다. 클러스터 멤버가 다른 사이트의 소유인 연결에 대한 패킷을 수신하는 경우를 예로 들 수 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.

연결에 PAT(Port Address Translation)가 사용되는 경우, PAT 유형(per-session 또는 multi-session)이 클러스터의 어떤 멤버가 새 연결의 소유자가 될지에 영향을 미칩니다.

- Per-session PAT(세션 단위 PAT) - 연결에서 초기 패킷을 수신하는 노드가 소유자입니다. 기본적으로 TCP 및 DNS UDP 트래픽은 per-session PAT를 사용합니다.
- Multi-session PAT(다중 세션 PAT) - 항상 제어 노드가 소유자입니다. multi-session PAT 연결이 초기에 데이터 노드에서 수신되면 데이터 노드는 해당 연결을 제어 노드로 전달합니다. 기본적으로 UDP(DNS UDP 제외) 및 ICMP 트래픽은 multi-session PAT를 사용하므로, 항상 제어 노드에서 해당 연결을 소유합니다.

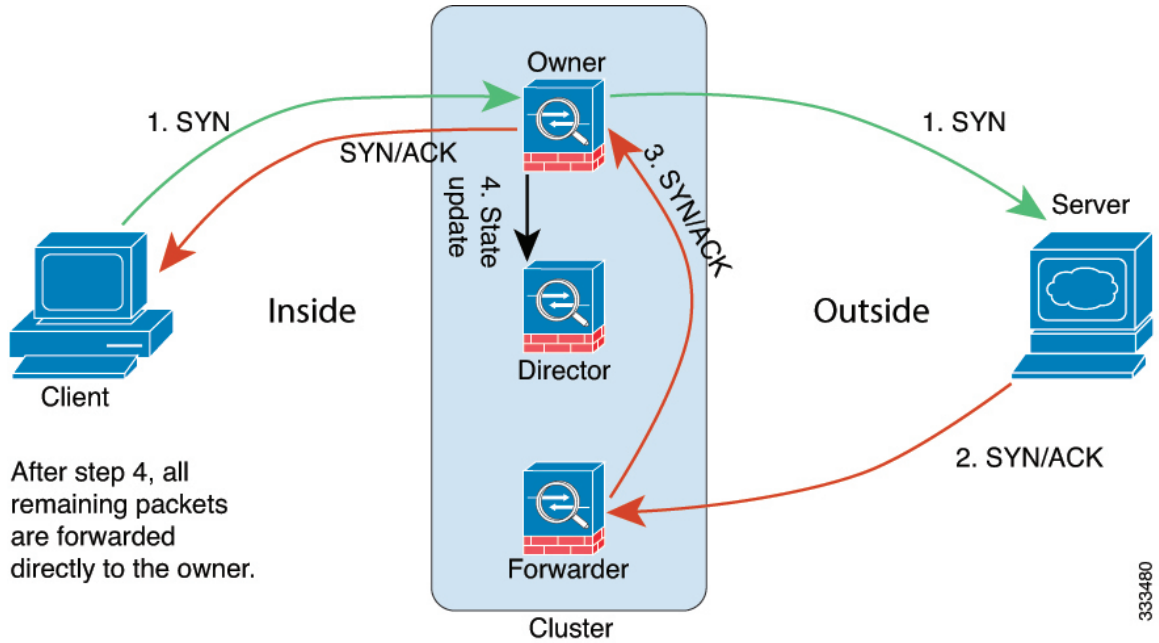
TCP 및 UDP에 대한 per-session PAT 기본값을 변경하여, 이러한 프로토콜에 대한 연결이 구성에 따라 세션 단위 또는 다중 세션으로 처리되도록 할 수 있습니다. ICMP의 경우 기본 multi-session PAT에서 변경할 수 없습니다. 세션당 PAT에 대한 자세한 내용은 방화벽 설정 가이드를 참조하십시오.

## 새 연결 소유권

로드 밸런싱을 통해 클러스터의 노드에 새 연결이 전송될 경우, 해당 노드에서는 연결의 양방향성을 모두 소유합니다. 다른 노드에 연결 패킷이 전송될 경우, 해당 패킷은 클러스터 제어 링크를 통해 소유자 노드에 전달됩니다. 최상의 성능을 실현하려면, 같은 노드에 전송될 수 있도록 흐름의 양방향에 적절한 외부 로드 밸런싱이 필요합니다. 또한 흐름은 노드 간에 균일하게 분산되어야 합니다. 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.

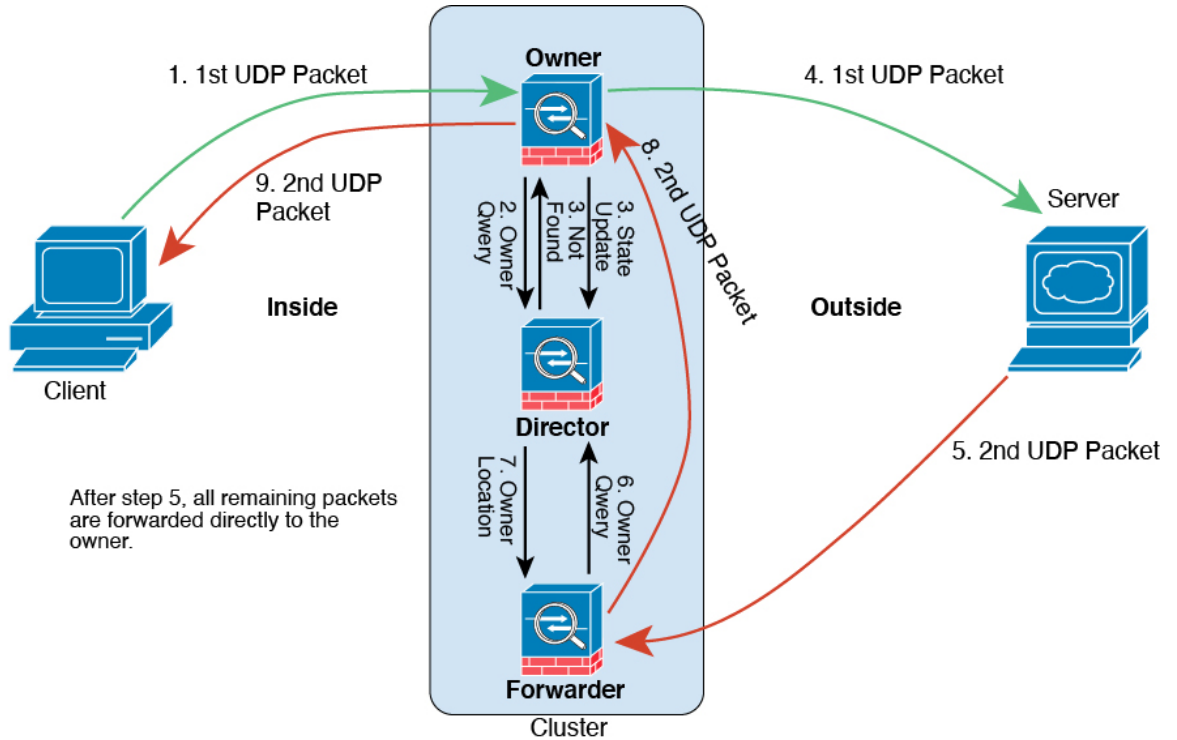


1. SYN 패킷은 클라이언트에서 시작되고 ASA에 전달(로드 밸런싱 방법을 기준으로)되며, 이 유닛이 소유자 유닛이 됩니다. 소유자 유닛에서는 흐름을 생성하고, 소유자 정보를 SYN 쿠키로 인코딩하며, 패킷을 서버에 전달합니다.
2. SYN-ACK 패킷은 서버에서 시작되고 다른 ASA에 전달(로드 밸런싱 방법을 기준으로)됩니다. 이 ASA는 전달자 유닛입니다.
3. 전달자 유닛에서는 연결을 소유하지 않으므로 SYN 쿠키에서 소유자 정보를 디코딩하고, 소유자에 대한 전달 흐름을 생성하며, SYN-ACK를 소유자 유닛에 전달합니다.
4. 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 UDP의 샘플 데이터 플로우

다음 예에는 새 연결을 설정하는 방법이 나와 있습니다.

1. 그림 9: ICMP 및 UDP 데이터 플로우



첫 번째 UDP 패킷은 클라이언트에서 시작되고 (로드 밸런싱 방법을 기준으로) ASA에 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 ASA 가상 클러스터링 기록

| 기능   | 버전      | 세부 사항   |
|--|---------|---|
| Azure의 ASAv: 게이트웨이 로드 밸런싱을 통한 클러스터링          | 9.20(2) | 이제 ARM(Azure Resource Manager) 템플릿을 사용하여 Azure에서 ASA 가상 클러스터링 구축을 지원합니다. 그런 다음 네트워크 트래픽의 로드 밸런싱에 GWLB(Gateway Load Balancer)를 사용하도록 ASAv 클러스터를 구성합니다.   |
| AWS에서 GWLB를 사용하는 ASA 가상 클러스터링을 위한 대상 페일오버 구성 | 9.20(2) | 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.   |
| 플로우 상태에 대해 구성 가능한 클러스터 keepalive 간격          | 9.20(2) | 플로우 소유자는 관리자 및 백업 소유자에게 keepalive(clu_keepalive 메시지)와 업데이트(clu_update 메시지)를 전송하여 플로우 상태를 새로 고칩니다. 이제 keepalive 간격을 설정할 수 있습니다. 기본값은 15초이며 15~55초 사이의 간격을 설정할 수 있습니다. 클러스터 제어 링크의 트래픽 양을 줄이기 위해 이 간격을 더 길게 설정할 수 있습니다.<br><br>신규/수정된 명령: <b>clu-keepalive-interval</b> |
| ASA 가상 AWS(Amazon Web Services) 클러스터링        | 9.19(1) | ASA 가상은 AWS에서 최대 16개의 노드에 대한 개별 인터페이스 클러스터링을 지원합니다. AWS 게이트웨이 로드 밸런서와 함께 또는 로드 밸런서 없이 클러스터링을 사용할 수 있습니다.  |





## 번역에 관하여

Cisco는 일부 지역에서 본 콘텐츠의 현지 언어 번역을 제공할 수 있습니다. 이러한 번역은 정보 제공의 목적으로만 제공되며, 불일치가 있는 경우 본 콘텐츠의 영어 버전이 우선합니다.