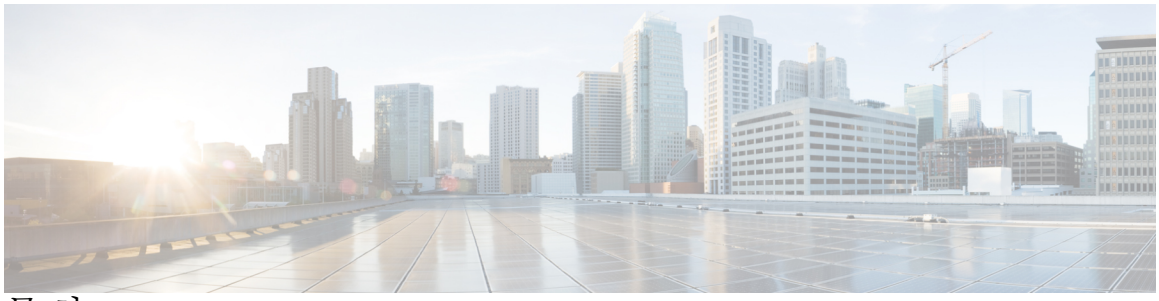
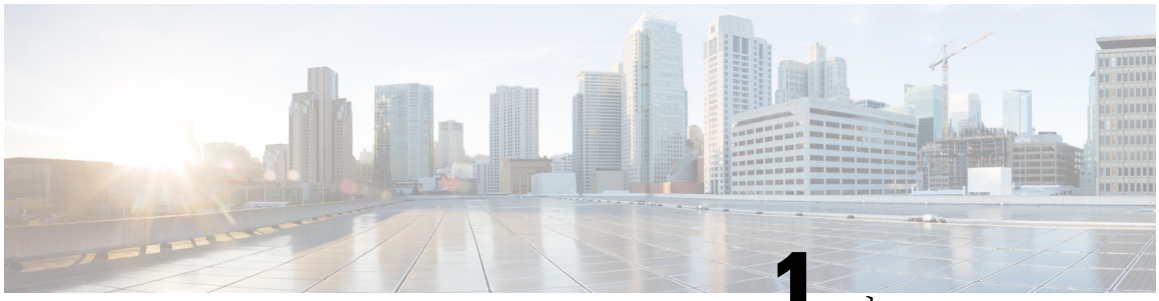
Cisco Secure Firewall ASA Virtual 시작 가이드, 9.18

초판: 2022년 6월 6일

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



목 차



1 장

Secure Firewall ASA Virtual 소개

ASA 가상(Adaptive Security Appliance Virtual)은 전체 방화벽 기능을 가상화된 환경으로 가져와 데이터 센터 트래픽과 멀티테넌트 환경을 보호합니다.

ASDM 또는 CLI를 사용하여 ASA 가상를 관리하고 모니터링할 수 있습니다. 다른 관리 옵션을 사용할 수도 있습니다.

Hypervisor 지원

하이퍼바이저 지원은 Cisco Secure Firewall ASA 호환성을 참조하십시오.

ASA 가상에 대한 라이선싱

ASA 가상에서는 Cisco Smart Software Licensing을 사용합니다. 자세한 내용은 Smart Software Licensing을 참조하십시오.

참고 ASA 가상에 스마트 라이센스를 설치해야 합니다. 라이센스를 설치할 때까지 예비 연결 테스트를 수행할 수 있도록 처리량이 100Kbps로 제한됩니다. 스마트 라이센스는 일반적인 운영에 필요합니다.

9.13(1)부터 모든 ASA 가상 라이선스는 지원되는 모든 ASA 가상 vCPU/메모리 구성에서 사용할 수 있습니다. 따라서 ASA 가상을 다양한 VM 리소스 사용 공간에서 구축할 수 있습니다. Secure Client 및 TLS 프록시에 대한 세션 제한은 모델 유형에 연결된 플랫폼 제한이 아니라 설치된 ASA 가상 플랫폼 엔타이틀먼트에 의해 결정됩니다.

지원되는 프라이빗 및 퍼블릭 구축 대상의 ASA 가상 라이선싱 엔타이틀먼트 및 리소스 사양에 대한 내용은 다음 섹션을 참조하십시오.

스마트 라이선스 엔타이틀먼트 정보

모든 ASA 가상 라이선스는 지원되는 ASA 가상 vCPU/메모리 설정에서 사용할 수 있습니다. 따라서 ASA 가상을 다양한 VM 리소스 사용 공간에서 실행할 수 있습니다. 또한 지원되는 AWS 및 Azure 인스턴스 유형의 수가 증가합니다. ASA 가상 머신을 구성할 때 지원되는 최대 vCPU 수는 16(ASAv100)이며, 지원되는 최대 메모리는 64GB RAM입니다.

중요 구축된 후에는 ASA 가상 인스턴스의 리소스 할당(메모리, CPU, 디스크 공간)을 변경할 수 없습니다. 어떤 이유로든 리소스 할당을 늘려야 한다면(예: 라이선스가 부여된 엔타이틀먼트를 ASAv30/2Gbps에서 ASAv50/10Gbps로 변경), 필요한 리소스를 이용해 새 인스턴스를 만들어야 합니다.

- vCPUs - ASA 가상은 vCPU 1~16개를 지원합니다.
- 메모리 - ASA 가상은 2GB~64GB의 RAM을 지원합니다.
- 디스크 스토리지 - ASA 가상은 기본적으로 최대 8GB의 가상 디스크를 지원합니다. 디스크 크기를 8GB 이상으로 늘릴 수 없습니다. VM 리소스를 프로비저닝할 때는 이 사실을 유의하십시오.

중요 ASA 가상의 최소 메모리 요구 사항은 2GB입니다. 현재 ASA 가상이 2GB 미만의 메모리로 실행되는 경우에는 ASA 가상 머신의 메모리를 늘리지 않고는 이전 버전에서 9.13(1) 이상으로 업그레이드할 수 없습니다. 최신 버전의 새 ASA 가상 머신을 재구축할 수도 있습니다.

2개 이상의 vCPU로 ASA 가상을 구축할 경우 ASA 가상의 최소 메모리 요구 사항은 4GB입니다.

라이선스 기능에 대한 세션 제한

Secure Client 및 TLS Proxy의 세션 제한은 설치된 ASA 가상 플랫폼 엔타이틀먼트 계층에 따라 결정되고, 속도 제한기를 통해 적용됩니다. 다음 테이블에는 엔타이틀먼트 계층 및 속도 제한기에 따른 세션 제한이 요약되어 있습니다.

표 ***1:*** 엔타이틀먼트에 따른 ***ASA*** 가상 세션 제한

자격	**Secure Client** 프리미엄 피어	총 **TLS** 프록시 세션	레이트 리미터:
표준 계층, 100M	50	500	150 Mbps
표준 계층, 1G	250	500	1Gbps
표준 계층, 2G	750	1000	2Gbps

자격	Secure Client 프리미엄 피어	총 TLS 프록시 세션	레이트 리미터:
표준 계층, 10G	10,000	10,000	10Gbps
표준 계층, 20G	20,000	20,000	20Gbps

위 표에 나와 있는 것처럼, 엔타이틀먼트에 의해 부여된 세션 제한은 플랫폼의 세션 제한을 초과할 수 없습니다. 플랫폼 세션 제한은 ASA 가상에 프로비저닝된 메모리의 양을 기반으로 합니다. 최대 ASA 가상 머신 크기는 vCPU 8개와 64GB 메모리입니다.

표 *2:* 메모리 요구 사항에 따른 ***ASA*** 가상 세션 제한

프로비저닝된 메모리	Secure Client 프리미엄 피어	총 TLS 프록시 세션
2GB~7.9GB	250	500
8GB~15.9GB	750	1000
16GB~31.9 GB	10,000	10,000
32GB~64GB	20,000	20,000

플랫폼 제한

방화벽 연결, 동시 및 VLAN은 ASA 가상 메모리를 기반으로 하는 플랫폼 제한입니다.

참고 ASA 가상이 라이선스가 해제된 상태라면 방화벽 연결을 100으로 제한합니다. 엔타이틀먼트로 라이선스가 부여되면 연결에는 플랫폼 제한이 적용됩니다. ASA 가상의 최소 메모리 요구 사항은 2GB입니다.

표 *3:* 플랫폼 제한

ASA 가상 메모리	방화벽 연결, 동시	VLAN
2GB~7.9GB	100,000	50
8GB~15.9GB	500,000	200
16GB~31.9	2,000,000	1024
32GB~64GB	4,000,000	1024

ASA 가상 프라이빗 클라우드 엔타이틀먼트(VMware, KVM, Hyper-V)

지원되는 모든 ASA 가상 vCPU/메모리 구성에서 모든 ASA 가상 라이선스를 사용할 수 있으므로, 프라이빗 클라우드 환경(VMware, KVM, Hyper-V)에서 ASA 가상을 더 유연하게 구축할 수 있습니다.

참고 ASAv50 및 ASAv100은 HyperV에서 지원되지 않습니다.

Secure Client 및 TLS 프록시의 세션 제한은 설치된 ASA 가상 플랫폼 엔타이틀먼트 계층에 따라 결정되고, 속도 제한기를 통해 적용됩니다. 다음 표에는 적용된 속도 제한기를 이용해 프라이빗 클라우드 환경에 구축된 ASA 가상의 엔타이틀먼트를 기반으로 하는 세션 제한이 요약되어 있습니다.

참고 ASA 가상 세션 제한은 ASA 가상에 프로비저닝된 메모리의 양을 기반으로 합니다. 표 2: 메모리 요구 사항에 따른 ASA 가상 세션 제한, 3 페이지를 참조하십시오.

표 *4: VMware/KVM/HyperV* 프라이빗 클라우드의 ***ASA*** 가상 - 엔타이틀먼트에 따른 라이선스 기능 제한

RAM(GB)		엔타이틀먼트 지원*				
최소	최대	표준 계층, **100M**	표준 계층, **1G**	표준 계층, **2G**	표준 계층, **10G**	표준 계층, **20G**
2	7.9	50/500/100M	250/500/1G	250/500/2G	250/500/10G	250/500/20G
8	15.9	50/500/100M	250/500/1G	750/1000/2G	750/1000/10G	750/1000/20G
16	31.9	50/500/100M	250/500/1G	750/1000/2G	10,000/10,000/10G	10K/10K/20G
32	64	50/500/100M	250/500/1G	750/1000/2G	10,000/10,000/10G	20K/20K/20G
*Secure Client 엔타이틀먼트/인스턴스당 세션/TLS 프록시 세션/속도 제한기						

ASA 가상 퍼블릭 클라우드 엔타이틀먼트(AWS)

지원되는 모든 ASA 가상 vCPU/메모리 구성에서 모든 ASA 가상 라이선스를 사용할 수 있으므로, 다양한 AWS 인스턴스 유형에서 ASA 가상을 구축할 수 있습니다. Secure Client 및 TLS Proxy의 세션 제한은 설치된 ASA 가상 플랫폼 엔타이틀먼트 계층에 따라 결정되고, 속도 제한기를 통해 적용됩니다.

다음 표에는 AWS 인스턴스 유형을 위한 엔타이틀먼트 계층에 따른 세션 제한이 요약되어 있습니다. 지원되는 인스턴스의 AWS VM 차원(vCPU 및 메모리)에 대한 자세한 내용은 "AWS 클라우드에서의 ASA 가상 구축 정보"를 참조하십시오.

표 *5: AWS*의 ***ASA*** 가상 - 엔타이틀먼트 기준 라이선스 기능 제한

Instance	**BYOL** 엔타이틀먼트 지원*				**PAYG****
	표준 계층, **100M**	표준 계층, **1G**	표준 계층, **2G**	표준 계층, **10G**	
c5.xlarge	50/500/100M	250/500/1G	750/1000/2G	750/1000/10G	750/1000
c5.2xlarge	50/500/100M	250/500/1G	750/1000/2G	10K/10K/10G	10K/10K
c4.large	50/500/100M	250/500/1G	250/500/2G	250/500/10G	250/500

Instance	BYOL 엔타이틀먼트 지원*				PAYG**
	표준 계층, 100M	표준 계층, 1G	표준 계층, 2G	표준 계층, 10G	
c4.xlarge	50/500/100M	250/500/1G	250/500/2G	250/500/10G	250/500
c4.2xlarge	50/500/100M	250/500/1G	750/1000/2G	10,000/10,000/10G	750/1000
c3.large	50/500/100M	250/500/1G	250/500/2G	250/500/10G	250/500
c3.xlarge	50/500/100M	250/500/1G	250/500/2G	250/500/10G	250/500
c3.2xlarge	50/500/100M	250/500/1G	750/1000/2G	10K/10K/10G	750/1000
m4.large	50/500/100M	250/500/1G	250/500/2G	250/500/10G	250/500
m4.xlarge	50/500/100M	250/500/1G	250/500/2G	250/500/10G	10K/10K
m4.2xlarge	50/500/100M	250/500/1G	750/1000/2G	10K/10K/10G	10,000/10,000
*Secure Client 엔타이틀먼트/인스턴스당 세션/TLS 프록시 세션/속도 제한기. **Secure Client 세션 / TLS 프록시 세션. 속도 제한기는 PAYG 모드로는 사용하지 않습니다.					

종량제**(PAYG)** 모드

다음 표에는 할당된 메모리를 기반으로 하는 시간당 청구(PAYG) 모드의 각 계층에 대한 Smart Licensing 엔타이틀먼트가 요약되어 있습니다.

표 ***6: AWS***에서의 ***ASA*** 가상 ***- PAYG***를 위한 스마트 라이선스 자격

RAM(GB)	시간당 청구 모드 엔타이틀먼트
2GB ~ 8GB 미만	표준 계층, 1G
8GB~16GB 미만	표준 계층, 2G
16GB~64GB	표준 계층, 10G

ASA 가상 퍼블릭 클라우드 엔타이틀먼트(Azure)

지원되는 모든 ASA 가상 vCPU/메모리 구성에서 모든 ASA 가상 라이선스를 사용할 수 있으므로, 다양한 Azure 인스턴스 유형에서 ASA 가상을 구축할 수 있습니다. Secure Client 및 TLS Proxy의 세션 제한은 설치된 ASA 가상 플랫폼 엔타이틀먼트 계층에 따라 결정되고, 속도 제한기를 통해 적용됩니다.

다음 표에는 Azure 인스턴스 유형을 위한 엔타이틀먼트 계층에 따른 세션 제한이 요약되어 있습니다. 지원되는 인스턴스의 Azure VM 차원(vCPU 및 메모리)에 대한 자세한 내용은 "Microsoft Azure Cloud 클라우드에서의 ASA 가상 구축 정보"를 참조하십시오.

참고 PAYG(종량과금제) 모드는 현재 Azure의 ASA 가상에서 지원되지 않습니다.

표 *7: Azure*의 *ASA* 가상 - 엔타이틀먼트 기준 라이선스 기능 제한

Instance	BYOL 엔타이틀먼트 지원*				
	표준 계층, **100M**	표준 계층, **1G**	표준 계층, **2G**	표준 계층, **10G**	표준 계층, **20G**
D1, D1_v2DS1, DS1_v2	50/500/100M	250/500/1G	250/500/2G	250/500/10G	250/500/20G
D2, D2_v2, DS2, DS2_v2	50/500/100M	250/500/1G	250/500/2G	250/500/10G	250/500/20G
D3, D3_v2, DS3, DS3_v2	50/500/100M	250/500/1G	750/1000/2G	750/1000/10G	750/1000/20G
D4, D4_v2, DS4, DS4_v2	50/500/100M	250/500/1G	750/1000/2G	10,000/10,000/10G	10K/10K/20G
D5, D5_v2, DS5, DS5_v2	50/500/100M	250/500/1G	750/1000/2G	10,000/10,000/10G	10K/20K/20G
D2_v3	50/500/100M	250/500/1G	750/1000/2G	750/1000/10G	750/1000/20G
D4_v3	50/500/100M	250/500/1G	750/1000/2G	10,000/10,000/10G	10K/10K/20G
D8_v3	50/500/100M	250/500/1G	750/1000/2G	10,000/10,000/10G	10K/10K/20G
F4, F4s	50/500/100M	250/500/1G	750/1000/2G	750/1000/10G	750/1000/20G
F8, F8s	50/500/100M	250/500/1G	750/1000/2G	10,000/10,000/10G	10K/20K/20G
F16, F16s	50/500/100M	250/500/1G	750/1000/2G	10,000/10,000/10G	10K/20K/20G
*Secure Client 엔타이틀먼트/인스턴스당 세션/TLS 프록시 세션/속도 제한기					

지침 및 제한 사항

ASA 가상 방화벽 기능은 ASA 하드웨어 방화벽과 매우 유사하지만 다음과 같은 지침 및 제한 사항이 적용됩니다.

ASA 가상 (모든 엔타이틀먼트)에 대한 지침 및 제한 사항

Smart Licensing 지침

- 지원되는 최대 vCPU 수는 8개입니다. 지원되는 최대 메모리는 64GB RAM입니다. 모든 ASA 가상 라이선스는 지원되는 ASA 가상 vCPU/메모리 설정에서 사용할 수 있습니다.
- 라이선스 받은 기능 및 라이선스를 받지 않은 플랫폼 기능에 대한 세션 제한은 VM 메모리의 양을 기준으로 설정됩니다.
- Secure Client 및 TLS 프록시의 세션 제한은 ASA 가상 플랫폼 엔타이틀먼트에 의해 결정됩니다. 세션 제한은 더 이상 ASA 가상 모델 유형(ASAv5/10/30/50/100)과 연관되지 않습니다.
- 세션 제한에는 최소 메모리 요구 사항이 있습니다. VM 메모리가 최소 요구 사항보다 낮으면 세션 제한은 메모리 양에서 지원되는 최대 수를 기준으로 설정됩니다.
- 기존 엔타이틀먼트는 변경되지 않습니다. 엔타이틀먼트 SKU 및 표시 이름에는 모델 번호(ASAv5/10/30/50/100)가 계속 포함됩니다.
- 엔타이틀먼트는 속도 제한기를 통해 최대 처리량을 설정합니다.
- 고객 주문 프로세스는 변경되지 않습니다.

디스크 스토리지

ASA 가상는 기본적으로 최대 8GB의 가상 디스크를 지원합니다. 디스크 크기를 8GB 이상으로 늘릴 수 없습니다. VM 리소스를 프로비저닝할 때는 이 사실을 유의하십시오.

컨텍스트 모드 지침

단일 컨텍스트 모드에서만 지원됩니다. 다중 상황 모드는 지원되지 않습니다.

고가용성을 위한 페일오버 지침

장애 조치 구축의 경우, 대기 유닛에 동일한 라이선스 엔타이틀먼트가 있는지 확인합니다. 예를 들어 두 유닛 모두 2Gbps 엔타이틀먼트가 있어야 합니다.

중요 ASA 가상을 이용해 고가용성 쌍을 만들 때는 동일한 순서로 각 ASA 가상에 데이터 인터페이스를 추가해야 합니다. 각 ASA 가상에 동일한 인터페이스를 추가했지만 순서가 다른 경우 ASA 가상 콘솔에서 오류가 나타날 수 있습니다. 페일오버 기능도 영향을 받을 수 있습니다.

지원되지 않는 ASA 기능

ASA 가상는 다음 ASA 기능을 지원하지 않습니다.

- 클러스터링(KVM 및 VMware를 제외한 모든 엔타이틀먼트용)
- 다중 컨텍스트 모드

- 활성/활성 장애 조치
- EtherChannel
- AnyConnect Premium 라이센스 공유

제한 사항

- ASA 가상은 x710 NIC용 1.9.5 i40en 호스트 드라이버와 호환되지 않습니다. 이전 또는 최신 드라이버 버전을 사용해야 합니다. (VMware만 해당)

1GB 엔타이틀먼트에 대한 지침 및 제한 사항

성능 지침

- 9개 이상의 e1000 인터페이스가 구성된 1GB 플랫폼에서 점보 프레임을 예약하면 디바이스가 다시 로드될 수 있습니다. 점보 프레임 예약이 활성화된 상태라면 인터페이스 수를 8개 이하로 줄이십시오. 인터페이스의 정확한 수는 구성된 다른 기능의 작업에 필요한 메모리의 양에 따라 다르며, 8보다 적을 수 있습니다.

10GB 엔타이틀먼트에 대한 지침 및 제한 사항

성능 지침

- 집계된 트래픽 10Gbps를 지원합니다.
- ASA 가상 성능을 개선하기 위한 다음 방법을 지원합니다.
  - Numa 노드
  - 여러 RX 대기열
  - SR-IOV 프로비저닝
  - 자세한 내용은 VMware의 ASA 가상에 대한 성능 조정, 37 페이지 및 KVM의 ASA 가상에 대한 성능 조정, 58 페이지를 참고하십시오.
- 전체 처리량을 달성하려면 CPU 고정을 하는 것이 좋습니다. ESXi 컨피그레이션의 성능 향상, 37 페이지 및 KVM 컨피그레이션의 성능 향상, 58 페이지를 참조하십시오.
- e1000 및 i40e-vf 인터페이스가 혼합된 점보 프레임 예약은 i40e-vf 인터페이스 중단을 유발할 수 있습니다. 점보 프레임 예약이 활성화된 경우 e1000 및 i40e-vf 드라이버를 사용하는 인터페이스 유형을 혼합하지 마십시오.

제한 사항

- 투명 모드는 지원되지 않습니다.

- ASA 가상은 x710 NIC용 1.9.5 i40en 호스트 드라이버와 호환되지 않습니다. 이전 또는 최신 드라이버 버전을 사용해야 합니다. (VMware만 해당)
- Hyper-V에서는 지원되지 않습니다.

20GB 엔타이틀먼트에 대한 지침 및 제한 사항

성능 지침

- 집계된 트래픽 20Gbps를 지원합니다.
- ASA 가상 성능을 개선하기 위한 다음 방법을 지원합니다.
  - Numa 노드
  - 여러 RX 대기열
  - SR-IOV 프로비저닝
  - 자세한 내용은 VMware의 ASA 가상에 대한 성능 조정, 37 페이지 및 KVM의 ASA 가상에 대한 성능 조정, 58 페이지를 참고하십시오.
- 전체 처리량을 달성하려면 CPU 고정을 하는 것이 좋습니다. ESXi 컨피그레이션의 성능 향상, 37 페이지 및 KVM 컨피그레이션의 성능 향상, 58 페이지를 참조하십시오.

제한 사항

- ASA 가상은 x710 NIC용 1.9.5 i40en 호스트 드라이버와 호환되지 않습니다. 이전 또는 최신 드라이버 버전을 사용해야 합니다. (VMware만 해당)
- 투명 모드는 지원되지 않습니다.
- AWS(Amazon Web Services) 및 Hyper-V에서는 지원되지 않습니다.

ASA 가상 인터페이스 및 가상 NIC

가상화 플랫폼의 게스트인 ASA 가상에서는 기본 물리적 플랫폼의 네트워크 인터페이스를 사용합니다. 각 ASA 가상 인터페이스는 가상 NIC(vNIC)에 매핑됩니다.

- ASA 가상 인터페이스
- 지원되는 vNIC

ASA 가상 인터페이스

ASA 가상에는 다음과 같은 기가비트 이더넷 인터페이스가 포함되어 있습니다.

- 관리 0/0

  AWS 및 Azure의 경우 Management 0/0이 트래픽을 전달하는 "외부" 인터페이스일 수 있습니다.

- GigabitEthernet 0/0에서 0/8까지 포함. GigabitEthernet 0/8은 ASA 가상를 장애 조치 쌍의 일부분으로 구축할 경우 장애 조치 링크에 사용됩니다.

참고 컨파그레이션 마이그레이션을 간소화하기 위해, VMXNET3 드라이버에서 사용할 수 있는 것과 같은 10 GigabitEthernet 인터페이스에는 GigabitEthernet이라는 레이블이 지정됩니다. 단순 식별 용도이며 실제 인터페이스 속도에는 영향을 주지 않습니다.

ASA 가상에서는 E1000 드라이버를 1Gbps 링크로 사용하는 GigabitEthernet 인터페이스를 정의합니다. VMware는 더 이상 E1000 드라이버 사용을 권장하지 않습니다.

- Hyper-V는 최대 8개의 인터페이스를 지원합니다. Management 0/0 및 GigabitEthernet 0/0 ~ 0/6입니다. GigabitEthernet 0/6를 장애 페일오버로 사용할 수 있습니다.

지원되는 vNIC

ASA 가상에서는 다음 vNIC를 지원합니다. 동일한 ASA 가상에서 e1000 및 vmxnet3 같은 vNIC를 혼합하는 것은 지원되지 않습니다.

표 *8:* 지원되는 *vNic*

vNIC 유형	Hypervisor 지원		ASA 가상 버전	참고
	VMWare	KVM		
vmxnet3	예	아니요	9.9(2) 이상	VMware 기본값 vmxnet3를 사용할 경우 LRO(Large Receive Offload)를 비활성화하여 TCP 성능 저하를 방지해야 합니다. VMware 및 VMXNET3에 대해 LRO 비활성화, 11 페이지의 내용을 참조하십시오.
e1000	예	예	9.2(1) 이상	VMware에서는 권장하지 않습니다.
virtio	아니요	예	9.3(2.200) 이상	KVM 기본값
ixgbe-vf	예	예	9.8(1) 이상	AWS 기본값. SR-IOV 지원을 위한 ESXi 및 KVM.
i40e-vf	아니요	예	9.10(1) 이상	SR-IOV 지원을 위한 KVM.

VMware 및 VMXNET3에 대해 LRO 비활성화

LRO(Large Receive Offload)는 CPU 오버헤드를 줄여 고대역폭 네트워크 연결의 인바운드 처리량을 늘리는 기술입니다. 단일 스트림에서 여러 수신 패킷을 더 큰 버퍼로 집계한 다음 네트워킹 스택의 더 높은 곳으로 전달하기 때문에, 처리해야 하는 패킷 수가 줄어듭니다. 그러나 LRO는 네트워크 패킷 전달이 일관적이지 않고 혼잡한 네트워크에서 "버스트"되는 TCP 성능 문제을 유발할 수 있습니다.

중요 VMware는 기본적으로 LRO를 활성화하여 전체 처리량을 늘립니다. 따라서 이 플랫폼에서는 ASA 가상 구축에 대해 LRO를 비활성화해야 합니다.

ASA 가상 머신에서 LRO를 바로 비활성화할 수 있습니다. 컨피그레이션을 변경하기 전에 가상 머신의 전원을 끕니다.

1. vSphere Web Client 인벤토리에서 ASA 가상 머신을 찾습니다.
    1. 가상 머신을 찾으려면 데이터 센터, 폴더, 클러스터, 리소스 풀 또는 호스트를 선택합니다.
    2. **Related Objects(**관련 개체**)** 탭을 클릭하고 **Virtual Machines(**가상 머신**)**를 클릭합니다.
2. 가상 머신을 마우스 오른쪽 버튼으로 클릭하고 **Edit Settings(**설정 수정**)**를 선택합니다.
3. **VM Options(VM** 옵션**)**를 클릭합니다.
4. **Advanced(**고급**)**를 확장합니다.
5. Configuration Parameters(컨피그레이션 매개변수) 아래에서 **Edit Configuration(**컨피그레이션 편집**)** 버튼을 클릭합니다.
6. **Add Parameter(**매개변수 추가**)**를 클릭하고 LRO 매개변수의 이름과 값을 입력합니다.
    - Net.VmxnetSwLROSL | 0
    - Net.Vmxnet3SwLRO | 0
    - Net.Vmxnet3HwLRO | 0
    - Net.Vmxnet2SwLRO | 0
    - Net.Vmxnet2HwLRO | 0

참고 (선택 사항) LRO 매개변수가 있다면 값을 검사하고 필요에 따라 변경할 수 있습니다. 매개변수가 1이면 LRO가 활성화됩니다. 0이면 LRO가 비활성화됩니다.

7. **OK(**확인**)**를 클릭하여 변경 사항을 저장하고 **Configuration Parameters(**컨피그레이션 매개변수**)** 대화상자를 닫습니다.
8. **Save(**저장**)**를 클릭합니다.

자세한 내용은 다음 VMware 지원 문서를 참조하십시오.

- VMware KB 1027511
- VMware KB 2055140

ASA 가상 및 SR-IOV 인터페이스 프로비저닝

SR-IOV(단일 루트 I/O 가상화)를 사용하면 다양한 게스트 운영 체제를 실행하는 여러 VM이 호스트 서버 내에서 단일 PCIe 네트워크 어댑터를 공유할 수 있습니다. SR-IOV를 사용하면 VM이 네트워크 어댑터에서 또는 네트워크 어댑터로 직접 데이터를 이동할 수 있으며, 이때 하이버파이저를 우회하여 네트워크 처리량을 높이고 서버 CPU 부담을 낮춥니다. 최신 x86 서버 프로세서는 Intel VT-d 기술과 같은 칩셋 개선 사항이 포함되어 SR-IOV에서 요구하는 직접 메모리 전송 및 기타 작업을 용이하게 합니다.

SR-IOV 사양은 다음 두 가지 디바이스 유형을 정의합니다.

- PF(물리적 기능) - 본질적으로 정적 NIC인 PF는 SR-IOV 기능을 포함하는 풀 PCIe 디바이스입니다. PF는 일반 PCIe 디바이스로 검색, 관리 및 구성됩니다. 단일 PF는 가상 기능(VF) 집합에 대한 관리 및 구성을 제공할 수 있습니다.
- VF(가상 기능) - 동적 vNIC와 마찬가지로, VF는 데이터 이동에 필요한 리소스를 하나 이상 제공하는 풀 또는 경량 가상 PCIe 디바이스입니다. VF는 직접 관리되지는 않지만 PF에서 파생되어 관리됩니다. VM에 하나 이상의 VF를 할당할 수 있습니다.

SR-IOV는 PCI 표준을 개발하고 관리하기 위해 설립된 업계 조직인 PCI SIG(Peripheral Component Interconnect Special Interconnect Group)에서 정의하고 유지 관리합니다. SR-IOV에 대한 자세한 내용은 PCI-SIG SR-IOV 입문: SR-IOV 기술 소개를 참고하십시오.

ASA 가상에서 SR-IOV 인터페이스를 프로비저닝하려면 적절한 운영 체제 수준, 하드웨어 및 CPU, 어댑터 유형, 어댑터 설정 등을 계획해야 합니다.

SR-IOV 인터페이스에 대한 지침 및 제한 사항

ASA 가상 구축에 사용되는 구체적인 하드웨어는 크기 및 사용 요구 사항에 따라 달라질 수 있습니다. ASA 가상에 대한 라이선싱, 1 페이지에서는 다양한 ASA 가상 플랫폼에 대한 라이선스 자격과 일치하는 규정 준수 리소스 시나리오에 대해 설명합니다. 또한 SR-IOV 가상 기능에는 특정 시스템 리소스가 필요합니다.

호스트 운영 체제 및 하이퍼바이저 지원

SR-IOV 지원 및 VF 드라이버는 다음에 사용할 수 있습니다.

- Linux 2.6.30 커널 이상

SR-IOV 인터페이스를 이용하는 ASA 가상은 현재 다음 하이퍼바이저에서 지원됩니다.

- VMware vSphere/ESXi

- QEMU/KVM
- AWS

하드웨어 플랫폼 지원

참고 지원되는 가상화 플랫폼을 실행할 수 있는 아무 서버 클래스 x86 CPU 디바이스에 ASA 가상을 구축해야 합니다.

이 섹션에서는 SR-IOV 인터페이스 관련 하드웨어 지침에 대해 설명합니다. 이러한 지침은 요구 사항이 아니라 지침이지만, 지침을 충족하지 않는 하드웨어를 사용하면 기능 문제가 발생하거나 성능이 저하될 수 있습니다.

SR-IOV를 지원하며 SR-IOV 지원 PCIe 어댑터를 장착한 서버가 필요합니다. 다음 하드웨어 고려 사항을 파악해야 합니다.

- 사용 가능한 VF 수, 벤더 및 디바이스 간 차이를 포함한 SR-IOV NIC 기능.
- 모든 PCIe 슬롯이 SR-IOV를 지원하지는 않습니다.
- SR-IOV 지원 PCIe 슬롯에 다양한 기능이 있을 수 있습니다.

참고 시스템의 SR-IOV 지원 여부는 제조업체의 설명서를 참조하십시오.

- VT-d 지원 칩셋, 마더보드 및 CPU의 경우, 가상화 지원 IOMMU 지원 하드웨어 페이지에서 정보를 찾을 수 있습니다. VT-d는 SR-IOV 시스템의 필수 BIOS 설정입니다.
- VMware의 경우 온라인 호환성 가이드에서 SR-IOV 지원을 검색할 수 있습니다.
- KVM의 경우에는 CPU 호환성을 확인할 수 있습니다. KVM의 ASA 가상에서는 x86 하드웨어만 지원합니다.

참고 Cisco에서는 ASA 가상을 UCS C-Series 랙 서버를 이용해 테스트했습니다. Cisco UCS-B 서버는 ixgbe-vf vNIC를 지원하지 않습니다.

지원되는 **SR-IOV**용 **NIC**

- Intel 이더넷 네트워크 어댑터 X710

주의 ASA 가상은 x710 NIC용 1.9.5 i40en 호스트 드라이버와 호환되지 않습니다. 이전 또는 최신 드라이버 버전을 사용해야 합니다. (VMware만 해당)

- Intel 이더넷 서버 어댑터 X520 - DA2

CPU

- x86_64 멀티코어 CPU

  Intel 샌디브리지 이상(권장)

참고 ASA 가상은(는) Intel 브로드웰 CPU(E5-2699-v4) 2.3GHz에서 테스트를 완료했습니다.

- 코어
  - CPU 소켓당 최소 8개의 물리적 코어
  - 8개 코어가 단일 소켓에 있어야 합니다.

참고 ASAv50 및 ASAv100에서 전체 처리량을 달성하려면 CPU 고정을 하는 것이 좋습니다. ESXi 컨피그레이션의 성능 향상, 37 페이지 및 KVM 컨피그레이션의 성능 향상, 58 페이지를 참조하십시오.

BIOS 설정

SR-IOV는 BIOS 및 하드웨어에서 실행 중인 운영 체제 인스턴스/하이퍼바이저에서의 지원을 필요로 합니다. 시스템 BIOS에서 다음 설정을 확인합니다.

- SR-IOV가 활성화됨
- VT-x(Virtualization Technology)가 활성화됨
- VT-d가 활성화됨
- (선택 사항) 하이퍼스레딩이 비활성화됨

시스템마다 BIOS 설정에 액세스하고 변경하는 방법이 다르므로, 벤더 설명서를 참조하여 프로세스를 확인하는 것이 좋습니다.

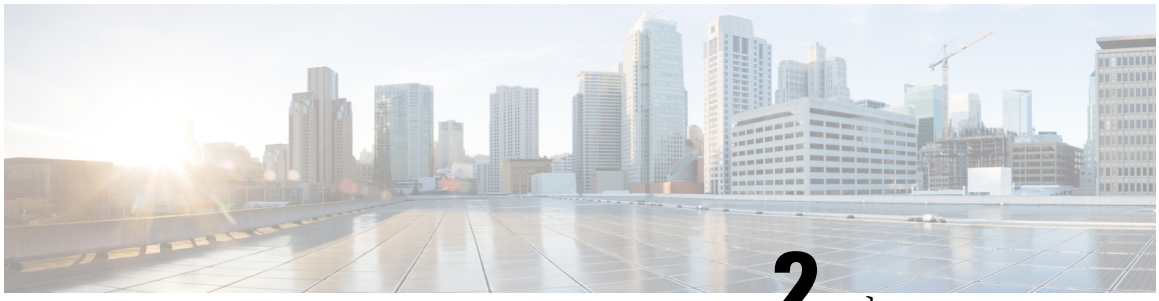
제한 사항

ixgbe-vf 인터페이스를 사용할 때는 다음 제한 사항에 유의해야 합니다.

- 게스트 VM은 VF를 무차별 모드로 설정할 수 없습니다. 따라서 ixgbe-vf를 사용할 때는 투명 모드가 지원되지 않습니다.
- 게스트 VM은 VF에서 MAC 주소를 설정할 수 없습니다. 따라서 MAC 주소는 다른 ASA 플랫폼에서나 다른 인터페이스 유형을 사용할 때처럼 HA 중에 전송되지는 않습니다. HA 페일오버는 IP 주소를 액티브에서 스탠바이로 전송하여 작동합니다.

참고 이 제한은 i40e-vf 인터페이스에도 적용됩니다.

- Cisco UCS-B 서버는 ixgbe-vf vNIC를 지원하지 않습니다.
- 페일오버 설정에서 페어링된 ASA 가상(기본 유닛)에 장애가 발생하면, 스탠바이 ASA 가상 유닛이 기본 유닛 역할을 수행하며 인터페이스 IP 주소는 스탠바이 ASA 가상 유닛의 새 MAC 주소로 업데이트됩니다. 그런 다음 ASA 가상은 동일한 네트워크의 다른 디바이스에 인터페이스 IP 주소의 MAC 주소 변경 사항을 알리기 위해 무료 ARP(Address Resolution Protocol) 업데이트를 전송합니다. 그러나 이러한 유형의 인터페이스와의 비호환성으로 인해, 인터페이스 IP 주소를 전역 IP 주소로 변환하기 위한 NAT 또는 PAT 명령문에 정의된 전역 IP 주소로 Gratuitous ARP 업데이트가 전송되지 않습니다.



2 장

VMware를 사용하여 ASA 가상 구축

VMware ESXi를 실행할 수 있는 서버 클래스 x86 CPU 디바이스에서 ASA 가상을 구축할 수 있습니다.

중요 ASA 가상의 최소 메모리 요구 사항은 2GB입니다. 현재 ASA 가상이 2GB 미만의 메모리로 실행되는 경우에는 ASA 가상 머신의 메모리를 늘리지 않고는 이전 버전에서 9.13(1) 이상으로 업그레이드할 수 없습니다. 최신 버전의 새 ASA 가상 머신을 재구축할 수도 있습니다.

VMware의 ASA 가상 지침 및 제한 사항

ESXi 서버에서 여러 ASA 가상 인스턴스를 생성하고 구축할 수 있습니다. ASA 가상 구축에 사용되는 특정 하드웨어는 구축된 인스턴스 수 및 사용 요구 사항에 따라 달라질 수 있습니다. 생성하는 각 가상 어플라이언스는 호스트 머신에서 최소 리소스 할당(메모리, CPU 수 및 디스크 공간)을 필요로 합니다.

중요 ASA 가상는 8GB 디스크 스토리지 크기로 구축됩니다. 디스크 공간의 리소스 할당은 변경할 수 없습니다.

ASA 가상을 구축하기 전에 다음 지침 및 제한 사항을 검토하십시오.

VMware ESXi의 ASA 가상 시스템 요구 사항

최적의 성능을 보장하려면 아래 사양을 준수해야 합니다. ASA 가상ASA 가상에는 다음 요구 사항이 적용됩니다.

- 호스트 CPU는 가상화 확장을 사용하는 서버 클래스 x86 기반 Intel 또는 AMD CPU여야 합니다.

  예를 들어 ASA 가상 성능 테스트 랩에서는 2.6GHz에서 실행되는 Intel® Xeon® CPU E5-2690v4 프로세서를 사용하는 Cisco UCS®(Cisco Unified Computing System™) C 시리즈 M4 서버가 최소 요구 사항입니다.

- ASA 가상은 ESXi 버전 6.0, 6.5, 6.7 및 7.0을 지원합니다.

권장 vNIC

최적의 성능을 위해 다음 vNIC를 사용하는 것이 좋습니다.

- PCI 패스스루의 i40e - 서버의 물리적 NIC를 VM 전용으로 지정하고 DMA(Direct Memory Access)를 통해 NIC와 VM 간에 패킷 데이터를 전송합니다. 패킷 이동에는 CPU 사이클이 필요하지 않습니다.

- i40evf/ixgbe-vf - 위와 사실상 동일하지만(NIC와 VM 간의 DMA 패킷) 여러 VM에서 NIC를 공유할 수 있습니다. 뛰어난 구축 유연성 때문에 일반적으로 SR-IOV를 선호합니다. 지침 및 제한 사항, 42 페이지의 내용을 참조하십시오.

- vmxnet3 - 10Gbps 작업을 지원하지만 CPU 사이클이 필요한 반가상화 네트워크 드라이버입니다. VMware 기본값입니다.

  vmxnet3를 사용할 경우 LRO(Large Receive Offload)를 비활성화하여 TCP 성능 저하를 방지해야 합니다.

성능 최적화

ASA 가상에서 최상의 성능을 얻으려면 VM과 호스트를 모두 조정할 수 있습니다. 자세한 내용은 VMware의 ASA 가상에 대한 성능 조정, 37 페이지를 참조하십시오.

- **NUMA** - 게스트 VM의 CPU 리소스를 단일 NUMA(Non-Uniform Memory Access) 노드로 격리하면 ASA 가상 성능을 개선할 수 있습니다. 자세한 내용은 NUMA 지침, 38 페이지를 참조하십시오.

- **Receive Side Scaling** — ASA 가상는 RSS(Receive Side Scaling)를 지원합니다. RSS는 네트워크 수신 트래픽을 여러 프로세서 코어로 분산하기 위해 네트워크 어댑터에서 활용하는 기술입니다. 버전 9.13(1) 이상에서 지원됩니다. 자세한 내용은 RSS(Receive Side Scaling)를 위한 다중 RX 대기열, 39 페이지를 참조하십시오.

- **VPN** 최적화 - ASA 가상을 사용한 VPN 성능 최적화를 위한 추가 고려 사항은 VPN 최적화, 63 페이지를 참조하십시오.

클러스터링

버전 9.17부터는 VMware에 구축된 ASA 가상 인스턴스에서 클러스터링이 지원됩니다. 자세한 내용은 ASAv용 ASA 클러스터를 참조하십시오.

OVF 파일 지침

asav-vi.ovf 또는 asav-esxi.ovf 파일은 구축 대상에 따라 선택합니다.

- asav-vi—vCenter에서 구축할 경우
- asav-esxi—ESXi에서 구축할 경우(vCenter 없음)
- ASA 가상 OVF 구축은 현지화(영어 이외의 언어 모드로 구성 요소 설치)를 지원하지 않습니다. 사용자 환경의 VMware vCenter와 LDAP 서버가 ASCII 호환 모드로 설치되어 있는지 확인해 주십시오.
- ASA 가상를 설치하고 VM 콘솔을 사용하려면 먼저 키보드를 영어(미국)로 설정해야 합니다.
- ASA 가상이 구축되면 두 개의 서로 다른 ISO 이미지가 ESXi 하이퍼바이저에 마운트됩니다.
  - 마운트된 첫 번째 드라이브에는 vSphere에서 생성된 OVF 환경 변수가 있습니다.
  - 마운트된 두 번째 드라이브는 day0.iso입니다.

주의 ASA 가상 머신이 부팅된 후에는 두 드라이브 모두를 마운트 해제할 수 있습니다. 그러나 **Connect at Power On(**전원을 켤 때 연결**)**이 선택되지 않은 경우에도, ASA 가상의 전원을 끄거나 켤 때마다 드라이브 1이(OVF 환경 변수를 이용해) 항상 마운트됩니다.

OVF 템플릿 내보내기 지침

vSphere의 OVF 템플릿 내보내기를 사용하면 기존 ASA 가상 인스턴스 패키지를 OVF 템플릿으로서 내보낼 수 있습니다. 내보낸 OVF 템플릿을 사용하여 동일하거나 다른 환경에서 ASA 가상 인스턴스를 구축할 수 있습니다. vSphere에서 내보낸 OVF 템플릿을 사용하여 ASA 가상 인스턴스를 구축하기 전에, OVF 파일에서 구성 세부 정보를 수정하여 구축 실패를 방지해야 합니다.

내보낸 ASA 가상의 OVF 파일을 수정합니다.

1. OVF 템플릿을 내보낸 로컬 머신에 로그인합니다.
2. 텍스트 편집기에서 OVF 파일을 검색하고 엽니다.
3. `<vmw:ExtraConfig vmw:key="monitor_control.pseudo_perfctr" vmw:value="TRUE"></vmw:ExtraConfig>` 태그가 존재하는지 확인합니다.
4. `<rasd:ResourceSubType>vmware.cdrom.iso</rasd:ResourceSubType>` 태그를 삭제합니다.

   또는

`<rasd:ResourceSubType>vmware.cdrom.iso</rasd:ResourceSubType>` 태그를 `<rasd:ResourceSubType>vmware.cdrom.remotepassthrough</rasd:ResourceSubType>` 태그로 교체합니다.

자세한 내용은 VMware에서 게시한 VMware 툴이 설치된 경우 vCenter Server 5.1/5.5에서의 OVF 구축 실패(2034422)를 참조하십시오.

5. `UserPrivilege`, `OvfDeployment` 및 `ControllerType`에 대한 속성 값을 입력하십시오.

   예를 들면 다음과 같습니다.

```
- <Property ovf:qualifiers="ValueMap{"ovf", "ignore", "installer"}" ovf:type="string"
ovf:key="OvfDeployment">
+ <Property ovf:qualifiers="ValueMap{"ovf", "ignore", "installer"}" ovf:type="string"
ovf:key="OvfDeployment" ovf:value="ovf">

- <Property ovf:type="string" ovf:key="ControllerType">
+ <Property ovf:type="string" ovf:key="ControllerType" ovf:value="ASAv">

- <Property ovf:qualifiers="MinValue(0) MaxValue(255)" ovf:type="uint8"
ovf:key="UserPrivilege">
+ <Property ovf:qualifiers="MinValue(0) MaxValue(255)" ovf:type="uint8"
ovf:key="UserPrivilege" ovf:value="15">
```

6. OVF 파일을 저장합니다.
7. OVF 템플릿을 사용하여 ASA 가상을 구축합니다. VMware vSphere Web Client를 사용하여 ASA 가상 구축을 참조하십시오.

고가용성을 위한 페일오버 지침

페일오버 구축의 경우, 대기 유닛에 동일한 라이선스 자격이 있는지 확인합니다. 예를 들어 두 유닛 모두 2Gbps 엔타이틀먼트가 있어야 합니다.

중요 ASA 가상을 이용해 고가용성 쌍을 만들 때는 동일한 순서로 각 ASA 가상에 데이터 인터페이스를 추가해야 합니다. 각 ASA 가상에 동일한 인터페이스를 추가했지만 순서가 다른 경우 ASA 가상 콘솔에서 오류가 나타날 수 있습니다. 페일오버 기능도 영향을 받을 수 있습니다.

ASA 가상 내부 인터페이스 또는 ASA 가상 페일오버 고가용성 링크에 사용하는 ESX 포트 그룹의 경우, 2개의 가상 NIC(액티브 업링크용 하나, 스탠바이 업링크용 하나)를 사용하여 ESX 포트 그룹 페일오버 순서를 구성합니다. 두 VM이 서로 ping하거나 ASA 가상 고가용성 링크를 가동하려면 이렇게 해야 합니다.

vMotion 지침

- VMware에서 vMotion을 사용하려면 공유 스토리지만 사용해야 합니다. 호스트 클러스터가 있는 경우, ASA 가상을 구축하는 동안 특정 호스트에 로컬로 스토리지를 프로비저닝하거나 공유 호스트에 스토리지를 프로비저닝할 수 있습니다. 그러나 ASA 가상에서 다른 호스트에 대한 vMotion을 실행하려고 하는 경우, 로컬 스토리지를 사용하면 오류가 발생합니다.

처리량 및 라이선싱을 위한 메모리 및 **vCPU** 할당

- ASA 가상에 할당된 메모리의 크기는 처리량 수준에 따라 지정됩니다. 다른 처리량 수준에 대한 라이선스를 요청할 때가 아니라면, Edit Settings(설정 수정) 대화 상자에서 메모리 설정이나 vCPU 하드웨어 설정을 변경하지 마십시오. 프로비저닝 부족은 성능에 영향을 줄 수 있습니다.

참고 메모리 또는 vCPU 하드웨어 설정을 변경해야 하는 경우 ASA 가상에 대한 라이선싱, 1 페이지에 나와 있는 값만 사용해야 합니다. VMware 권장 메모리 컨피그레이션 최소값, 기본값, 최대값을 사용하지 마십시오.

CPU 예약

- 기본적으로 ASA 가상을 위해 예약된 CPU는 1000MHz입니다. 공유, 예약 및 제한 설정(Edit Settings(설정 수정) > Resources(리소스) > CPU)을 사용하여 ASA 가상에 할당된 CPU 리소스의 양을 변경할 수 있습니다. ASA 가상이 낮은 설정을 사용하는 필수 트래픽 로드 이하인 동안 필요한 목적을 수행할 수 있는 경우, 1000Mhz에서 CPU 예약 설정을 낮게 설정하는 작업을 수행할 수 있습니다. ASA 가상에서 사용된 CPU 양은 실행 중인 하드웨어 플랫폼과 수행 중인 작업의 유형 및 양에 따라 달라집니다.

  CPU Usage(CPU 사용량(MHz)) 차트에서 모든 가상 머신에 대한 호스트의 CPU 사용량 관점을 확인할 수 있습니다. 이 차트는 가상 머신의 Performance(성능) 탭의 Home(홈) 보기에 있습니다. ASA 가상이 일반적인 트래픽 볼륨을 처리 중인 경우 CPU 사용량에 대한 벤치마크를 설정하면 CPU 예약을 조정할 때 해당 정보를 입력으로 사용할 수 있습니다.

  자세한 내용은 VMware에서 공개한 CPU 성능 개선 사항 조언을 참조하십시오.

- ASA 가상 **show vm** 및 **show cpu** 명령이나 ASDM**Home(**홈**)** > **Device Dashboard(**디바이스 대시보드**)** > **Device Information(**디바이스 정보**)** > **Virtual Resources(**가상 리소스**)** 탭 또는 **Monitoring(**모니터링**)** > **Properties(**속성**)** > **System Resources Graphs(**시스템 리소스 그래프**)** > **CPU** 창을 사용하여 리소스 할당 및 과도하거나 부족하게 프로비저닝된 모든 리소스를 확인할 수 있습니다.

투명 모드의 **UCS B** 및 시리즈 하드웨어 지침

Cisco UCS B 시리즈 하드웨어에서 투명 모드로 실행되는 일부 ASA 가상 컨피그레이션에서 MAC 플랩이 관찰되었습니다. MAC 주소가 다른 위치에서 표시되면 패킷이 손실됩니다.

다음 지침은 VMware 환경에서 투명 모드로 ASA 가상을 구축할 때 MAC 플랩을 방지하는 데 도움이 됩니다.

- VMware NIC 팀 구성 - UCS B 시리즈에서 투명 모드로 ASA 가상을 구축하는 경우, 내부 및 외부 인터페이스에 사용하는 포트 그룹에는 활성 업링크 1개만 있어야 하며, 이 업링크는 동일해야 합니다. vCenter에서 VMware NIC 팀을 구성할 수 있습니다.

  NIC 팀 구성 방법에 대한 자세한 내용은 VMware 설명서를 참조하십시오.

- ARP 검사 - ASA 가상에서 ARP 검사를 활성화하고, ARP 검사를 수신해야 하는 인터페이스에서 MAC 및 ARP 항목을 정적으로 구성합니다. ARP 검사 및 검사를 활성화하는 방법은 Cisco Secure Firewall ASA 시리즈 일반 운영 구성 가이드를 참조하십시오.

추가 지침 및 제한

- ESXi 6.7, vCenter 6.7, ASA Virtual 9.12 이상을 실행 중인 경우 ASA Virtual은 두 CD/DVD IDE 드라이브 없이 부팅됩니다.
- vSphere Web 클라이언트는 ASA 가상 OVF 구축에 지원되지 않습니다. 대신 vSphere 클라이언트를 사용하십시오.

VMware 기능 지원 - ASA 가상

다음 표에는 ASA 가상에 지원되는 VMware 기능이 나와 있습니다.

표 *9: VMware* 기능 지원 *- ASA* 가상

기능	설명	지원(예/아니요)	코멘트
Cold Clone	복제하는 동안 VM의 전원이 꺼집니다.	예	–
DRS	동적 리소스 예약 및 DPM(Distributed Power Management)에 사용됩니다.	예	VMware 지침을 참조합니다.
Hot add	추가하는 동안 VM이 실행됩니다.	아니오	–
Hot clone	복제하는 동안 VM이 실행됩니다.	아니오	–
Hot removal	제거하는 동안 VM이 실행됩니다.	아니오	–
Snapshot	VM이 몇 초간 중지됩니다.	예	주의해서 사용해야 합니다. 트래픽이 손실될 수 있습니다. 장애 조치가 발생할 수 있습니다.
일시 중지 및 재개	VM이 일시 중지되었다가 재개됩니다.	예	–
vCloud Director	VM의 자동 구축을 허용합니다.	아니오	–

기능	설명	지원(예/아니요)	코멘트
VM 마이그레이션	마이그레이션하는 동안 VM의 전원이 꺼집니다.	예	–
vMotion	VM의 라이브 마이그레이션에 사용됩니다.	예	공유 스토리지를 사용합니다. vMotion 지침, 20 페이지을 참조하십시오.
VMware FT	VM의 HA에 사용됩니다.	아니오	ASA 가상 머신 장애에는 ASA 가상 페일오버를 사용합니다.
VMware HA	ESXi 및 서버 장애에 사용됩니다.	예	ASA 가상 머신 장애에는 ASA 가상 장애 조치를 사용합니다.
VM 하트비트를 지원하는 VMware HA	VM 장애에 사용됩니다.	아니오	ASA 가상 머신 장애에는 ASA 가상 장애 조치를 사용합니다.
VMware vSphere 독립 실행형 Windows 클라이언트	VM을 구축하는 데 사용됩니다.	예	–
VMware vSphere Web Client	VM을 구축하는 데 사용됩니다.	예	–

ASA 가상 및 VMware 사전 요건

VMware vSphere Web Client, vSphere 독립형 클라이언트 또는 OVF 툴을 사용하여 ASA 가상을 구축할 수 있습니다. 시스템 요구 사항은 Cisco Secure Firewall ASA 호환성을 참조하십시오.

vSphere 표준 스위치에 대한 보안 정책

vSphere 스위치의 경우 계층 2 보안 정책을 수정하고 ASA 가상 인터페이스에서 사용하는 포트 그룹에 대한 보안 정책 예외를 적용할 수 있습니다. 다음 기본 설정을 확인하십시오.

- Promiscuous Mode(무차별 모드): **Reject(**거부**)**
- MAC Address Changes(MAC 주소 변경): **Accept(**허용**)**
- Forged Transmits(위조된 전송): **Accept(**허용**)**

다음 ASA 가상 컨피그레이션에 대해 이러한 설정을 수정해야 할 수도 있습니다. 자세한 내용은 vSphere 설명서를 참조하십시오.

표 10: 포트 그룹 보안 정책 예외

보안 예외	라우팅 방화벽 모드		투명 방화벽 모드	
	장애 조치 없음	장애 조치	장애 조치 없음	장애 조치
무차별 모드	<any>	<any>	수락	수락
MAC 주소 변경	<any>	수락	<any>	수락
위조된 전송	<any>	수락	수락	수락

ASA 가상 소프트웨어 압축 풀기 및 Day 0 컨피그레이션 파일 생성

ASA 가상를 실행하기 전에 Day 0 구성 파일을 준비할 수 있습니다. 이 파일은 ASA 가상을 시작할 때 적용하는 ASA 가상 컨피그레이션이 포함된 텍스트 파일입니다. 이 초기 컨피그레이션은 사용자가 선택하는 작업 디렉토리의 "day0-config"라는 이름의 텍스트 파일에 위치하며, 이 파일은 최초 부팅 시 마운트되고 읽히는 day0.iso 파일로 조작됩니다. Day 0 컨피그레이션 파일에는 최소한 관리 인터페이스를 활성화하고 공용 키 인증용 SSH 서버를 설정하는 명령이 포함되어야 할 뿐만 아니라, 완전한 ASA 컨피그레이션도 포함되어야 합니다. 빈 day0-config를 포함하는 기본 day0.iso이 이번 릴리스에 제공됩니다. 최초 부팅 동안 day0.iso 파일(사용자 정의 day0.iso 또는 기본 day0.iso)을 사용할 수 있어야 합니다.

시작하기 전에

이 예에서는 Linux를 사용하지만 Windows에도 유사한 유틸리티가 있습니다.

- 초기 구축 동안 ASA 가상 라이센스를 자동으로 적용하려면, Cisco Smart Software Manager에서 다운로드한 Smart Licensing ID(Identity) Token을 Day 0 컨피그레이션 파일과 같은 디렉토리에 있는 'idtoken'이라는 이름의 텍스트 파일로 가져옵니다.
- 가상 VGA 콘솔 대신 하이퍼바이저의 시리얼 포트에서 ASA 가상에 액세스하고 구성하려면, Day 0 컨피그레이션 파일에 콘솔 시리얼 설정을 포함하여 첫 부팅 시 시리얼 포트를 사용해야 합니다.
- 투명 모드에서 ASA 가상을 구축하려는 경우, 투명 모드에서 실행 중인 알려진 ASA 컨피그레이션 파일을 Day 0 컨피그레이션 파일로 사용해야 합니다. 이 사항은 라우팅 방화벽용 Day 0 컨피그레이션 파일에는 적용되지 않습니다.
- ESXi 하이퍼바이저에서 ISO 이미지를 마운트하는 자세한 방법은 VMware의 ASA 가상 지침 및 제한 사항, 17 페이지에 있는 OVF 파일 지침을 참조하십시오.

단계 1 Cisco.com에서 ZIP 파일을 다운로드하고 로컬 디스크에 저장합니다.

https://www.cisco.com/go/asa-software

참고 Cisco.com 로그인 및 Cisco 서비스 계약이 필요합니다.

단계 **2** 작업 디렉터리에 파일의 압축을 풉니다. 이 디렉터리의 어떤 파일도 삭제하지 마십시오. 다음 파일이 포함됩니다.

- asav-vi.ovf—vCenter 구축용
- asav-esxi.ovf—비 vCenter 구축용
- boot.vmdk—부팅 디스크 이미지
- disk0.vmdk—ASA 가상 디스크 이미지.
- day0.iso—day0-config 파일과 선택적으로 idtoken 파일을 포함하는 ISO
- asav-vi.mf—vCenter 구축용 매니페스트 파일
- asav-esxi.mf—비 vCenter 구축용 매니페스트 파일

단계 **3** "day0 config"라는 텍스트 파일에 ASA 가상에 대한 CLI 컨피그레이션을 입력합니다. 3개의 인터페이스에 대한 인터페이스 컨피그레이션 및 원하는 기타 모든 컨피그레이션을 추가합니다.

첫 줄은 ASAv 버전으로 시작해야 합니다. day0-config는 유효한 ASA 컨피그레이션이어야 합니다. day0-config를 생성하는 가장 좋은 방법은 기존 ASA 또는 ASA 가상에서 실행 중인 컨피그레이션 중 원하는 부분을 복사하는 것입니다. day0-config에서의 줄의 순서가 중요하며 기존 **show running-config** 명령 출력의 순서와 일치해야 합니다.

day0-config 파일의 두 가지 예가 있습니다. 첫 번째 예는 기가비트 이더넷 인터페이스를 이용해 ASA 가상을 구축할 때의 day0-config를 보여줍니다. 두 번째 예는 10기가비트 이더넷 인터페이스를 이용해 ASA 가상을 구축할 때의 day0-config를 보여줍니다. 이 day0-config를 바탕으로 SR-IOV 인터페이스를 이용해 ASA 가상을 구축합니다. 지침 및 제한 사항, 42 페이지를 참조하십시오.

예제:

```
ASA Version 9.4.1
!
console serial
interface management0/0
nameif management
security-level 100
ip address 192.168.1.1 255.255.255.0
no shutdown
interface gigabitethernet0/0
nameif inside
security-level 100
ip address 10.1.1.2 255.255.255.0
no shutdown
interface gigabitethernet0/1
nameif outside
security-level 0
ip address 198.51.100.2 255.255.255.0
no shutdown
http server enable
http 192.168.1.0 255.255.255.0 management
crypto key generate rsa modulus 1024
username AdminUser password paSSw0rd
ssh 192.168.1.0 255.255.255.0 management
aaa authentication ssh console LOCAL
call-home
http-proxy 10.1.1.1 port 443
license smart
```

```
feature tier standard
throughput level 2G
```

예제:

```
ASA Version 9.8.1
!
console serial
interface management 0/0
management-only
nameif management
security-level 0
ip address 192.168.0.230 255.255.255.0
!
interface GigabitEthernet0/0
nameif inside
security-level 100
ip address 10.10.10.10 255.255.255.0
!
interface GigabitEthernet0/1
nameif outside
security-level 0
ip address 10.10.20.10 255.255.255.0
!
route management 0.0.0.0 0.0.0.0 192.168.0.254
!
username cisco password cisco123 privilege 15
!
aaa authentication ssh console LOCAL
ssh 0.0.0.0 0.0.0.0 management
ssh timeout 60
ssh version 2
!
http 0.0.0.0 0.0.0.0 management
!
logging enable
logging timestamp
logging buffer-size 99999
logging buffered debugging
logging trap debugging
!
dns domain-lookup management
DNS server-group DefaultDNS
name-server 64.102.6.247
!
license smart
feature tier standard
throughput level 10G
!
crypto key generate rsa modulus 2048
```

단계 **4** (선택 사항) Cisco Smart Software Manager에서 발급한 Smart License ID 토큰 파일을 PC에 다운로드합니다.

단계 **5** (선택 사항) 다운로드 파일에서 ID 토큰을 복사하고 ID 토큰만 포함된 'idtoken'이라는 텍스트 파일에 붙여넣습니다.

ID 토큰은 ASA 가상을 Smart Licensing 서버에 자동으로 등록합니다.

단계 **6** 텍스트 파일을 ISO 파일로 전환하여 가상 CD-ROM을 생성합니다.

예제:

```
stack@user-ubuntu:-/KvmAsa$ sudo genisoimage -r -o day0.iso day0-config idtoken
I: input-charset not specified, using utf-8 (detected in locale settings)
Total translation table size: 0
Total rockridge attributes bytes: 252
```

```
Total directory bytes: 0
Path table size (byptes): 10
Max brk space used 0
176 extents written (0 MB)
stack@user-ubuntu:-/KvmAsa$
```

단계 **7** day0.iso를 위해 Linux의 새 SHA1 값을 계산합니다.

예제:

```
openssl dgst -sha1 day0.iso
SHA1(day0.iso)= e5bee36e1eb1a2b109311c59e2f1ec9f731ecb66 day0.iso
```

단계 **8** 작업 디렉터리의 asav-vi.mf 파일에 새 체크섬을 넣고 day0.iso SHA1 값을 새로 생성된 값으로 대체합니다.

예제:

```
SHA1(asav-vi.ovf)= de0f1878b8f1260e379ef853db4e790c8e92f2b2
SHA1(disk0.vmdk)= 898b26891cc68fa0c94ebd91532fc450da418b02
SHA1(boot.vmdk)= 6b0000ddebfc38ccc99ac2d4d5dbfb8abfb3d9c4
SHA1(day0.iso)= e5bee36e1eb1a2b109311c59e2f1ec9f731ecb66
```

단계 **9** ZIP 파일의 압축을 푼 디렉터리에 day0.iso 파일을 복사합니다. 기본 (비어 있는) day0.iso 파일을 덮어쓸 것입니다.

이 디렉터리에서 구축되는 VM이 있을 경우 새로 생성된 day0.iso내부의 컨피그레이션이 적용됩니다.

VMware vSphere Web Client를 사용하여 ASA 가상 구축

이 섹션에서는 VMware vSphere Web Client를 사용하여 ASA 가상를 구축하는 방법에 대해 설명합니다. Web Client에는 vCenter가 필요합니다. vCenter가 없다면 VMware vSphere 독립형 클라이언트 및 Day 0 컨피그레이션을 사용하여 ASA 가상 구축 또는 OVF 툴과 Day 0 컨피그레이션을 사용하여 ASA 가상 구축을 참조하십시오.

- vSphere Web Client에 액세스하여 클라이언트 통합 플러그인 설치, 27 페이지
- VMware vSphere Web Client를 사용하여 ASA 가상 구축, 27 페이지

vSphere Web Client에 액세스하여 클라이언트 통합 플러그인 설치

이 섹션에서는 vSphere Web Client에 액세스하는 방법에 대해 설명합니다. 또한 ASA 가상 콘솔에 액세스하는 데 필요한 클라이언트 통합 플러그인을 설치하는 방법에 대해서도 설명합니다. 일부 Web Client 기능(플러그인 포함)은 Macintosh에서 지원되지 않습니다. 전체 클라이언트 지원 정보는 VMware 웹사이트를 참조하십시오.

단계 **1** 브라우저에서 VMware vSphere Web Client를 실행합니다.

https://*vCenter_server:port***/vsphere-client/**

기본적으로 포트는 9443입니다.

단계 **2** (한 번만 실행) ASA 가상 콘솔에 액세스할 수 있도록 클라이언트 통합 플러그인을 설치합니다.

1. 로그인 화면에서 **Download the Client Integration Plug-in(**클라이언트 통합 플러그인 다운로드**)**을 클릭하여 플러그인을 다운로드합니다.
2. 브라우저를 닫고 설치 프로그램을 사용하여 플러그인을 설치합니다.
3. 플러그인이 설치되고 나면 vSphere Web Client에 다시 연결합니다.

단계 **3** 사용자 이름과 비밀번호를 입력하고 **Login(**로그인**)**을 클릭하거나, **Use Windows session authentication(Windows** 세션 인증 사용**)** 확인란(Windows에만 해당)을 선택합니다.

VMware vSphere Web Client를 사용하여 ASA 가상 구축

ASA 가상을 구축하려면 VMware vSphere Web Client(또는 vSphere 클라이언트)와 OVF(open virtualization format) 형식의 템플릿 파일을 사용합니다. vSphere Web Client에서 Deploy OVF Template(OVF 템플릿 구축) 마법사를 사용하여 ASA 가상용 Cisco 패키지를 구축할 수 있습니다. 이 마법사에서는 ASA 가상 OVF 파일의 구문을 분석하고, ASA 가상을 실행할 가상 머신을 만들며, 패키지를 설치합니다.

마법사의 단계는 대부분 VMware 표준 단계입니다. Deploy OVF Template(OVF 템플릿 구축)에 대한 자세한 내용은 VMware vSphere Web Client 온라인 도움말을 참조하십시오.

시작하기 전에

ASA 가상를 구축하기 전에 vSphere에서 하나 이상의 네트워크(관리용)를 구성해야 합니다.

단계 **1** Cisco.com에서 ASA 가상 ZIP 파일을 다운로드하여 PC에 저장합니다.

http://www.cisco.com/go/asa-software

참고 Cisco.com 로그인 및 Cisco 서비스 계약이 필요합니다.

단계 **2** vSphere Web Client **Navigator(**탐색기**)** 창에서 **vCenter**를 클릭합니다.

단계 **3** **Hosts and Clusters(**호스트 및 클러스터**)**를 클릭합니다.

단계 **4** ASA 가상를 구축할 데이터 센터, 클러스터 또는 호스트를 마우스 오른쪽 버튼으로 클릭하고 **Deploy OVF Template(OVF** 템플릿 구축**)**을 선택합니다.
Deploy OVF Template(OVF 템플릿 구축**)** 마법사가 나타납니다.

단계 **5** 마법사 화면의 지시를 따릅니다.

단계 **6** **Setup networks(**네트워크 설정**)** 화면에서 사용하려는 각 ASA 가상 인터페이스에 네트워크를 매핑합니다.

네트워크는 사전 순이 아닐 수도 있습니다. 네트워크를 찾기 어려운 경우 나중에 설정 수정 대화 상자에서 네트워크를 변경할 수 있습니다. 구축 후 ASA 가상 인스턴스를 마우스 오른쪽 버튼으로 클릭하고 **Edit Settings(**설정 수정**)**를 선택하면 **Edit Settings(**설정 수정**)** 대화 상자에 액세스할 수 있습니다. 그러나 ASA 가상 인터페이스 ID는 이 화면에 표시되지 않습니다(네트워크 어댑터 ID만 표시됨). 네트워크 어댑터 ID에 해당하는 ASA 가상 인터페이스 ID는 다음 표를 참조하십시오.

네트워크 어댑터 **ID**	**ASA** 가상 인터페이스 **ID**
네트워크 어댑터 1	Management 0/0
네트워크 어댑터 2	GigabitEthernet 0/0
네트워크 어댑터 3	GigabitEthernet 0/1
네트워크 어댑터 4	GigabitEthernet 0/2
네트워크 어댑터 5	GigabitEthernet 0/3
네트워크 어댑터 6	GigabitEthernet 0/4
네트워크 어댑터 7	GigabitEthernet 0/5
네트워크 어댑터 8	GigabitEthernet 0/6
네트워크 어댑터 9	GigabitEthernet 0/7
네트워크 어댑터 10	GigabitEthernet 0/8

모든 ASA 가상 인터페이스를 사용할 필요는 없지만 vSphere Web Client에서는 모든 인터페이스에 네트워크를 할당해야 합니다. 인터페이스를 비활성화된 상태로 두려면 ASA 가상 컨피그레이션 내에서 해당 인터페이스를 비활성화된 상태로 그대로 두면 됩니다. ASA 가상를 구축한 후 선택적으로 vSphere Web Client로 돌아가 Edit Settings(설정 수정) 대화 상자에서 추가 인터페이스를 삭제할 수 있습니다. 자세한 내용은 vSphere Web Client 온라인 도움말을 참조하십시오.

참고 페일오버/HA 구축의 경우 GigabitEthernet 0/8이 페일오버 인터페이스로 사전 설정됩니다.

단계 **7** 네트워크에서 인터넷 액세스에 HTTP 프록시를 사용하는 경우 **Smart Call Home Settings(Smart Call Home** 설정**)** 영역에서 스마트 라이센스를 위한 프록시 주소를 구성해야 합니다. 일반적으로 이 프록시는 Smart Call Home에도 사용됩니다.

단계 **8** 페일오버/HA 구축의 경우 Customize template(템플릿 사용자 정의) 화면에서 다음을 구성합니다.

- 대기 관리 IP 주소를 지정합니다.

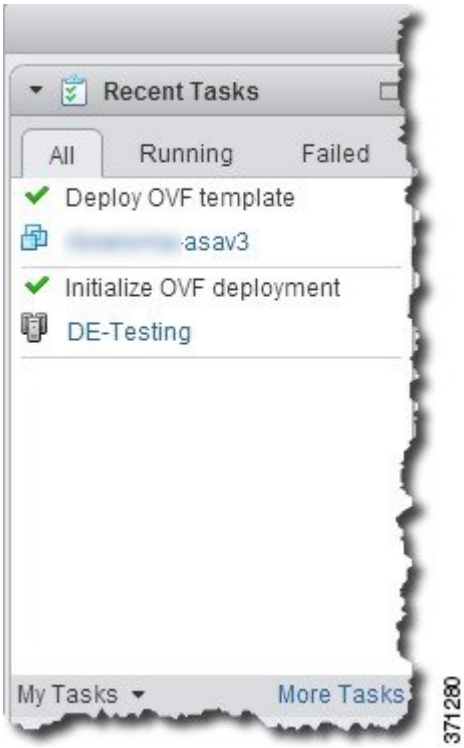
  인터페이스를 구성할 경우, 동일한 네트워크에서 액티브 IP 주소 및 스탠바이 IP 주소를 지정해야 합니다. 기본 유닛 또는 장애 조치 그룹에서 장애 조치를 시작할 경우, 보조 유닛에서는 기본 유닛의 IP 주소와 MAC 주소를 가정하고 트래픽 전달을 시작합니다. 이제 대기 상태가 된 유닛에서는 대기 IP 주소와 MAC 주소를 인수합니다. 네트워크 디바이스에서는 MAC-IP 주소 쌍의 변화가 감지되지 않으므로 네트워크 어디에서도 ARP 항목의 변경 또는 시간 초과가 발생하지 않습니다.

- **HA Connection Settings(HA** 연결 설정**)** 영역에서 페일오버 링크 설정을 구성합니다.

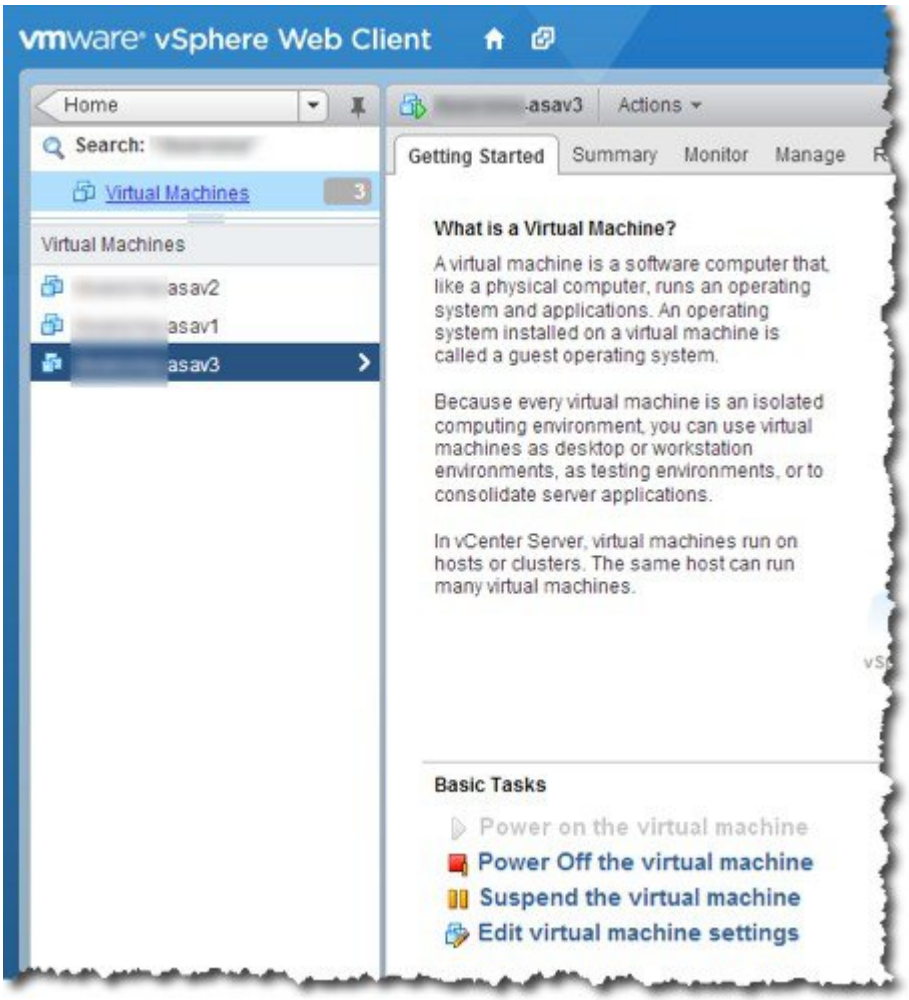
  장애 조치 쌍의 유닛 2개에서는 장애 조치 링크를 통해 지속적으로 통신을 수행하여 각 유닛의 작동 상태를 확인합니다. GigabitEthernet 0/8이 페일오버 링크로 사전 설정됩니다. 링크의 활성 및 대기 IP 주소를 같은 네트워크에 있는 주소로 입력합니다.

단계 **9** 마법사를 완료하고 나면 vSphere Web Client에서 VM을 처리합니다. **Recent Tasks(**최근 작업**)** 창의 **Global Information(**전체 정보**)** 영역에서 "Initialize OVF deployment"(OVF 구축 초기화) 상태를 확인할 수 있습니다.

작업이 완료되면 Deploy OVF Template(OVF 템플릿 구축) 완료 상태가 표시됩니다.



그런 다음 인벤토리의 지정된 데이터 센터 아래에 ASA 가상 머신 인스턴스가 표시됩니다.



단계 **10** ASA 가상 머신을 아직 실행하지 않은 경우 **Power on the virtual machine(**가상 머신 전원 켜기**)**을 클릭합니다.

ASA 가상가 부팅될 때까지 기다렸다가 ASDM 또는 콘솔에 연결합니다. ASA 가상는 처음 시작될 때 OVF 파일을 통해 제공된 매개변수를 읽어 ASA 가상 시스템 컨피그레이션에 추가합니다. 그런 다음 가동 및 실행될 때까지 자동으로 부팅을 다시 시작합니다. 이러한 이중 부팅은 ASA 가상를 처음 구축한 경우에만 발생합니다. 부팅 메시지를 보려면 **Console(**콘솔**)** 탭을 클릭하여 ASA 가상 콘솔에 액세스합니다.

단계 **11** 장애 조치/HA 구축의 경우 이 절차를 반복하여 보조 유닛을 추가합니다. 다음 지침을 참조하십시오.

- 기본 유닛과 동일한 처리량 레벨을 설정합니다.
- 기본 유닛에 정확히 동일한 *IP* 주소 설정을 입력합니다. 두 유닛의 부트스트랩 구성은 유닛을 기본 유닛 또는 보조 유닛으로 식별하는 파라미터를 제외하고 동일합니다.

다음에 수행할 작업

ASA 가상를 Cisco Licensing Authority에 성공적으로 등록하려면 ASA 가상에 인터넷 액세스가 필요합니다. 구축 후 인터넷 액세스 및 성공적인 라이선스 등록을 위해 추가 컨피그레이션이 필요할 수 있습니다.

VMware vSphere 독립형 클라이언트 및 Day 0 컨피그레이션을 사용하여 ASA 가상 구축

ASA 가상을 구축하려면 VMware vSphere 클라이언트 및 OVF(open virtualization format) 템플릿 파일(vCenter 구축은 asav-vi.ovf, 비 vCenter 구축에서는 asav-esxi.ovf)을 사용합니다. vSphere 클라이언트에서 Deploy OVF Template(OVF 템플릿 구축) 마법사를 사용하여 ASA 가상용 Cisco 패키지를 구축할 수 있습니다. 이 마법사에서는 ASA 가상 OVF 파일의 구문을 분석하고, ASA 가상을 실행할 가상 머신을 만들며, 패키지를 설치합니다.

마법사의 단계는 대부분 VMware 표준 단계입니다. Deploy OVF Template(OVF 템플릿 구축) 마법사에 대한 자세한 내용은 VMware vSphere Web Client 온라인 도움말을 참조하십시오.

시작하기 전에

- ASA 가상를 구축하기 전에 vSphere에서 하나 이상의 네트워크(관리용)를 구성해야 합니다.
- ASA 가상 소프트웨어 압축 풀기 및 Day 0 컨피그레이션 파일 생성, 24 페이지의 단계에 따라 Day 0 컨피그레이션을 생성합니다.

단계 **1** VMware vSphere 클라이언트를 실행하고 **File(파일)** > **Deploy OVF Template(OVF 템플릿 구축)**을 선택합니다.

Deploy OVF Template(OVF 템플릿 구축) 마법사가 나타납니다.

단계 **2** asav-vi.ovf 파일의 압축을 푼 작업 디렉터리로 이동하여 이 파일을 선택합니다.

단계 **3** OVF 템플릿 세부 정보가 표시됩니다. 다음 화면을 진행합니다. 사용자 지정 Day 0 컨피그레이션 파일을 사용한다면 어떤 컨피그레이션도 변경할 필요가 없습니다.

단계 **4** 구축 설정의 요약이 마지막 화면에 표시됩니다. **Finish(마침)**를 클릭하여 VM을 구축합니다.

단계 **5** ASA 가상을 켜고 VMware 콘솔을 연 다음 2번째 부팅을 기다립니다.

단계 **6** ASA 가상에 SSH를 적용하고 원하는 컨피그레이션을 완료합니다. 원하는 컨피그레이션 중 일부가 Day 0 컨피그레이션 파일에 빠졌을 경우 VMware 콘솔을 열고 필요한 컨피그레이션을 완료합니다.

이제 ASA 가상이 정상적으로 작동합니다.

OVF 툴과 Day 0 컨피그레이션을 사용하여 ASA 가상 구축

이 섹션에서는 day 0 컨피그레이션 파일이 필요한 OVF 툴을 사용하여 ASA 가상을 구축하는 방법을 설명합니다.

시작하기 전에

- OVF 툴을 사용하여 ASA 가상을 구축할 경우 day0.iso 파일이 필요합니다. ZIP 파일에 제공된 비어 있는 기본 day0.iso 파일을 사용하거나 맞춤형 Day 0 컨피그레이션 파일을 생성하여 사용할 수 있습니다. Day 0 컨피그레이션 생성에 대해서는 ASA 가상 소프트웨어 압축 풀기 및 Day 0 컨피그레이션 파일 생성, 24 페이지를 참조하십시오.
- OVF 툴이 Linux 또는 Windows PC에 설치되어 있고 대상 ESXi 서버와 연결되어 있어야 합니다.

단계 **1** OVF 툴이 설치되어 있음을 확인합니다.

예제:

```
linuxprompt# which ovftool
```

단계 **2** 원하는 구축 옵션으로 .cmd 파일을 생성합니다.

예제:

```
linuxprompt# cat launch.cmd
ovftool \
--name="asav-941-demo" \
--powerOn \
--deploymentOption=4Core8GB \
--diskMode=thin \
--datastore=datastore1 \
--acceptAllEulas \
--net:Management0-0="Portgroup_Mgmt" \
--net:GigabitEthernet0-1="Portgroup_Inside" \
--net:GigabitEthernet0-0="Portgroup_Outside" \
--prop:HARole=Standalone \
asav-esxi.ovf \
vi://root@10.1.2.3/
```

단계 **3** cmd 파일을 실행합니다.

예제:

```
linuxprompt# ./launch.cmd
```

ASA 가상이 켜집니다. 2번째 부팅될 때까지 기다립니다.

단계 **4** ASA 가상에 SSH를 적용하여 필요한 컨피그레이션을 완료합니다. 추가 컨피그레이션이 필요할 경우 ASA 가상에 대한 VMware 콘솔을 열고 필요한 컨피그레이션을 적용합니다.

이제 ASA 가상이 정상적으로 작동합니다.

ASA 가상 콘솔 액세스

ASDM을 사용할 때 경우에 따라 문제 해결에 CLI를 사용해야 할 수 있습니다. 기본적으로 내장형 VMware vSphere 콘솔에 액세스할 수 있습니다. 또는 복사 및 붙여넣기를 포함하여 더 나은 기능을 갖춘 네트워크 직렬 콘솔을 구성할 수 있습니다.

- VMware vSphere 콘솔 사용
- 네트워크 직렬 콘솔 포트 구성

참고 Day 0 컨피그레이션 파일을 사용하여 ASA 가상을 구축하는 경우, 컨피그레이션 파일에 콘솔 시리얼 설정을 포함하여 첫 부팅 시 가상 VGA 콘솔 대신 시리얼 포트를 사용할 수 있습니다. ASA 가상 소프트웨어 압축 풀기 및 Day 0 컨피그레이션 파일 생성, 24 페이지을 참조하십시오.

VMware vSphere 콘솔 사용

초기 컨피그레이션 또는 문제 해결의 경우 VMware vSphere Web Client를 통해 제공된 가상 콘솔에서 CLI에 액세스합니다. 나중에 텔넷(Telnet) 또는 SSH에 대해 CLI 원격 액세스를 구성할 수 있습니다.

시작하기 전에

vSphere Web Client의 경우 ASA 가상 콘솔에 액세스하는 데 필요한 클라이언트 통합 플러그인을 설치합니다.

단계 **1** VMware vSphere Web Client의 인벤토리에서 ASA 가상 인스턴스를 마우스 오른쪽 버튼으로 클릭하고 **Open Console(**콘솔 열기**)**을 선택합니다. 또는 Summary(요약) 탭에서 **Launch Console(**콘솔 실행**)**을 클릭합니다.

단계 **2** 콘솔을 클릭하고 **Enter** 키를 누릅니다. 참고: 커서를 놓으려면 **Ctrl+Alt** 키를 누릅니다.

ASA 가상가 여전히 시작 중인 경우 부팅 메시지가 나타납니다.

ASA 가상는 처음 시작될 때 OVF 파일을 통해 제공된 매개변수를 읽어 ASA 가상 시스템 컨피그레이션에 추가합니다. 그런 다음 가동 및 실행될 때까지 자동으로 부팅을 다시 시작합니다. 이러한 이중 부팅은 ASA 가상를 처음 구축한 경우에만 발생합니다.

참고 라이센스를 설치할 때까지 예비 연결 테스트를 수행할 수 있도록 처리량이 100Kbps로 제한됩니다. 라이센스는 일반적인 운영에 필요합니다. 또한 라이센스를 설치할 때까지 콘솔에 다음 메시지가 반복적으로 표시됩니다.

```
Warning: ASAv platform license state is Unlicensed.
Install ASAv platform license for full functionality.
```

다음 프롬프트가 표시됩니다.

```
ciscoasa>
```

이 프롬프트는 현재 사용자 EXEC 모드에 있음을 의미합니다. 사용자 EXEC 모드에서는 기본 명령만 사용 가능합니다.

단계 **3** 특권 실행 모드에 액세스합니다.

예제**:**

```
ciscoasa> enable
```

다음 프롬프트가 나타납니다.

```
Password:
```

단계 **4** **Enter** 키를 눌러 계속합니다. 기본적으로 비밀번호는 비어 있습니다. 이전에 enable 비밀번호를 설정한 경우 Enter 키를 누르는 대신 enable을 입력합니다.

프롬프트가 다음과 같이 변경됩니다.

```
ciscoasa#
```

모든 비 컨피그레이션 명령은 특권 EXEC 모드에서 사용할 수 있습니다. 또한 특권 EXEC 모드에서 컨피그레이션 모드를 입력할 수도 있습니다.

특권 모드를 종료하려면 **disable**, **exit** 또는 **quit** 명령을 입력합니다.

단계 **5** 전역 컨피그레이션 모드에 액세스합니다.

```
ciscoasa# configure terminal
```

프롬프트가 다음으로 변경됩니다.

```
ciscoasa(config)#
```

전역 컨피그레이션 모드에서 ASA 가상 컨피그레이션을 시작할 수 있습니다. 전역 컨피그레이션 모드를 종료하려면 **exit**, **quit** 또는 **end** 명령을 입력합니다.

네트워크 직렬 콘솔 포트 구성

더 나은 콘솔 경험을 위해 콘솔에 액세스할 수 있는 네트워크 직렬 포트를 단독으로 구성하거나 vSPC(Virtual Serial Port Concentrator)에 연결하여 구성할 수 있습니다. 각 방법에 대한 자세한 내용은 VMware vSphere 설명서를 참조하십시오. ASA 가상에서 가상 콘솔 대신 직렬 포트로 콘솔 출력을 보내야 합니다. 이 절차에서는 직렬 포트 콘솔을 사용하는 방법에 대해 설명합니다.

단계 **1** VMware vSphere에서 네트워크 직렬 포트를 구성합니다. VMware vSphere 설명서를 참조하십시오.

단계 **2** ASA 가상에서 disk0의 루트 디렉토리에 "use_ttyS0"이라는 파일을 만듭니다. 파일 내용은 없어도 됩니다. 이 위치에 파일이 있기만 하면 됩니다.

disk0:/use_ttyS0

- ASDM에서 **Tools(**도구**)** > **File Management(**파일 관리**)** 대화 상자를 사용하여 이 이름으로 빈 텍스트 파일을 업로드할 수 있습니다.

- vSphere 콘솔에서 파일 시스템에 있는 기존 파일(임의의 파일)을 새 이름으로 복사할 수 있습니다. 예를 들면 다음과 같습니다.

```
ciscoasa(config)# cd coredumpinfo
ciscoasa(config)# copy coredump.cfg disk0:/use_ttyS0
```

단계 **3** ASA 가상를 다시 로드합니다.

- ASDM에서 **Tools(**도구**)** > **System Reload(**시스템 다시 로드**)**를 선택합니다.
- vSphere 콘솔에서 **reload(**다시 로드**)**를 입력합니다.

  ASA 가상에서 vSphere 콘솔로 보내는 것을 중지하고 대신 직렬 콘솔로 보냅니다.

단계 **4** 직렬 포트를 추가할 때 지정한 vSphere 호스트 IP 주소와 포트 번호로 텔넷 전송하거나, vSPC IP 주소 및 포트로 텔넷 전송합니다.

vCPU 또는 처리량 라이센스 업그레이드

ASA 가상에서는 처리량 라이센스를 사용합니다. 이는 사용 가능한 vCPU 수에 영향을 미칩니다.

ASA 가상에 대한 vCPU 수를 늘리거나 줄이려면 새 라이센스를 요청하고 이를 적용한 후 VMware에서 VM 속성을 새 값과 일치하도록 변경하면 됩니다.

참고 지정된 vCPU가 ASA 가상 가상 CPU 라이센스 또는 처리량 라이센스와 일치해야 합니다. RAM도 vCPU에 맞게 크기가 지정되어야 합니다. 업그레이드하거나 다운그레이드할 때 다음 절차에 따라 라이센스 및 vCPU를 즉시 조정하십시오. 영구적인 불일치가 있는 경우 ASA 가상가 제대로 작동하지 않습니다.

단계 **1** 새 라이센스를 요청합니다.

단계 **2** 새 라이센스를 적용합니다. 장애 조치 쌍의 경우 두 유닛 모두에 새 라이센스를 적용합니다.

단계 **3** 페일오버 사용 여부에 따라 다음 중 하나를 수행합니다.

- 페일오버 - vSphere Web Client에서 스탠바이 ASA 가상의 전원을 끕니다. 예를 들어 ASA 가상을 클릭한 다음 **Power Off the virtual machine(**가상 머신 전원 끄기**)**을 클릭하거나, ASA 가상을 마우스 오른쪽 버튼으로 클릭하고 **Shut Down Guest OS(**게스트 **OS** 종료**)**를 선택합니다.
- 장애 조치를 사용하지 않는 경우 - vSphere Web Client에서 ASA 가상의 전원을 끕니다. 예를 들어 ASA 가상을 클릭한 다음 **Power Off the virtual machine(**가상 머신 전원 끄기**)**을 클릭하거나, ASA 가상을 마우스 오른쪽 버튼으로 클릭하고 **Shut Down Guest OS(**게스트 **OS** 종료**)**를 선택합니다.

단계 **4** ASA 가상을 클릭한 다음 **Edit Virtual machine settings(**가상 머신 설정 수정**)**를 클릭하거나, ASA 가상을 마우스 오른쪽 버튼으로 클릭하고 **Edit Settings(**설정 수정**)**를 선택합니다.

Edit Settings(설정 수정**)** 대화 상자가 나타납니다.

단계 **5** ASA 가상에 대한 라이선싱, 1 페이지의 CPU 및 메모리 요구 사항을 참조하여 새 vCPU 라이센스에 대한 올바른 값을 확인합니다.

단계 **6** **Virtual Hardware(**가상 하드웨어**)** 탭의 **CPU** 드롭다운 목록에서 새 값을 선택합니다.

단계 **7** **Memory(**메모리**)**에 RAM에 대한 새 값을 입력합니다.

단계 **8** **OK(**확인**)**를 클릭합니다.

단계 **9** ASA 가상의 전원을 켭니다. 예를 들어 **Power On the Virtual Machine(**가상 머신 전원 켜기**)**을 클릭합니다.

단계 **10** 장애 조치 쌍의 경우 다음을 수행합니다.

1. 활성 유닛에 대한 콘솔을 열거나 활성 유닛에서 ASDM을 실행합니다.
2. 대기 유닛의 시작이 완료되면 대기 유닛에 장애 조치를 수행합니다.
   - ASDM: **Monitoring(**모니터링**)** > **Properties(**속성**)** > **Failover(**장애 조치**)** > **Status(**상태**)**를 선택하고**Make Standby(**대기로 전환**)**을 클릭합니다.
   - CLI: **failover active**
3. 활성 유닛에 대해 3~9단계를 반복합니다.

다음에 수행할 작업

자세한 내용은 ASA 가상에 대한 라이선싱, 1 페이지를 참조하십시오.

VMware의 ASA 가상에 대한 성능 조정

ESXi 컨피그레이션의 성능 향상

ESXi 호스트 CPU 구성 설정을 조정하여 ESXi 환경에서 ASA 가상의 성능을 높일 수 있습니다. 선호도 예약 옵션을 사용하면 호스트의 물리적 코어(하이퍼스레딩이 활성화된 경우 하이퍼스레드도 포함) 전체에서 가상 머신 CPU가 분산되는 방식을 제어할 수 있습니다. 이 기능을 사용하면 각 가상 머신을 지정된 선호도 집합의 프로세서에 할당할 수 있습니다.

자세한 내용은 다음 VMware 문서를 참조하십시오.

- vSphere 리소스 관리의 *CPU* 리소스 관리 장.
- VMware vSphere에 대한 성능 모범 사례.
- vSphere 클라이언트 온라인 도움말.

NUMA 지침

NUMA(Non-Uniform Memory Access)는 멀티프로세서 시스템의 프로세서와 관련한 기본 메모리 모듈의 배치를 설명하는 공유 메모리 아키텍처입니다. 프로세서가 자체 노드 내에 있지 않은 메모리(원격 메모리)에 액세스하는 경우, 데이터를 로컬 메모리에 액세스할 때보다 느린 속도로 NUMA 연결을 통해 전송해야 합니다.

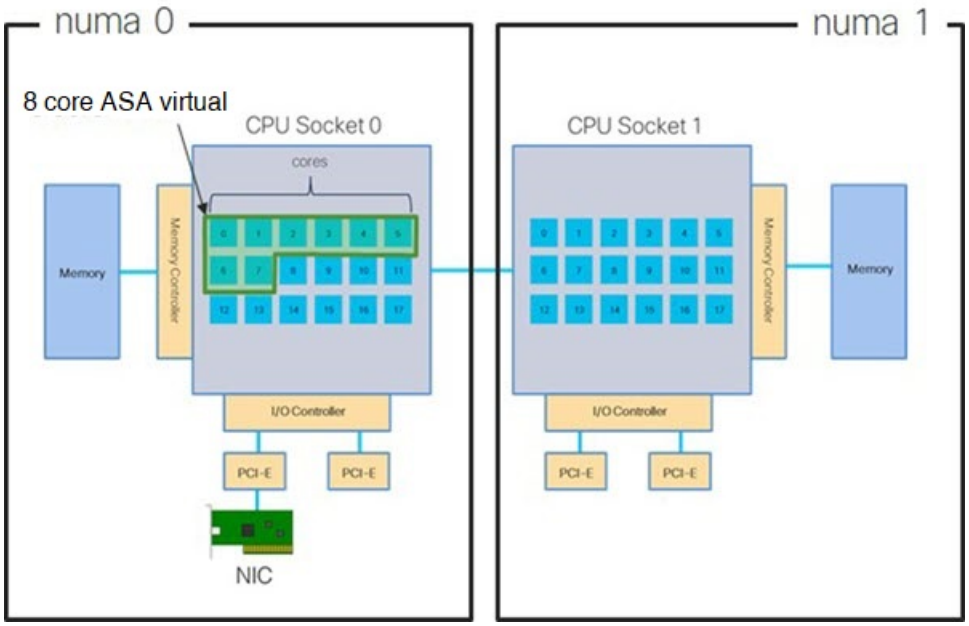
x86 서버 아키텍처는 여러 소켓 및 단일 소켓 내의 여러 코어로 구성됩니다. 각 CPU 소켓과 소켓의 메모리 및 I/O를 NUMA 노드라고 합니다. 메모리에서 패킷을 효율적으로 읽으려면 게스트 애플리케이션 및 관련 주변 장치(예: NIC)가 동일한 노드 내에 있어야 합니다.

최적의 ASA 가상 성능을 얻으려면 다음 조건을 충족해야 합니다.

- ASA 가상 머신은 단일 NUMA 노드에서 실행해야 합니다. 단일 ASA 가상이 소켓 2개에서 실행되도록 구축되면 성능이 크게 저하됩니다.
- 8코어 ASA 가상(그림 1: 8코어 NUMA 아키텍처 예시, 38 페이지)을 사용하려면 호스트 CPU의 각 소켓에 최소 8개의 코어가 있어야 합니다. 서버에서 실행 중인 다른 VM도 고려해야 합니다.
- 16코어 ASA 가상(그림 2: 16코어 ASA 가상 NUMA 아키텍처 예시, 39 페이지)을 사용하려면 호스트 CPU의 각 소켓에 최소 16개의 코어가 있어야 합니다. 서버에서 실행 중인 다른 VM도 고려해야 합니다.
- NIC는 ASA 가상 머신과 동일한 NUMA 노드에 있어야 합니다.

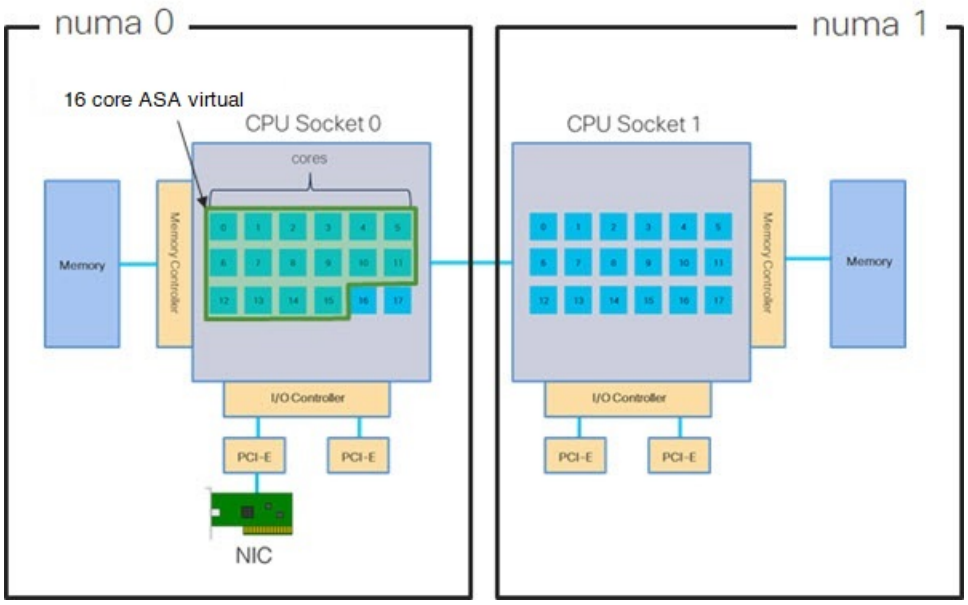
다음 그림에서는 CPU 소켓 2개가 있으며 각 CPU에 코어 18개가 있는 서버를 확인할 수 있습니다. 8코어 ASA 가상을 사용하려면 호스트 CPU의 각 소켓에 최소 8개의 코어가 있어야 합니다.

그림 ***1: 8***코어 ***NUMA*** 아키텍처 예시



다음 그림에서는 CPU 소켓 2개가 있으며 각 CPU에 코어 18개가 있는 서버를 확인할 수 있습니다. 16코어 ASA 가상을 사용하려면 호스트 CPU의 각 소켓에 최소 8개의 코어가 있어야 합니다.

그림 *2: 16*코어 *ASA* 가상 *NUMA* 아키텍처 예시



NUMA 시스템을 ESXi와 함께 사용하는 자세한 방법은 해당 VMware ESXi 버전의 VMware 문서 *vSphere* 리소스 관리에서 확인할 수 있습니다. 이 문서 및 기타 관련 문서의 최신 버전을 확인하려면 http://www.vmware.com/support/pubs를 참조하십시오.

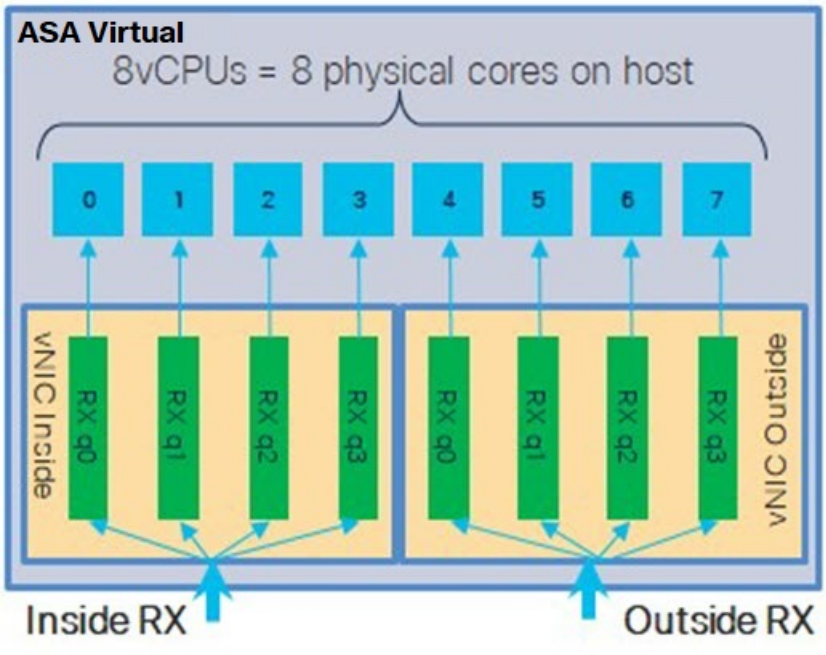
RSS(Receive Side Scaling)를 위한 다중 RX 대기열

ASA 가상은 RSS(Receive Side Scaling)를 지원합니다. RSS는 네트워크 수신 트래픽을 여러 프로세서 코어에 병렬로 분산하기 위해 네트워크 어댑터에서 활용하는 기술입니다. 최대 처리량을 확보하려면 각 vCPU(코어)에 자체 NIC RX 대기열이 있어야 합니다. 일반적인 RA VPN 구축에서는 단일 내부/외부 인터페이스 쌍을 사용합니다.

중요 여러 RX 대기열을 사용하려면 ASA 가상 버전 9.13(1) 이상이 필요합니다.

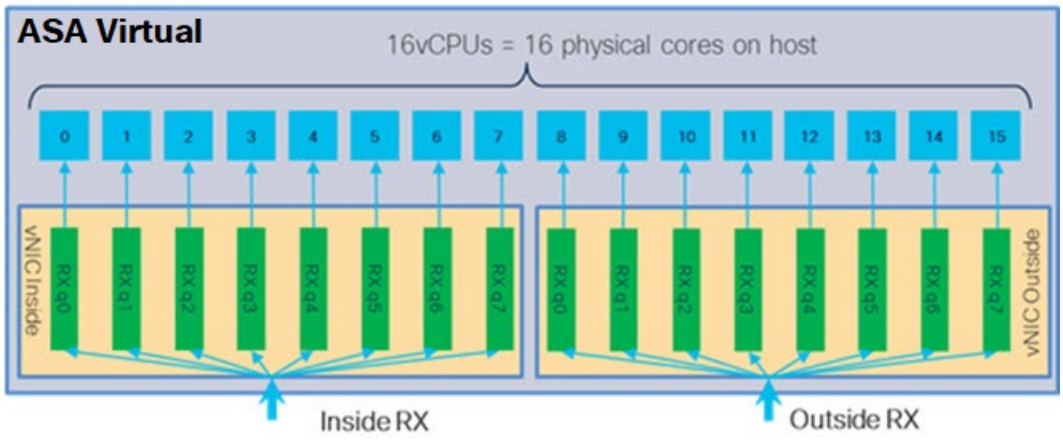
내부/외부 인터페이스 쌍이 있는 8코어 VM의 경우, 각 인터페이스에는 그림 3: 8코어 ASA 가상 RSS RX 대기열, 40 페이지에서처럼 RX 대기열 4개가 있습니다.

그림 *3: 8*코어 *ASA* 가상 *RSS RX* 대기열



내부/외부 인터페이스 쌍이 있는 16코어 VM의 경우, 각 인터페이스에는 그림 4: 16코어 ASA 가상 RSS RX 대기열, 40 페이지에서처럼 RX 대기열 8개가 있습니다.

그림 *4: 16*코어 *ASA* 가상 *RSS RX* 대기열



다음 표에는 VMware용 ASA 가상의 vNIC 및 지원되는 RX 대기열 수가 나와 있습니다. 지원되는 vNIC에 대한 설명은 권장 vNIC, 18 페이지를 참조하십시오.

표 *11: VMware* 권장 *NIC/vNIC*

NIC카드	vNIC 드라이버	드라이버 기술	RX 대기열 수	성능
x710*	i40e	PCI 패스스루	최대 8개	PCI 패스스루는 테스트한 NIC의 최고 성능을 제공합니다. 통과 모드에서 NIC는 ASA 가상 전용이며 Virtual에는 최적 선택지가 아닙니다.
	i40evf	SR-IOV	4	x710 NIC를 사용하는 SR-IOV는 PCI 패스스루보다 처리량이 낮습니다(~30%). VMware의 i40evf는 i40evf당 최대 RX 대기열 수가 4개입니다. 16코어 VM에서 최대 처리량을 달성하려면 RX 대기열 8개가 필요합니다.
x520	ixgbe-vf	SR-IOV	2	—
	ixgbe	PCI 패스스루	6	(PCI 통과 모드의 경우) ixgbe 드라이버에는 RX 대기열 6개가 있습니다. 성능은 i40evf(SR-IOV)와 비슷한 수준입니다.
해당 없음	vmxnet3	반가상화	최대 8개	ASAv100에는 권장되지 않습니다.
해당 없음	e1000	VMware에서는 권장하지 않습니다.		
*ASA 가상은 x710 NIC용 1.9.5 i40en 호스트 드라이버와 호환되지 않습니다. 이전 또는 최신 드라이버 버전을 사용해야 합니다. NIC 드라이버 및 펌웨어 버전을 식별하거나 확인하는 ESXCLI 명령에 관한 정보는 NIC 드라이버 및 펌웨어 버전 식별, 41 페이지를 참고하십시오.				

NIC 드라이버 및 펌웨어 버전 식별

특정 펌웨어 및 드라이버 버전 정보를 식별하거나 확인해야 하는 경우 ESXCLI 명령을 사용하여 데이터를 찾을 수 있습니다.

- 설치된 NIC 목록을 가져오려면 SSH를 통해 해당 호스트에 연결하고 `esxcli network nic list` 명령을 실행합니다. 이 명령은 디바이스 및 일반 정보 레코드를 제공합니다.
- 설치된 NIC 목록이 있으면 자세한 컨피그레이션 정보를 가져올 수 있습니다. `esxcli network nic get` 명령을 실행하여 필요한 NIC의 이름인 `esxcli network nic get –n <nic name>`을 지정합니다.

참고 일반 네트워크 어댑터 정보는 VMware vSphere 클라이언트에서도 확인할 수 있습니다. 어댑터와 드라이버는 **Configure(**구성**)** 탭의 **Physical Adapters(**물리적 어댑터**)**에서 확인할 수 있습니다.

SR-IOV 인터페이스 프로비저닝

SR-IOV를 사용하면 여러 VM이 호스트 내에서 단일 PCIe 네트워크 어댑터를 공유할 수 있습니다. SR-IOV는 다음 기능을 정의합니다.

- PF(물리적 기능) - PF는 SR-IOV 기능을 포함하는 전체 PCIe 기능입니다. 이러한 기능은 호스트 서버에서 일반 고정 NIC로 표시됩니다.
- VF(가상 기능) - VF는 데이터 전송을 지원하는 경량 PCIe 기능입니다. VF는 PF에서 파생되어 관리됩니다.

VF는 가상화된 운영 체제 프레임워크 내에서 ASA 가상 머신에 최대 10Gbps의 연결을 제공할 수 있습니다. 이 섹션에서는 KVM 환경에서 VF를 구성하는 방법을 설명합니다. ASA 가상에서의 SR-IOV 지원은 ASA 가상 및 SR-IOV 인터페이스 프로비저닝, 12 페이지에서 설명합니다.

지침 및 제한 사항

SR-IOV 인터페이스에 대한 지침

VMware vSphere 5.1 이상 릴리스는 특정 구성의 환경에서만 SR-IOV를 지원합니다. SR-IOV가 활성화된 경우 vSphere의 일부 기능이 작동하지 않습니다.

SR-IOV 인터페이스에 대한 지침 및 제한 사항, 12 페이지에서 설명하는 ASA 가상 및 SR-IOV에 대한 시스템 요구 사항에 더해, VMware 문서의 SR-IOV 사용을 위해 지원되는 구성도 검토하여 요구 사항, 지원되는 NIC, 기능 작동 여부, VMware 및 SR-IOV를 위한 업그레이드 요구 사항에 대한 자세한 정보를 확인해야 합니다.

이 섹션에서는 VMware 시스템에서 SR-IOV 인터페이스를 프로비저닝 하는 다양한 설정 및 구성 단계를 설명합니다. 이 섹션의 정보는 VMware ESXi 6.0 및 vSphere 웹 클라이언트, Cisco UCS C 시리즈 서버 및 Intel Ethernet 서버 어댑터 X520-DA2를 사용한 특정 랩 환경의 디바이스에서 생성되었습니다.

SR-IOV 인터페이스에 대한 제한 사항

ASA 가상가 부팅될 때 SR-IOV 인터페이스가 ESXi에서 표시되는 순서의 역순으로 표시될 수 있다는 점에 유의하십시오. 이로 인해 인터페이스 구성 오류가 발생하여 특정 ASA 가상 머신에 대한 네트워크 연결이 부족할 수 있습니다.

주의 ASA 가상에서 SR-IOV 네트워크 인터페이스 구성을 시작하기 전에 인터페이스 매핑을 확인하는 것이 중요합니다. 이렇게 하면 네트워크 인터페이스 구성이 VM 호스트의 올바른 물리적 MAC 주소 인터페이스에 적용됩니다.

ASA 가상이 부팅되면 인터페이스에 매핑되는 MAC 주소를 확인할 수 있습니다. 인터페이스에 대한 MAC 주소를 포함해 자세한 인터페이스 정보를 확인하려면 **show interface** 명령을 사용합니다. MAC 주소를 **show kernel ifconfig** 명령의 결과와 비교해 올바른 인터페이스 할당을 확인합니다.

ESXi Host BIOS 확인

VMware에서 SR-IOV 인터페이스를 사용해 ASA 가상을 구축하려면 가상화를 지원하고 활성화해야 합니다. VMware는 SR-IOV 지원을 위한 온라인 호환성 가이드 및 가상화 활성화 여부를 탐지할 수 있는 CPU 식별 유틸리티의 다운로드 등 가상화 지원을 확인하는 몇 가지 방법을 제공합니다.

ESXi 호스트에 로그인하여 BIOS에서 가상화가 활성화되었는지 여부를 확인할 수도 있습니다.

단계 **1** 다음 방법 중 하나를 사용해 ESXi 셸에 로그인합니다.

- 호스트에 직접 액세스하는 경우 Alt + F2를 눌러 시스템의 물리적 콘솔에서 로그인 페이지를 엽니다.
- 호스트에 원격으로 연결하는 경우에는 SSH나 다른 원격 콘솔 연결을 사용해 호스트에서 세션을 시작합니다.

단계 **2** 호스트에서 인식하는 사용자 이름 및 비밀번호를 입력합니다.

단계 **3** 다음 명령을 실행합니다.

예제:

```
esxcfg-info|grep "\----\HV Support"
```

HV Support 명령의 출력은 사용 가능한 하이퍼바이저 지원 유형을 나타냅니다. 다음은 가능한 값에 대한 설명입니다.

0 - VT/AMD-V는 이 하드웨어에 대한 지원이 제공되지 않음을 나타냅니다.

1 - VT/AMD-V는 VT 또는 AMD-V를 사용할 수 있지만, 이 하드웨어에서는 지원되지 않음을 나타냅니다.

2 - VT/AMD-V는 VT 또는 AMD-V를 사용할 수 있지만 현재 BIOS에서 활성화되어 있지 않음을 나타냅니다.

3 - VT/AMD-V는 BIOS에서 VT 또는 AMD-V가 활성화되어 있으며 이를 사용할 수 있음을 나타냅니다.

예제:

```
~ # esxcfg-info|grep "\----\HV Support"
        |----HV Support.........................3
```

값 3은 가상화가 지원되고 활성화되었음을 나타냅니다.

다음에 수행할 작업

- 호스트의 물리 어댑터에서 SR-IOV를 활성화합니다.

호스트 물리적 어댑터에서 SR-IOV 활성화

vSphere 웹 클라이언트를 사용해 SR-IOV를 활성화하고 호스트에서 가상 기능 수를 설정합니다. 이 작업을 하기 전에는 가상 머신을 가상 기능에 연결할 수 없습니다.

시작하기 전에

- SR-IOV 호환 NIC(네트워크 인터페이스 카드)가 설치되어 있는지 확인합니다. 지원되는 SR-IOV용 NIC, 13 페이지를 참조하십시오.

단계 1 vSphere 웹 클라이언트에서 SR-IOV를 활성화하려는 ESXi 호스트로 이동합니다.

단계 2 **Manage(**관리**)** 탭에서 **Networking(**네트워킹**)**을 클릭하고 **Physical adapters(**물리 어댑터**)**를 선택합니다.

SR-IOV 속성을 확인하여 물리 어댑터의 SR-IOV 지원 여부를 확인할 수 있습니다.

단계 3 물리 어댑터를 선택하고 **Edit adapter settings(**어댑터 설정 수정**)**를 클릭합니다.

단계 4 SR-IOV의 **Status(**상태**)** 드롭다운 메뉴에서 **Enabled(**활성화됨**)**를 선택합니다.

단계 5 **Numer of virtual functions(**가상 기능 수**)** 텍스트 상자에 어댑터에서 구성할 가상 기능 수를 입력합니다.

참고 ASAv50의 경우 인터페이스당 1개 이상의 VF를 사용하지 않는 것을 권장합니다. 물리 인터페이스를 여러 가상 기능과 공유하는 경우 성능 저하가 발생할 가능성이 높습니다.

단계 6 **OK(**확인**)**를 클릭합니다.

단계 7 ESXi 호스트를 재시작합니다.

물리 어댑터 항목으로 표시되는 NIC 포트에서 가상 기능이 활성화됩니다. 이는 호스트의 **Settings(**설정**)** 탭 내 PCI 디바이스 목록에 표시됩니다.

다음에 수행할 작업

- 표준 vSwitch를 생성하여 SR-IOV 기능 및 구성을 관리합니다.

vSphere 스위치 생성

vSphere 스위치를 생성하여 SR-IOV 인터페이스를 관리 합니다.

단계 1 vSphere 웹 클라이언트에서 ESXi 호스트로 이동합니다.

단계 2 **Manage(**관리**)**에서 **Networking(**네트워킹**)**, **Virtual switches(**가상 스위치**)**를 선택합니다.

단계 3 더하기(+) 기호가 있는 초록색 지구본 아이콘인 **Add host networking(**호스트 네트워킹 추가**)** 아이콘을 클릭합니다.

단계 4 표준 스위치용 가상 머신 포트 그룹 연결 유형을 선택하고 **Next(**다음**)**를 클릭합니다.

단계 5 새 표준 스위치를 선택하고 **Next(**다음**)**를 클릭합니다.

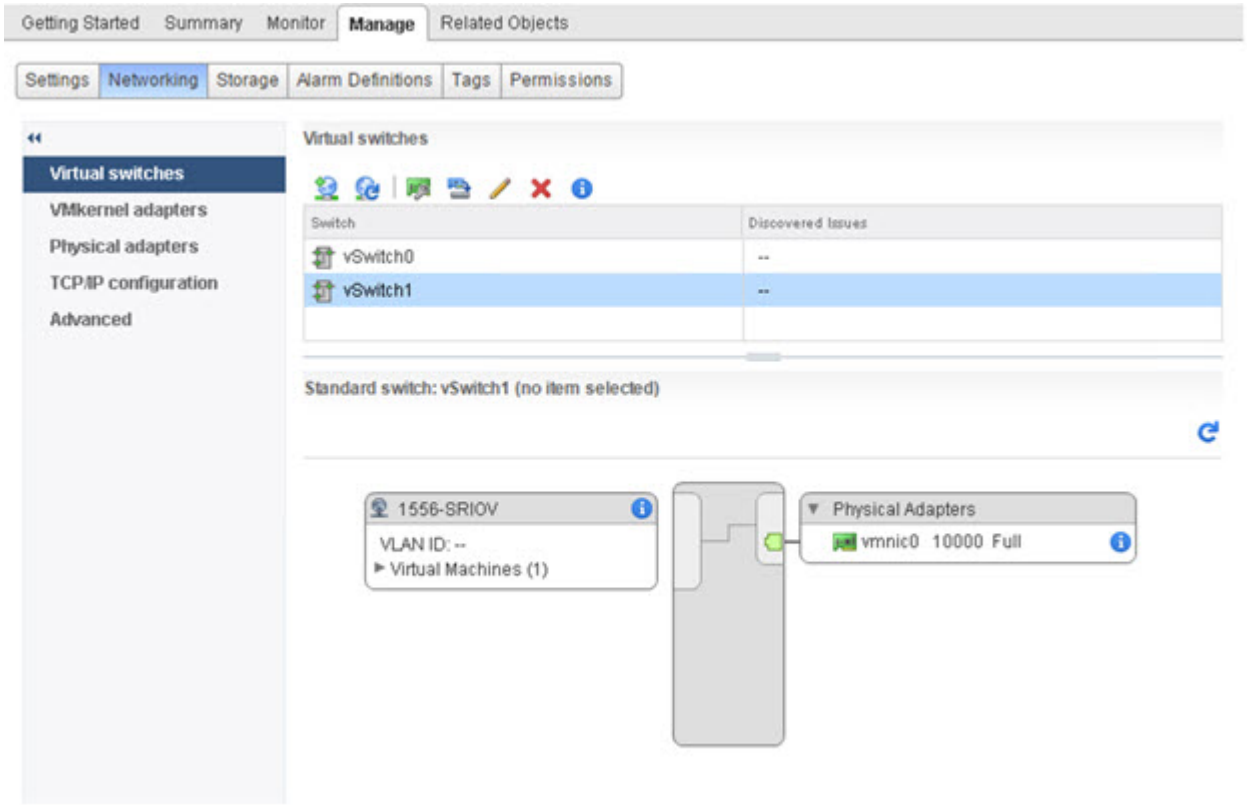
단계 6 새 표준 스위치에 물리 네트워크 어댑터를 추가합니다.

a) 할당된 어댑터에서 녹색 더하기(+) 기호를 클릭하여 어댑터를 추가합니다.
b) 목록에서 SR-IOV에 해당하는 네트워크 인터페이스를 선택합니다. 예를 들면 Intel(R) 82599 10 Gigabit 듀얼 포트 네트워크 연결입니다.
c) **Failover order Group(**페일오버 순서 그룹**)** 드롭다운 메뉴에서 활성 어댑터를 선택합니다.
d) **OK(**확인**)**를 클릭합니다.

단계 **7** SR-IOV vSwitch에 대한 네트워크 레이블을 입력하고 **Next(**다음**)**를 클릭합니다.

단계 **8** 완료 대기 페이지에서 선택한 항목을 검토한 다음 **Finish(**마침**)**를 클릭합니다.

그림 ***5: SR-IOV*** 인터페이스가 연결된 새 ***vSwitch***



다음에 수행할 작업

- 가상 머신의 호환성 수준을 검토합니다.

가상 머신 호환성 수준 업그레이드

호환성 수준은 호스트 머신에서 사용 가능한 물리 하드웨어에 해당하는 가상 머신에서 사용할 수 있는 가상 하드웨어를 결정합니다. ASA 가상 머신은 하드웨어 수준이 10 이상이어야 합니다. 이렇게 하면 ASA 가상에서 SR-IOV 통과 기능을 제공합니다. 이 절차에서는 ASA 가상을 지원되는 최신 가상 하드웨어 버전으로 즉시 업그레이드합니다.

가상 머신 하드웨어 버전 및 호환성에 대한 자세한 내용은 vSphere 가상 머신 관리 설명서를 참조하십시오.

단계 1 vSphere 웹 클라이언트에서 vCenter 서버에 로그인합니다.

단계 2 수정할 ASA 가상 머신을 찾습니다.

a) 데이터 센터, 폴더, 클러스터, 리소스 풀 또는 호스트를 선택하고 관련 개체 탭을 클릭합니다.

b) **Virtual Machines(**가상 머신**)**를 클릭하고 목록에서 ASA 가상 머신을 선택합니다.

단계 3 선택한 가상 머신의 전원을 끕니다.

단계 4 ASA 가상을 마우스 오른쪽 버튼으로 클릭하고 **Actions(**작업**)** > **All vCenter Actions(**모든 **vCenter** 작업**)** > **Compatibility(**호환성**)** > **Upgrade VM Compatibility(VM** 호환성 업그레이드**)**을 선택합니다.

단계 5 **Yes(**예**)**를 클릭하여 업그레이드를 확인합니다.

단계 6 가상 머신 호환성을 위해 **ESXi 5.5** 이상 옵션을 선택합니다.

단계 7 (선택 사항) 일반 게스트 **OS** 종료가 완료된 후 업그레이드를 선택합니다.

선택한 가상 머신이 선택한 호환성 설정에 해당하는 하드웨어 버전으로 업그레이드되고, 가상 머신의 Summary(요약) 탭에 새 하드웨어 버전이 업데이트됩니다.

다음에 수행할 작업

- ASA 가상을 SR-IOV 통과 네트워크 어댑터를 통해 가상 기능과 연결합니다.

ASA 가상에 SR-IOV NIC 할당

ASA 가상 머신 및 물리 NIC가 데이터를 교환할 수 있도록 하려면, SR-IOV 통과 네트워크 어댑터로 ASA 가상을(를) 하나 이상의 가상 기능과 연결해야 합니다. 다음 절차에서는 vSphere 웹 클라이언트를 사용해 ASA 가상 머신에 SR-IOV NIC를 할당하는 방법을 설명합니다.

단계 1 vSphere 웹 클라이언트에서 vCenter 서버에 로그인합니다.

단계 2 수정할 ASA 가상 머신을 찾습니다.

a) 데이터 센터, 폴더, 클러스터, 리소스 풀 또는 호스트를 선택하고 관련 개체 탭을 클릭합니다.

b) **Virtual Machines(**가상 머신**)**를 클릭하고 목록에서 ASA 가상 머신을 선택합니다.

단계 3 가상 머신의 **Manager(**관리**)** 탭에서 **Settings(**설정**)** > **VM Hardware(VM** 하드웨어**)**를 선택합니다.

단계 4 **Edit(**수정**)**을 클릭하고 **Virtual Hardware(**가상 하드웨어**)** 탭을 선택합니다.

단계 5 **New device(**새 디바이스**)** 드롭다운 메뉴에서 **Network(**네트워크**)**를 선택하고 **Add(**추가**)**를 클릭합니다.

New Network(새 네트워크**)** 인터페이스가 표시됩니다.

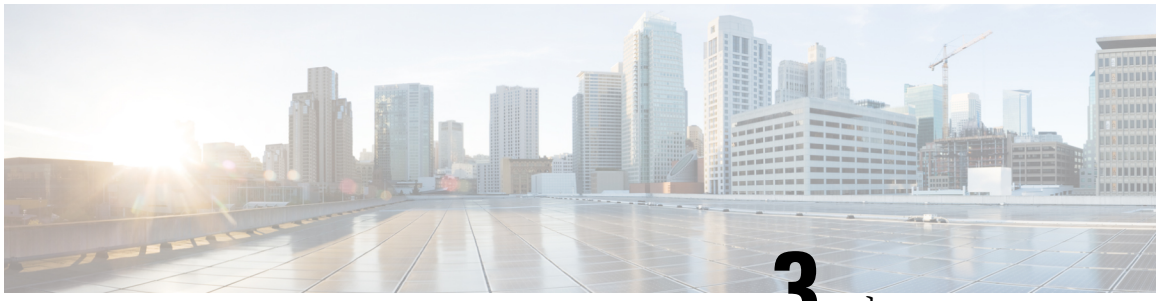
단계 6 **New Network(**새 네트워크**)** 섹션을 확장하고 사용 가능한 SRIOV 옵션을 선택합니다.

단계 7 **Adapter Type (**어댑터 유형**)** 드롭다운 메뉴에서 **SR-IOV passthrough(SR-IOV** 통과**)**를 선택 합니다.

단계 8 **Physical Function(**물리적 기능**)** 드롭다운 메뉴에서 통과 가상 머신 어댑터에 해당하는 물리 어댑터를 선택합니다.

단계 **9** 가상 머신을 켭니다.

가상 머신의 전원을 켤 때 ESXi 호스트는 물리 어댑터에서 무료 가상 기능을 선택하고, 이를 SR-IOV 통과 어댑터에 매핑합니다. 호스트는 가상 머신 어댑터의 모든 속성 및 기본 가상 기능을 검증합니다.



3 장

KVM을 사용하여 ASA 가상 구축

KVM(Kernel-based Virtual Machine)을 실행할 수 있는 모든 서버 클래스 x86 CPU 디바이스에 ASA 가상을 구축할 수 있습니다.

중요 ASA 가상의 최소 메모리 요구 사항은 2GB입니다. 현재 ASA 가상이 2GB 미만의 메모리로 실행되는 경우에는 ASA 가상 머신의 메모리를 늘리지 않고는 이전 버전에서 9.13(1) 이상으로 업그레이드할 수 없습니다. 최신 버전의 새 ASA 가상 머신을 재구축할 수도 있습니다.

KVM 지침 및 제한 사항에서의 ASA 가상

ASA 가상 구축에 사용되는 특정 하드웨어는 구축된 인스턴스 수 및 사용 요구 사항에 따라 달라질 수 있습니다. 생성하는 각 가상 어플라이언스는 호스트 머신에서 최소 리소스 할당(메모리, CPU 수 및 디스크 공간)을 필요로 합니다.

중요 ASA 가상는 8GB 디스크 스토리지 크기로 구축됩니다. 디스크 공간의 리소스 할당은 변경할 수 없습니다.

ASA 가상을 구축하기 전에 다음 지침 및 제한 사항을 검토하십시오.

KVM 시스템 요구 사항에서의 ASA 가상

최적의 성능을 보장하려면 아래 사양을 준수해야 합니다. ASA 가상에는 다음 요구 사항이 적용됩니다.

- 호스트 CPU는 가상화 확장을 사용하는 서버 클래스 x86 기반 Intel 또는 AMD CPU여야 합니다.

  예를 들어 ASA 가상 성능 테스트 랩에서는 2.6GHz에서 실행되는 Intel® Xeon® CPU E5-2690v4 프로세서를 사용하는 Cisco UCS®(Unified Computing System™) C 시리즈 M4 서버가 최소 요구 사항입니다.

권장 vNIC

최적의 성능을 위해 다음 vNIC를 사용하는 것이 좋습니다.

- PCI 패스스루의 i40e - 서버의 물리적 NIC를 VM 전용으로 지정하고 DMA(Direct Memory Access)를 통해 NIC와 VM 간에 패킷 데이터를 전송합니다. 패킷 이동에는 CPU 사이클이 필요하지 않습니다.
- i40evf/ixgbe-vf - 위와 사실상 동일하지만(NIC와 VM 간의 DMA 패킷) 여러 VM에서 NIC를 공유할 수 있습니다. 뛰어난 구축 유연성 때문에 일반적으로 SR-IOV를 선호합니다. 확인
- virtio - 10Gbps 작업을 지원하지만 CPU 사이클이 필요한 반가상화 네트워크 드라이버입니다.

참고 KVM 시스템에서 실행 중인 ASA 가상 인스턴스는 vNIC 드라이버 i40e 버전 2.11.25를 사용하는 SR-IOV 인터페이스에서 데이터 연결 문제가 발생할 수 있습니다. 이 문제를 해결하려면 vNIC 버전을 다른 버전으로 업그레이드하는 것이 좋습니다.

성능 최적화

ASA 가상에서 최상의 성능을 얻으려면 VM과 호스트를 모두 조정할 수 있습니다. 자세한 내용은 KVM의 ASA 가상에 대한 성능 조정, 58 페이지를 참조하십시오.

- **NUMA** - 게스트 VM의 CPU 리소스를 단일 NUMA(Non-Uniform Memory Access) 노드로 격리하면 ASA 가상 성능을 개선할 수 있습니다. 자세한 내용은 NUMA 지침, 59 페이지를 참조하십시오.
- **Receive Side Scaling** — ASA 가상는 RSS(Receive Side Scaling)를 지원합니다. RSS는 네트워크 수신 트래픽을 여러 프로세서 코어로 분산하기 위해 네트워크 어댑터에서 활용하는 기술입니다. 자세한 내용은 RSS(Receive Side Scaling)를 위한 다중 RX 대기열, 61 페이지를 참조하십시오.
- **VPN** 최적화 - ASA 가상을 사용한 VPN 성능 최적화를 위한 추가 고려 사항은 VPN 최적화, 63 페이지를 참조하십시오.

클러스터링

버전 9.17부터는 KVM에 구축된 ASA 가상 인스턴스에서 클러스터링이 지원됩니다. 자세한 내용은 ASAv용 ASA 클러스터를 참조하십시오.

CPU 고정

ASA 가상이 KVM 환경에서 작동하려면 CPU 고정이 필요합니다. CPU 피닝 활성화, 58 페이지를 참조하십시오.

고가용성을 위한 장애 조치 지침

장애 조치 구축의 경우, 대기 유닛에 동일한 라이선스 엔타이틀먼트가 있는지 확인합니다. 예를 들어 두 유닛 모두 2Gbps 엔타이틀먼트가 있어야 합니다.

중요 ASA 가상을 이용해 고가용성 쌍을 만들 때는 동일한 순서로 각 ASA 가상에 데이터 인터페이스를 추가해야 합니다. 각 ASA 가상에 동일한 인터페이스를 추가했지만 순서가 다른 경우 ASA 가상 콘솔에서 오류가 나타날 수 있습니다. 장애 조치 기능도 영향을 받을 수 있습니다.

Proxmox VE에서의 ASA 가상

Proxmox VE(Virtual Environment)는 KVM 가상 머신을 관리할 수 있는 오픈 소스 서버 가상화 플랫폼입니다. Proxmox VE는 웹 기반 관리 인터페이스도 제공합니다.

Proxmox VE에서 ASA 가상을 구축할 때는 에뮬레이션된 시리얼 포트를 갖도록 VM을 구성해야 합니다. 시리얼 포트가 없으면 ASA 가상은 부팅 프로세스 중에 루프에 빠집니니다. 모든 관리 작업은 Proxmox VE 웹 기반 관리 인터페이스를 사용하여 수행할 수 있습니다.

참고 Unix 쉘 또는 Windows Powershell에 익숙한 고급 사용자를 위해, Proxmox VE는 가상 환경의 모든 구성 요소를 관리할 수 있는 명령줄 인터페이스를 제공합니다. 이 명령줄 인터페이스는 지능형 탭 완성 기능과 UNIX 매뉴얼 페이지 형식의 전체 설명서를 제공합니다.

ASA 가상이 올바르게 부팅하려면 VM에 시리얼 디바이스가 구성되어 있어야 합니다.

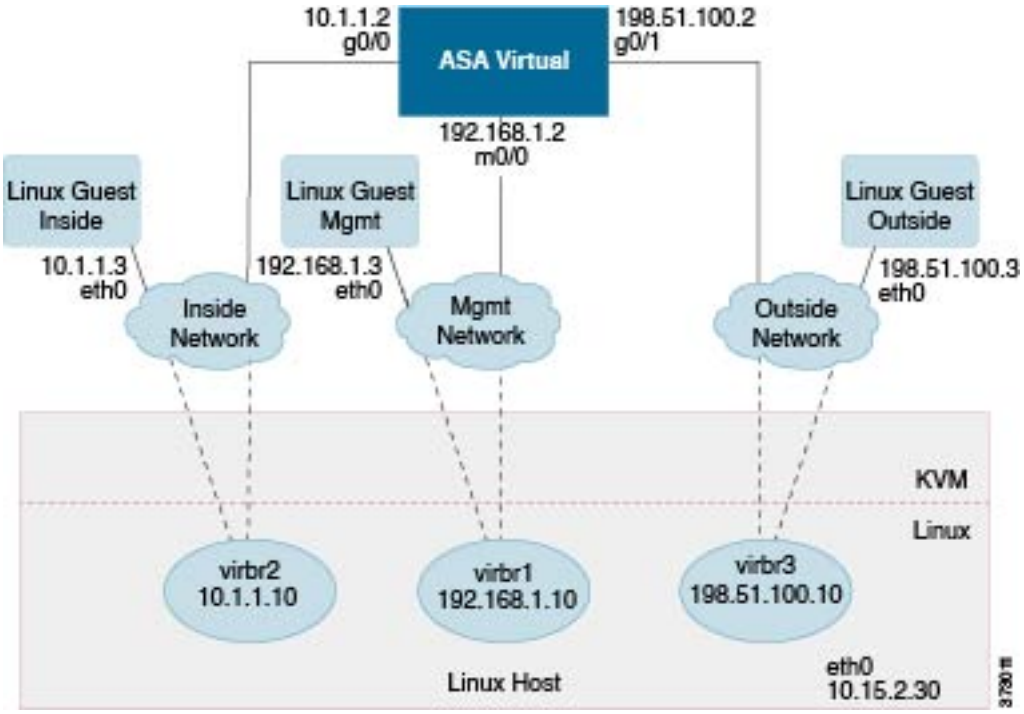
1. 기본 관리 센터의 왼쪽 탐색 트리에서 ASA 가상 머신을 선택합니다.
2. 가상 머신의 전원을 끕니다.
3. **Hardware(**하드웨어**)** > **Add(**추가**)** > **Network Device(**네트워크 디바이스**)**를 선택하고 시리얼 포트를 추가합니다.
4. 가상 머신을 켭니다.
5. Xterm.js를 사용하여 ASA 가상 머신에 액세스합니다.

게스트/서버에서 터미널을 설정하고 활성화하는 자세한 방법은 Proxmox Serial Terminal(시리얼 터미널) 페이지를 참조하십시오.

KVM을 사용한 ASA 가상 구축 정보

다음 그림에서는 ASA 가상 및 KVM을 사용하면 샘플 네트워크 토폴로지를 확인할 수 있습니다. 이 장에 설명된 절차는 샘플 토폴로지를 기반으로 합니다. ASA 가상는 내부 및 외부 네트워크 사이의 방화벽으로 작동합니다. 별도의 관리 네트워크도 구성됩니다.

그림 *6: KVM*을 사용하여 샘플 *ASA* 가상 구축



ASA 가상 및 KVM에 대한 사전 요건

- Cisco.com에서 ASA 가상 qcow2 파일을 다운로드하고 이를 Linux 호스트에 넣습니다.

  http://www.cisco.com/go/asa-software

참고 Cisco.com 로그인 및 Cisco 서비스 계약이 필요합니다.

- 이 문서에 나와 있는 샘플 구축의 경우, 사용자가 Ubuntu 18.04 LTS를 사용한다고 가정합니다. Ubuntu 18.04 LTS 호스트의 상위에 다음 패키지를 설치합니다.
  - qemu-kvm
  - libvirt-bin

  - bridge-utils
  - virt-manager
  - virtinst
  - virsh tools
  - genisoimage

- 성능은 호스트 및 해당 컨피그레이션에 영향을 받습니다. 호스트를 조정하여 KVM에서 ASA 가상의 처리량을 극대화할 수 있습니다. 일반적인 호스트 조정 개념은 NFV가 Intel을 이용하여 전달하는 패킷 처리 성능을 참조하십시오.
- Ubuntu 18.04에 유용한 최적화는 다음과 같습니다.
  - macvtap - 고성능 Linux 브리지로, Linux 브리지 대신 macvtap을 사용할 수 있습니다. Linux 브리지 대신 macvtap을 사용하려면 특정 설정을 구성해야 합니다.
  - Transparent Huge Pages - 메모리 페이지 크기를 늘리며 18.04에서 기본적으로 설정됩니다.

    Hyperthread 비활성화 - 두 개의 vCPU를 단일 코어로 줄입니다.
  - txqueuelength - 기본 txqueuelength를 4000 패킷으로 늘이고 삭제율을 줄입니다.
  - 고정 - qemu 및 vhost 프로세스를 특정 CPU 코어에 고정합니다. 특정 조건에서 고정 기능을 사용하면 성능이 대폭 향상됩니다.
- RHEL 기반 배포 최적화에 대한 자세한 내용은 Red Hat Enterprise Linux 7 가상화 조정 및 최적화 가이드를 참조하십시오.
- ASA 소프트웨어 및 ASA 가상 하이퍼바이저 호환성에 대해서는 Cisco Secure Firewall ASA 호환성을 참조하십시오.

Day 0 컨피그레이션 파일 준비

ASA 가상를 실행하기 전에 Day 0 구성 파일을 준비할 수 있습니다. 이 파일은 ASA 가상을 시작할 때 적용한 ASA 가상 컨피그레이션이 포함된 텍스트 파일입니다. 이 초기 컨피그레이션은 사용자가 선택하는 작업 디렉토리의 “day0-config”라는 이름의 텍스트 파일에 위치하며, 이 파일은 최초 부팅 시 마운트되고 읽히는 day0.iso 파일로 조작됩니다. Day 0 컨피그레이션 파일에는 최소한 관리 인터페이스를 활성화하고 공용 키 인증용 SSH 서버를 설정하는 명령이 포함되어야 할 뿐만 아니라, 완전한 ASA 컨피그레이션도 포함되어야 합니다.

최초 부팅 동안 day0.iso 파일(사용자 정의 day0.iso 또는 기본 day0.iso)을 사용할 수 있어야 합니다.

- 초기 구축 동안 ASA 가상 라이센스를 자동으로 적용하려면, Cisco Smart Software Manager에서 다운로드한 Smart Licensing ID(Identity) Token을 Day 0 컨피그레이션 파일과 같은 디렉토리에 있는 ‘idtoken’이라는 이름의 텍스트 파일로 가져옵니다.

- 가상 VGA 콘솔 대신 하이퍼바이저의 시리얼 포트에서 ASA 가상에 액세스하고 구성하려면, Day 0 컨피그레이션 파일에 콘솔 시리얼 설정을 포함하여 첫 부팅 시 시리얼 포트를 사용해야 합니다.
- 투명 모드에서 ASA 가상을 구축하려는 경우, 투명 모드에서 실행 중인 알려진 ASA 컨피그레이션 파일을 Day 0 컨피그레이션 파일로 사용해야 합니다. 이 사항은 라우팅 방화벽용 Day 0 컨피그레이션 파일에는 적용되지 않습니다.

참고 이 예에서는 Linux를 사용하지만 Windows의 경우에도 유사한 유틸리티가 있습니다.

단계 **1** "day0 config"라는 텍스트 파일에 ASA 가상에 대한 CLI 컨피그레이션을 입력합니다. 3개의 인터페이스에 대한 인터페이스 컨피그레이션 및 원하는 기타 모든 컨피그레이션을 추가합니다.

첫 줄은 ASAv 버전으로 시작해야 합니다. day0-config는 유효한 ASA 컨피그레이션이어야 합니다. day0-config를 생성하는 가장 좋은 방법은 기존 ASA 또는 ASA 가상에서 실행 중인 컨피그레이션 중 관련 부분을 복사하는 것입니다. day0-config에서의 줄의 순서가 중요하며 기존 **show running-config** 명령 출력의 순서와 일치해야 합니다.

예제:

```
ASA Version 9.4.1
!
console serial
interface management0/0
nameif management
security-level 100
ip address 192.168.1.2 255.255.255.0
no shutdown
interface gigabitethernet0/0
nameif inside
security-level 100
ip address 10.1.1.2 255.255.255.0
no shutdown
interface gigabitethernet0/1
nameif outside
security-level 0
ip address 198.51.100.2 255.255.255.0
no shutdown
http server enable
http 192.168.1.0 255.255.255.0 management
crypto key generate rsa modulus 1024
username AdminUser password paSSw0rd
ssh 192.168.1.0 255.255.255.0 management
aaa authentication ssh console LOCAL
```

단계 **2** (선택 사항) 초기 ASA 가상 구축 동안 라이센싱이 자동으로 이루어진 경우, day0-config 파일에 다음 정보가 포함되어 있는지 확인합니다.

- 관리 인터페이스 IP 주소
- (선택 사항) Smart Licensing에 사용할 HTTP 프록시
- HTTP 프록시(지정된 경우) 또는 tools.cisco.com에 대한 연결을 지원하는 **route** 명령

- tools.cisco.com을 IP 주소에 확인하는 DNS 서버
- 사용자가 요청하는 ASA 가상 라이센스를 지정하는 Smart Licensing 컨피그레이션
- (선택 사항) ASA 가상이 CSSM에서 검색을 더욱 쉽게 수행할 수 있도록 하는 고유한 호스트 이름

단계 **3** (선택 사항) Cisco Smart Software Manager에서 발급한 Smart License ID 토큰 파일을 컴퓨터에 다운로드하고, 다운로드 파일에서 ID 토큰을 복사한 다음 ID 토큰만 포함하는 'idtoken'이라는 텍스트 파일에 붙여넣습니다.

단계 **4** 텍스트 파일을 ISO 파일로 전환하여 가상 CD-ROM을 생성합니다.

예제:

```
stack@user-ubuntu:-/KvmAsa$ sudo genisoimage -r -o day0.iso day0-config idtoken
I: input-charset not specified, using utf-8 (detected in locale settings)
Total translation table size: 0
Total rockridge attributes bytes: 252
Total directory bytes: 0
Path table size (bytes): 10
Max brk space used 0
176 extents written (0 MB)
stack@user-ubuntu:-/KvmAsa$
```

ID 토큰은 ASA 가상을 Smart Licensing 서버에 자동으로 등록합니다.

단계 **5** 1단계~5단계를 반복하여 구축하려는 각 ASA 가상에 대해 적절한 IP 주소가 포함된 별도의 기본 컨피그레이션 파일을 만듭니다.

가상 브리지 XML 파일 준비

ASA 가상 게스트를 KVM 호스트에 연결하고 게스트를 서로 연결하는 가상 네트워크를 설정해야 합니다.

참고 이 절차는 KVM 호스트 밖의 외부 환경에 대한 연결을 설정하지 않습니다.

KVM 호스트에서 가상 브리지 XML 파일을 준비합니다. Day 0 컨피그레이션 파일 준비, 53 페이지에 설명된 샘플 가상 네트워크 토폴로지 경우, virbr1.xml, virbr2.xml, virbr3.xml이라는 3개의 가상 브리지 파일이 필요합니다(이러한 3개의 파일 이름을 사용해야 함. 예를 들어, virbr0은 이미 존재하므로 사용할 수 없음). 각 파일에는 가상 브리지를 설정하는 데 필요한 정보가 포함되어 있습니다. 가상 브리지에 이름과 고유한 MAC 주소를 제공해야 합니다. IP 주소를 제공하는 것은 선택 사항입니다.

단계 **1** 가상 네트워크 브리지 XML 파일 3개를 만듭니다. 예: virbr1.xml, virbr2.xml, virbr3.xml:

예제:

```
<network>
<name>virbr1</name>
<bridge name='virbr1' stp='on' delay='0' />
```

```
<mac address='52:54:00:05:6e:00' />
<ip address='192.168.1.10' netmask='255.255.255.0' />
</network>
```

예제:

```
<network>
<name>virbr2</name>
<bridge name='virbr2' stp='on' delay='0' />
<mac address='52:54:00:05:6e:01' />
<ip address='10.1.1.10' netmask='255.255.255.0' />
</network>
```

예제:

```
<network>
<name>virbr3</name>
<bridge name='virbr3' stp='on' delay='0' />
<mac address='52:54:00:05:6e:02' />
<ip address='198.51.100.10' netmask='255.255.255.0' />
</network>
```

단계 **2** 다음 정보가 포함된 스크립트를 만듭니다(이 예에서는 스크립트 이름을 virt_network_setup.sh로 지정함).

```
virsh net-create virbr1.xml
virsh net-create virbr2.xml
virsh net-create virbr3.xml
```

단계 **3** 이 스크립트를 실행하여 가상 네트워크를 설정합니다. 이 스크립트는 가상 네트워크를 불러옵니다. 네트워크는 KVM 호스트가 실행되는 동안 계속 가동됩니다.

```
stack@user-ubuntu:-/KvmAsa$ virt_network_setup.sh
```

참고 Linux 호스트를 다시 로드할 경우, virt_network_setup.sh 스크립트를 다시 실행해야 합니다. 재부팅되면 스크립트가 지속되지 않습니다.

단계 **4** 가상 네트워크가 만들어졌는지 확인합니다.

```
stack@user-ubuntu:-/KvmAsa$ brctl show
bridge name bridge id STP enabled Interfaces
virbr0 8000.0000000000000 yes
virbr1 8000.5254000056eed yes virb1-nic
virbr2 8000.5254000056eee yes virb2-nic
virbr3 8000.5254000056eec yes virb3-nic
stack@user-ubuntu:-/KvmAsa$
```

단계 **5** virbr1 브리지에 할당된 IP 주소가 표시됩니다. 이는 XML 파일에 할당한 IP 주소입니다.

```
stack@user-ubuntu:-/KvmAsa$ ip address show virbr1
S: virbr1: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN
link/ether 52:54:00:05:6e:00 brd ff:ff:ff:ff:ff:ff
inet 192.168.1.10/24 brd 192.168.1.255 scope global virbr1
valid_lft forever preferred_lft forever
```

실행 ASA 가상

virt-install 기반 구축 스크립트를 사용하여 ASA 가상을 시작합니다.

단계 **1** "virt_install_asav.sh"라는 virt-install 스크립트를 만듭니다.

ASA 가상 머신의 이름은 이 KVM 호스트의 모든 기타 VM에서 고유해야 합니다.

ASA 가상은 최대 10개의 네트워크를 지원합니다. 이 예에서는 3개의 네트워크를 사용합니다. 네트워크 브리지 절의 순서가 중요합니다. 첫 번째 줄의 항목은 항상 ASA 가상의 관리 인터페이스(Management 0/0)이고, 두 번째 줄의 항목은 ASA 가상의 GigabitEthernet 0/0이며, 세 번째 줄의 항목은 ASA 가상의 GigabitEthernet 0/1입니다. 이런 식으로 GigabitEthernet 0/8까지 이어집니다. 가상 NIC는 Virtio여야 합니다.

예제:

```
virt-install \
--connect=qemu:///system \
--network network=default,model=virtio \
--network network=default,model=virtio \
--network network=default,model=virtio \
--name=asav \
--cpu host \
--arch=x86_64 \
--machine=pc-1.0 \
--vcpus=1 \
--ram=2048 \
--os-type=linux \
--virt-type=kvm \
--import \
--disk path=/home/kvmperf/Images/desmo.qcow2,format=qcow2,device=disk,bus=virtio,cache=none \
--disk path=/home/kvmperf/asav_day0.iso,format=iso,device=cdrom \
--console pty,target_type=virtio \
--serial tcp,host=127.0.0.1:4554,mode=bind,protocol=telnet
```

단계 **2** virt_install 스크립트를 실행합니다.

예제:

```
stack@user-ubuntu:-/KvmAsa$ ./virt_install_asav.sh

Starting install...
Creating domain...
```

VM의 콘솔을 표시하는 창이 나타납니다. VM이 부팅 중인 것으로 표시됩니다. VM이 부팅될 때까지 몇 분 정도 소요됩니다. VM이 부팅을 멈추면 콘솔 화면에서 CLI 명령을 발급할 수 있습니다.

KVM의 ASA 가상에 대한 성능 조정

KVM 컨피그레이션의 성능 향상

KVM 호스트의 설정을 변경하여 KVM 환경에서 ASA 가상의 성능을 높일 수 있습니다. 이러한 설정은 호스트 서버의 컨피그레이션 설정과 무관합니다. 이 옵션은 Red Hat Enterprise Linux 7.0 KVM에서 사용할 수 있습니다.

CPU 고정을 활성화하여 KVM 컨피그레이션에서 성능을 개선할 수 있습니다.

CPU 피닝 활성화

ASA 가상을 사용하려면 KVM CPU 선호도 옵션을 사용하여 KVM 환경에서 ASA 가상 성능을 높여야 합니다. 프로세서 선호도 또는 CPU 고정을 사용하면 프로세스 또는 스레드를 CPU(중앙 처리 장치) 또는 CPU 범위에 바인딩하거나 바인딩 해제하여, 프로세스나 스레드가 아무 CPU가 아닌 지정된 CPU에서만 실행되게 할 수 있습니다.

고정되지 않은 인스턴스가 고정된 인스턴스의 리소스 요구 사항을 사용하지 못하다록, CPU 고정을 사용하는 인스턴스를 CPU 고정을 사용하지 않는 인스턴스와 다른 호스트에 구축하도록 호스트 집계를 구성하십시오.

주의 NUMA 토폴로지가 없는 인스턴스와 동일한 호스트에 NUMA 토폴로지가 있는 인스턴스를 구축하지 마십시오.

이 옵션을 사용하려면 KVM 호스트에서 CPU 고정을 구성합니다.

단계 **1** KVM 호스트 환경에서 호스트 토폴로지를 확인하여 고정에 사용할 수 있는 vCPU 수를 확인합니다.

예제:

```
virsh nodeinfo
```

단계 **2** 사용 가능한 vCPU 수를 확인합니다.

예제:

```
virsh capabilities
```

단계 **3** 프로세서 코어 집합에 vCPU를 고정합니다.

예제:

```
virsh vcpupin <vm-name> <vcpu-number> <host-core-number>
```

virsh vcpupin 명령은 ASA 가상의 각 vCPU를 대상으로 실행해야 합니다. 다음 예는 ASA 가상 컨피그레이션에 vCPU 4개가 있고 호스트에 8개의 코어가 있는 경우 필요한 KVM 명령을 보여줍니다.

```
virsh vcpupin asav 0 2
virsh vcpupin asav 1 3
```

```
virsh vcpupin asav 2 4
virsh vcpupin asav 3 5
```

호스트 코어 숫자는 0~7일 수 있습니다. 자세한 내용은 KVM 설명서를 참조하십시오.

참고 CPU 고정을 구성할 때는 호스트 서버의 CPU 토폴로지를 신중하게 고려하십시오. 여러 코어로 구성된 서버를 사용하는 경우, 여러 소켓에서 CPU 고정을 구성하지 마십시오.

KVM 컨피그레이션에서의 성능 개선은 전용 시스템 리소스가 필요하다는 단점이 있습니다.

NUMA 지침

NUMA(Non-Uniform Memory Access)는 멀티프로세서 시스템의 프로세서와 관련한 기본 메모리 모듈의 배치를 설명하는 공유 메모리 아키텍처입니다. 프로세서가 자체 노드 내에 있지 않은 메모리(원격 메모리)에 액세스하는 경우, 데이터를 로컬 메모리에 액세스할 때보다 느린 속도로 NUMA 연결을 통해 전송해야 합니다.

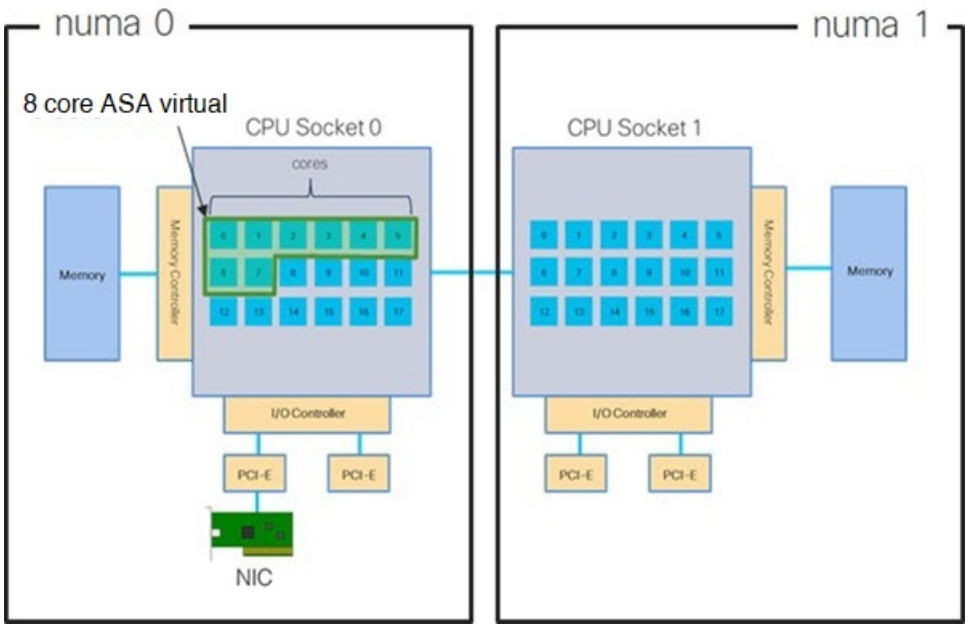
x86 서버 아키텍처는 여러 소켓 및 단일 소켓 내의 여러 코어로 구성됩니다. 각 CPU 소켓과 소켓의 메모리 및 I/O를 NUMA 노드라고 합니다. 메모리에서 패킷을 효율적으로 읽으려면 게스트 애플리케이션 및 관련 주변 장치(예: NIC)가 동일한 노드 내에 있어야 합니다.

최적의 ASA 가상 성능을 얻으려면 다음 조건을 충족해야 합니다.

- ASA 가상 머신은 단일 NUMA 노드에서 실행해야 합니다. 단일 ASA 가상이 소켓 2개에서 실행되도록 구축되면 성능이 크게 저하됩니다.
- 8코어 ASA 가상(그림 7: 8코어 ASA 가상 NUMA 아키텍처 예시, 60 페이지)을 사용하려면 호스트 CPU의 각 소켓에 최소 8개의 코어가 있어야 합니다. 서버에서 실행 중인 다른 VM도 고려해야 합니다.
- 16코어 ASA 가상(그림 8: 16코어 ASA 가상 NUMA 아키텍처 예시, 60 페이지)을 사용하려면 호스트 CPU의 각 소켓에 최소 16개의 코어가 있어야 합니다. 서버에서 실행 중인 다른 VM도 고려해야 합니다.
- NIC는 ASA 가상 머신과 동일한 NUMA 노드에 있어야 합니다.

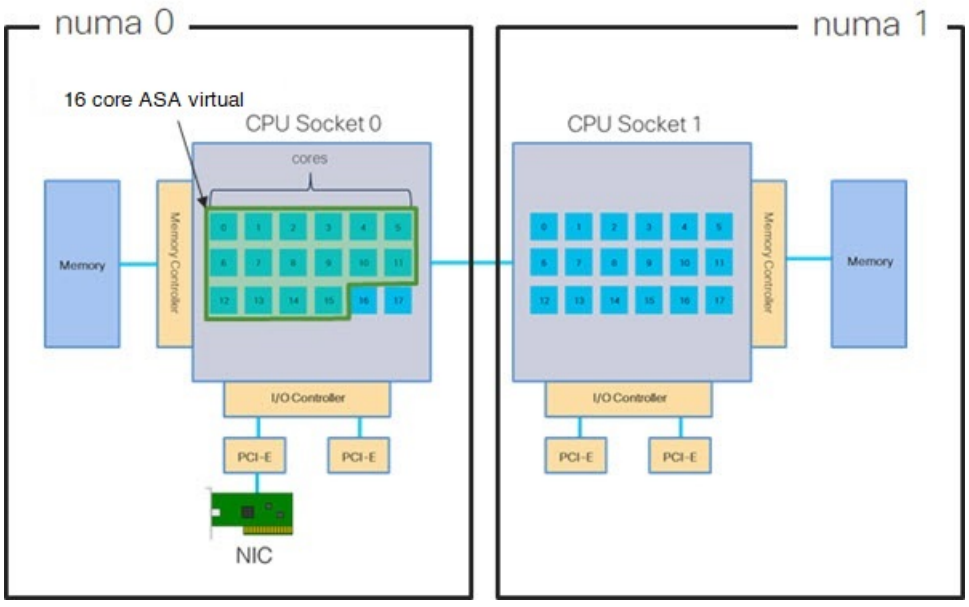
다음 그림에서는 CPU 소켓 2개가 있으며 각 CPU에 코어 18개가 있는 서버를 확인할 수 있습니다. 8코어 ASA 가상을 사용하려면 호스트 CPU의 각 소켓에 최소 8개의 코어가 있어야 합니다.

그림 *7: 8*코어 *ASA* 가상 *NUMA* 아키텍처 예시



다음 그림에서는 CPU 소켓 2개가 있으며 각 CPU에 코어 18개가 있는 서버를 확인할 수 있습니다. 16코어 ASA 가상을 사용하려면 호스트 CPU의 각 소켓에 최소 8개의 코어가 있어야 합니다.

그림 *8: 16*코어 *ASA* 가상 *NUMA* 아키텍처 예시



NUMA 최적화

NIC가 실행 중인 것과 동일한 numa 노드에서 ASA 가상 머신을 실행하는 것이 가장 좋습니다. 이렇게 하려면 다음을 수행합니다.

1. 노드의 다이어그램을 표시하려면 "lstopo"를 사용하여 NIC가 있는 노드를 결정합니다. NIC를 찾아 연결된 노드를 기록해 둡니다.
2. KVM 호스트에서 `virsh 목록`을 사용하여 ASA 가상을 찾습니다.
3. `virsh edit <VM Number>`를 이용해 VM을 편집합니다.
4. 선택한 노드에서 ASA 가상을 정렬합니다. 다음 예에서는 18 코어 노드를 가정합니다.

   노드 0에 정렬:

   ```
   <vcpu placement='static' cpuset='0-17'>16</vcpu>
   <numatune>
       <memory mode='strict' nodeset='0'/>
   </numatune>
   ```

   노드 1에 정렬:

   ```
   <vcpu placement='static' cpuset='18-35'>16</vcpu>
   <numatune>
       <memory mode='strict' nodeset='1'/>
   </numatune>
   ```

5. .xml 변경 사항을 저장하고 ASA 가상 머신 전원을 껐다가 켭니다.
6. 원하는 노드에서 VM이 실행 중인지 확인하려면 `ps aux | grep<name of your ASAv VM>`을 실행하여 프로세스 ID를 가져옵니다.
7. `sudo numastat -c <ASAv VM Process ID>`를 실행하여 ASA 가상 머신이 올바르게 정렬되었는지 확인합니다.

KVM를 이용한 NUMA 조정에 관한 자세한 내용은 RedHat 문서 9.3. libvirt NUMA 조정에서 확인할 수 있습니다.

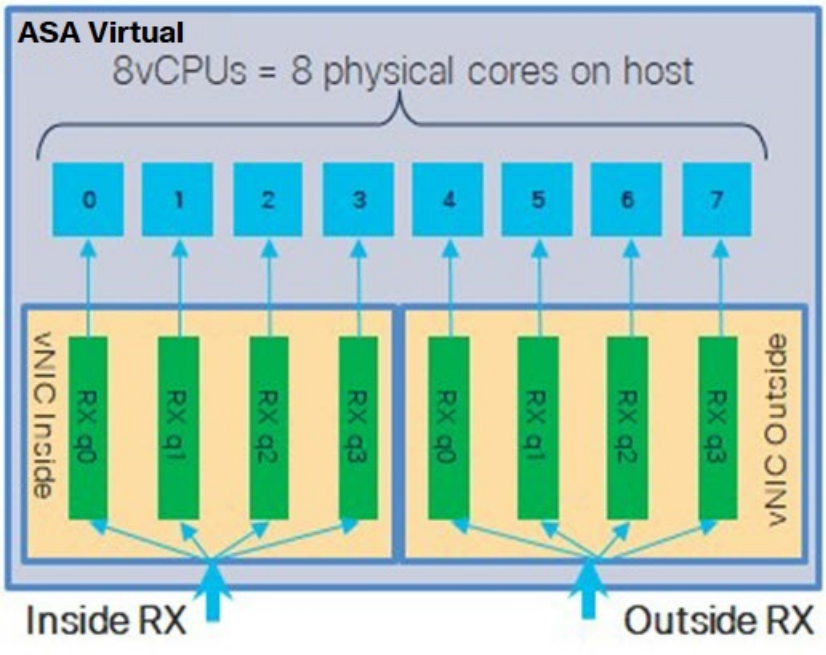
RSS(Receive Side Scaling)를 위한 다중 RX 대기열

ASA 가상은 RSS(Receive Side Scaling)를 지원합니다. RSS는 네트워크 수신 트래픽을 여러 프로세서 코어에 병렬로 분산하기 위해 네트워크 어댑터에서 활용하는 기술입니다. 최대 처리량을 확보하려면 각 vCPU(코어)에 자체 NIC RX 대기열이 있어야 합니다. 일반적인 RA VPN 구축에서는 단일 내부/외부 인터페이스 쌍을 사용합니다.

중요 여러 RX 대기열을 사용하려면 ASA 가상 버전 9.13(1) 이상이 필요합니다. KVM의 경우 *libvirt* 버전이 1.0.6 이상이어야 합니다.

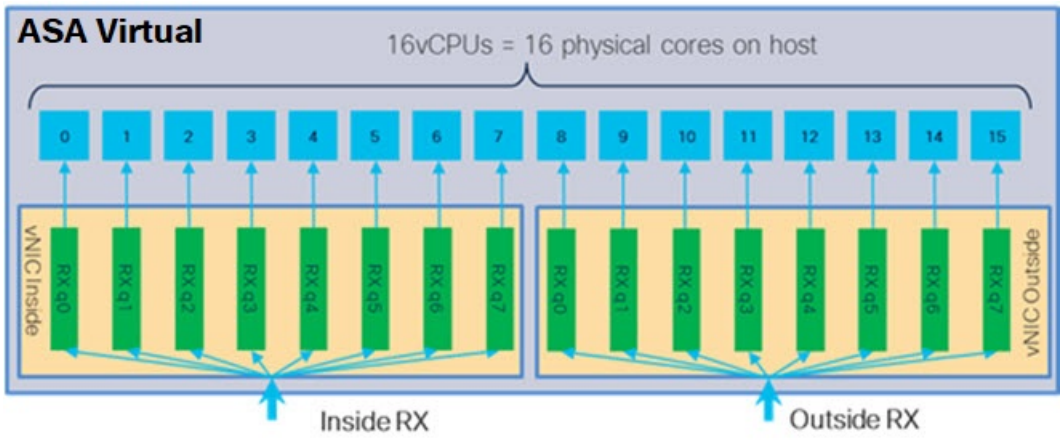
내부/외부 인터페이스 쌍이 있는 8코어 VM의 경우, 각 인터페이스에는 그림 9: 8코어 ASA 가상 RSS RX 대기열, 62 페이지에서처럼 RX 대기열 4개가 있습니다.

그림 *9: 8*코어 *ASA* 가상 *RSS RX* 대기열



내부/외부 인터페이스 쌍이 있는 16코어 VM의 경우, 각 인터페이스에는 그림 10: 16코어 ASA 가상 RSS RX 대기열, 62 페이지에서처럼 RX 대기열 8개가 있습니다.

그림 *10: 16*코어 *ASA* 가상 *RSS RX* 대기열



다음 표에는 KVM에 대한 의 vNIC 및 지원되는 RX 대기열 수가 나와 있습니다.ASA 가상 지원되는 vNIC에 대한 설명은 권장 vNIC, 50 페이지를 참조하십시오.

표 *12: KVM* 권장 *NIC/vNIC*

NIC카드	vNIC 드라이버	드라이버 기술	RX 대기열 수	성능
x710	i40e	PCI 패스스루	최대 8	x710의 PCI 통과 및 SR-IOV 모드는 최상의 성능을 제공합니다. NIC는 여러 VM에서 공유할 수 있으므로 가상 구축에는 일반적으로 SR-IOV가 선호됩니다.
	i40evf	SR-IOV	8	
x520	ixgbe	PCI 패스스루	6	x520 NIC는 x710보다 성능이 10~30% 낮습니다. x520의 PCI 통과 및 SR-IOV 모드는 유사한 성능을 제공합니다. NIC는 여러 VM에서 공유할 수 있으므로 가상 구축에는 일반적으로 SR-IOV가 선호됩니다.
	ixgbe-vf	SR-IOV	2	
해당 없음	virtio	반가상화	최대 8	ASAv100에는 권장되지 않습니다. 다른 구축에 대해서는 KVM에서 Virtio에 대한 다중 대기열 지원 활성화, 63 페이지를 참조하십시오.

KVM에서 **Virtio**에 대한 다중 대기열 지원 활성화

다음 예에서는 virsh를 사용하여 libvirt xml을 편집하여 Virtio NIC RX 대기열 수를 4로 구성하는 방법을 보여 줍니다.

```
<interface type='bridge'>
 <mac address='52:54:00:43:6e:3f'/>
 <source bridge='clients'/>
 <model type='virtio'/>
 <driver name='vhost' queues='4'/>
 <address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x0'/>
</interface>
```

중요 여러 RX 대기열을 지원하려면 *libvirt* 버전이 1.0.6 이상이어야 합니다.

VPN 최적화

다음은 ASA 가상을 사용하여 VPN 성능을 최적화하기 위한 몇 가지 추가 고려 사항입니다.

- IPSec은 DTLS보다 처리량이 높습니다.
- 암호 - GCM의 처리량은 CBC의 약 2배입니다.

SR-IOV 인터페이스 프로비저닝

SR-IOV를 사용하면 여러 VM이 호스트 내에서 단일 PCIe 네트워크 어댑터를 공유할 수 있습니다. SR-IOV는 다음 기능을 정의합니다.

- PF(물리적 기능) - PF는 SR-IOV 기능을 포함하는 전체 PCIe 기능입니다. 이러한 기능은 호스트 서버에서 일반 고정 NIC로 표시됩니다.
- VF(가상 기능) - VF는 데이터 전송을 지원하는 경량 PCIe 기능입니다. VF는 PF에서 파생되어 관리됩니다.

VF는 가상화된 운영 체제 프레임워크 내에서 ASA 가상 머신에 최대 10Gbps의 연결을 제공할 수 있습니다. 이 섹션에서는 KVM 환경에서 VF를 구성하는 방법을 설명합니다. ASA 가상에서의 SR-IOV 지원은 ASA 가상 및 SR-IOV 인터페이스 프로비저닝, 12 페이지에서 설명합니다.

SR-IOV 인터페이스 프로비저닝 요구 사항

SR-IOV를 지원하는 물리적 NIC가 있다면 SR-IOV가 활성화된 VF 또는 vNIC(Virtual NIC)를 ASA 가상 인스턴스에 연결할 수 있습니다. 또한 SR-IOV는 BIOS 및 하드웨어에서 실행 중인 운영 체제 인스턴스/하이퍼바이저에서의 지원을 필요로 합니다. 다음은 KVM 환경에서 실행되는 ASA 가상의 SR-IOV 인터페이스 프로비저닝에 대한 일반 지침 목록입니다.

- 호스트 서버에 SR-IOV을 지원하는 물리적 NIC가 있어야 합니다. SR-IOV 인터페이스에 대한 지침 및 제한 사항, 12 페이지를 참조하십시오.
- 호스트 서버의 BIOS에서 가상화를 활성화해야 합니다. 자세한 내용은 벤더 설명서를 참조하십시오.
- 호스트 서버의 BIOS에서 SR-IOV에 대한 IOMMU 전역 지원을 활성화해야 합니다. 자세한 내용은 하드웨어 벤더 설명서를 참조하십시오.

KVM 호스트 BIOS 및 호스트 OS 수정

이 섹션에서는 KVM 머신에서 SR-IOV 인터페이스를 프로비저닝하는 다양한 설정 및 구성 단계를 설명합니다. 이 섹션의 정보는 Intel Ethernet 서버 어댑터 X520-DA2를 사용하는 Cisco UCS C 시리즈 서버에서 Ubuntu 14.04를 사용하는 특정 랩 환경의 디바이스에서 생성되었습니다.

시작하기 전에

- SR-IOV 호환 NIC(네트워크 인터페이스 카드)가 설치되어 있는지 확인합니다.
- Intel VT-x(Virtualization Technology) 및 VT-d 기능이 활성화되어 있는지 확인합니다.

참고 일부 시스템 제조업체는 기본적으로 이러한 확장을 비활성화합니다. 시스템마다 BIOS 설정에 액세스하고 변경하는 방법이 다르므로, 벤더 설명서를 참조하여 프로세스를 확인하는 것이 좋습니다.

- 운영 체제를 설치하는 동안 모든 Linux KVM 모듈, 라이브러리, 사용자 툴 및 유틸리티가 설치되었는지 확인합니다. ASA 가상 및 KVM에 대한 사전 요건, 52 페이지를 참조하십시오.
- 물리적 인터페이스가 UP(작동) 상태인지 확인합니다. ifconfig <*ethname*>을 사용하여 확인합니다.

단계 **1** "root" 사용자 계정 및 비밀번호를 사용하여 시스템에 로그인합니다.

단계 **2** Intel VT-d가 활성화되어 있는지 확인합니다.

예제:

```
kvmuser@kvm-host:/$ dmesg | grep -e DMAR -e IOMMU
[    0.000000] ACPI: DMAR 0x000000006F9A4C68 000140 (v01 Cisco0 CiscoUCS 00000001 INTL 20091013)
[    0.000000] DMAR: IOMMU enabled
```

마지막 줄은 VT-d가 활성화되었음을 나타냅니다.

단계 **3** *intel_iommu=on* 매개변수를 */etc/default/grub* 컨피그레이션 파일의 GRUB_CMDLINE_LINUX 항목에 추가하여 커널에서 Intel VT-d를 활성화합니다.

예제:

```
# vi /etc/default/grub
...
GRUB_CMDLINE_LINUX="nofb splash=quiet console=tty0 ... intel_iommu=on"
...
```

참고 AMD 프로세서를 사용하는 경우 대신 부팅 매개변수에 *amd_iommu=on*을 추가합니다.

단계 **4** iommu 변경 사항을 적용하려면 서버를 재부팅합니다.

예제:

```
> shutdown -r now
```

단계 **5** 다음 형식을 사용하여 *sysfs* 인터페이스를 통해 *sriov_numvfs* 매개변수에 적절한 값을 작성하여 VF를 생성합니다.

```
#echo n > /sys/class/net/device name/device/sriov_numvfs
```

서버의 전원을 껐다가 켤 때마다 원하는 수의 VF가 생성되게 하려면 */etc/rc.d/* 디렉터리에 있는 *rc.local* 파일에 위의 명령을 추가합니다. Linux OS는 부팅 프로세스가 끝날 때 *rc.local* 스크립트를 실행합니다.

예를 들어 다음은 포트당 하나의 VF를 생성하는 방법을 보여줍니다. 인터페이스는 설정에 따라 다릅니다.

예제:

```
echo '1' > /sys/class/net/eth4/device/sriov_numvfs
echo '1' > /sys/class/net/eth5/device/sriov_numvfs
echo '1' > /sys/class/net/eth6/device/sriov_numvfs
echo '1' > /sys/class/net/eth7/device/sriov_numvfs
```

단계 **6** 서버를 재부팅합니다.

예제:

```
> shutdown -r now
```

단계 **7** *lspci*를 사용하여 VF가 생성되었는지 확인합니다.

예제:

```
 > lspci | grep -i "Virtual Function"
kvmuser@kvm-racetrack:~$ lspci | grep -i "Virtual Function"
0a:10.0 Ethernet controller: Intel Corporation 82599 Ethernet Controller Virtual Function (rev 01)
0a:10.1 Ethernet controller: Intel Corporation 82599 Ethernet Controller Virtual Function (rev 01)
0a:10.2 Ethernet controller: Intel Corporation 82599 Ethernet Controller Virtual Function (rev 01)
0a:10.3 Ethernet controller: Intel Corporation 82599 Ethernet Controller Virtual Function (rev 01)
```

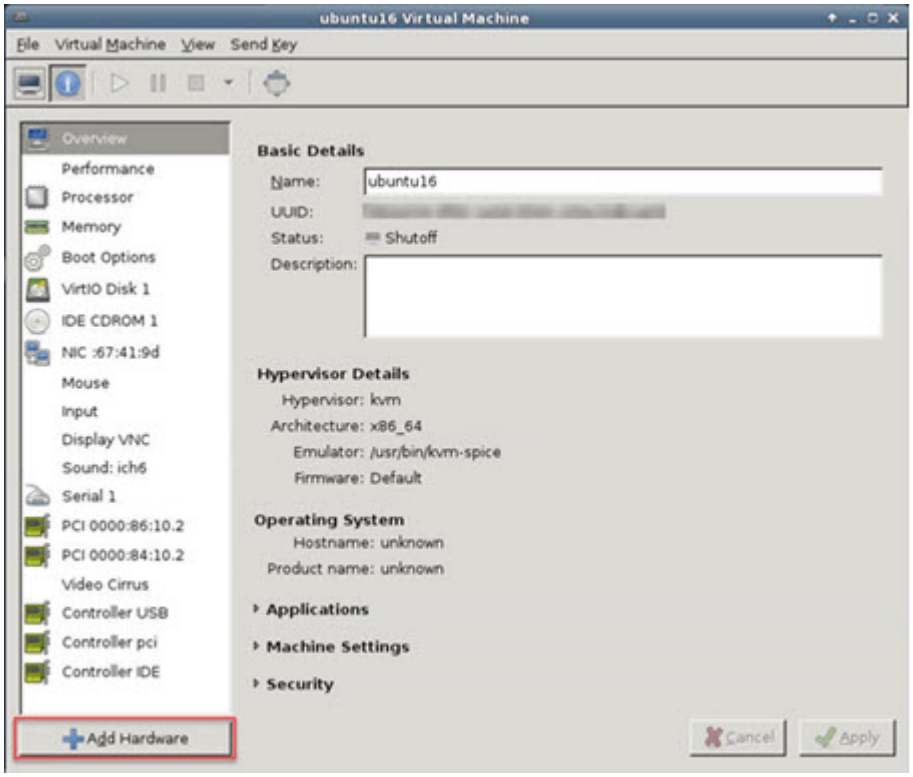
참고 **ifconfig** 명령을 사용하면 추가 인터페이스가 표시됩니다.

ASA 가상에 PCI 디바이스 할당

VF를 생성한 후에는 PCI 디바이스를 추가할 때처럼 ASA 가상에 VF를 추가할 수 있습니다. 다음 예에서는 그래픽 **virt-manager** 툴을 사용하여 이더넷 VF 컨트롤러를 ASA 가상에 추가하는 방법을 설명합니다.

단계 **1** ASA 가상을 열고 **Add Hardware(**하드웨어 추가**)** 버튼을 클릭하여 가상 머신에 새 디바이스를 추가합니다.

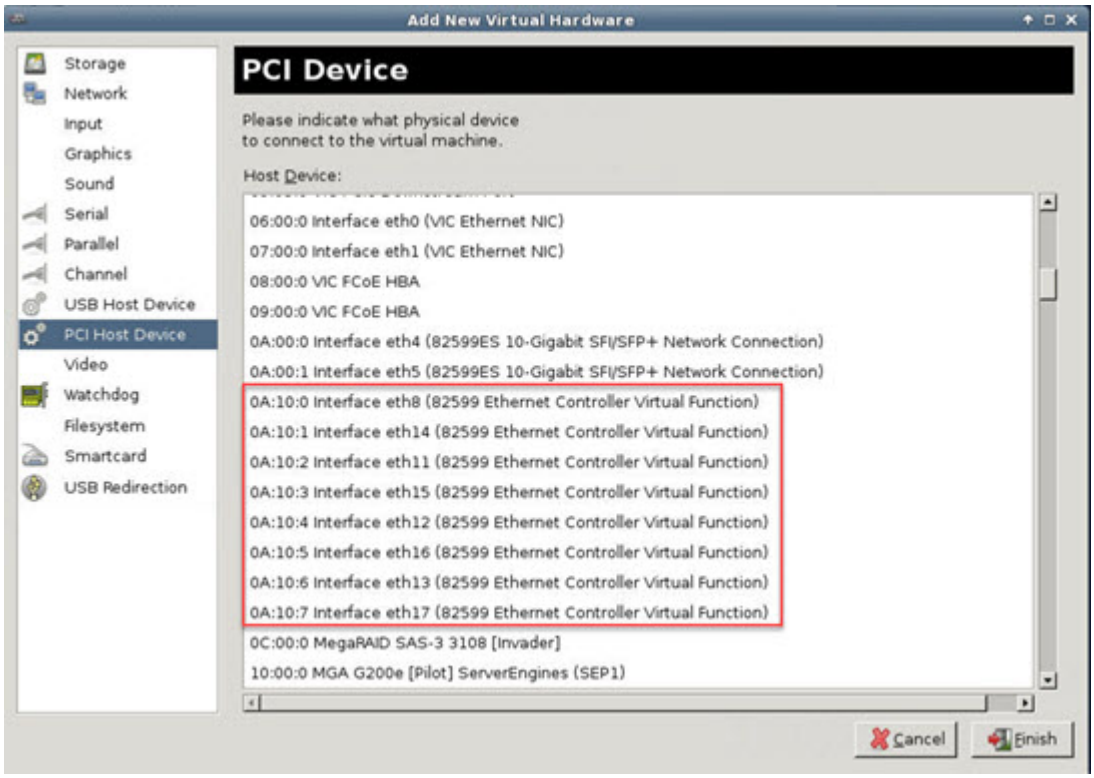
그림 ***11:*** 하드웨어 추가



단계 **2** 왼쪽 창의 **Hardware(**하드웨어**)** 목록에서 **PCI Host Device(PCI** 호스트 디바이스**)**를 클릭합니다.

VF를 포함한 PCI 디바이스 목록이 중앙 창에 표시됩니다.

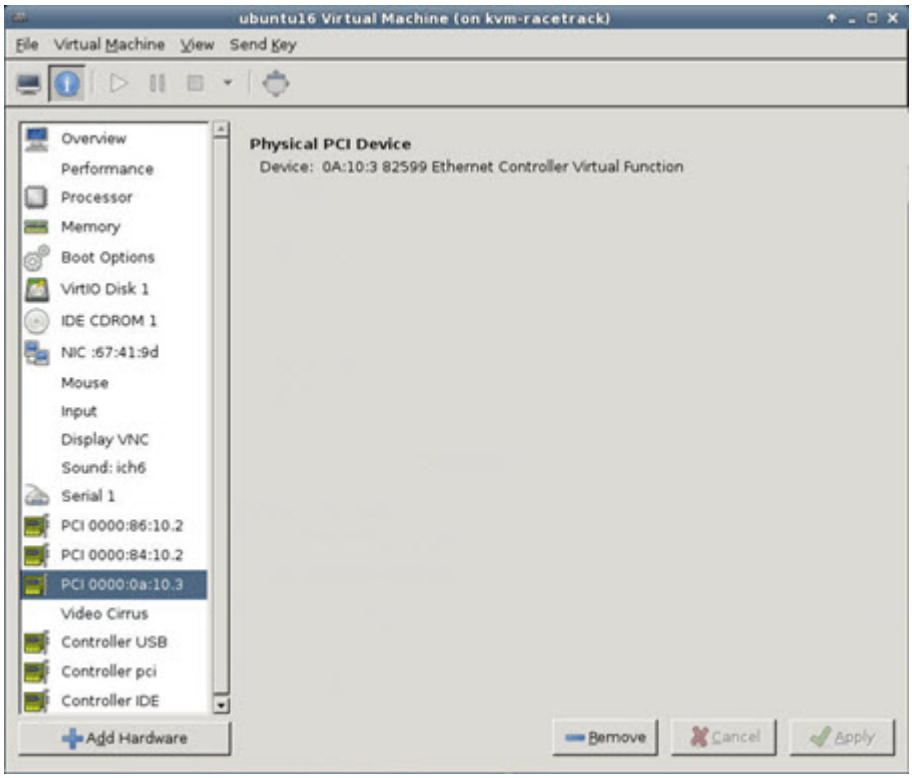
그림 *12:* 가상 기능 목록



단계 **3** 사용 가능한 가상 기능 중 하나를 선택하고 **Finish(**마침**)**를 클릭합니다.

PCI Device(PCI 디바이스)가 Hardware List(하드웨어 목록)에 표시됩니다. 이더넷 컨트롤러 가상 기능 역할을 하는 디바이스의 설명을 확인하십시오.

그림 *13:* 가상 기능 추가됨



다음에 수행할 작업

- ASA 가상명령줄에서 **show interface**명령을 사용하여, 새로 구성된 인터페이스를 확인합니다.
- 트래픽 전송 및 수신을 위해 인터페이스를 구성하고 활성화하려면 ASA 가상에서 인터페이스 구성 모드를 사용해야 합니다. 자세한 내용은 Cisco Secure Firewall ASA Series 일반적인 작업 CLI 구성 가이드의 기본 인터페이스 구성 장을 참조하십시오.

CPU 사용량 및 보고

CPU Utilization(CPU 사용률) 보고서에는 지정된 시간 내에 사용된 CPU의 백분율이 요약되어 있습니다. 일반적으로 코어는 사용량이 적은 시간에는 총 CPU 용량의 약 30~40%, 사용량이 많은 시간에는 약 60~70%로 작동합니다.

중요 9.13(1)부터 모든 ASA Virtual 라이선스는 지원되는 모든 ASA Virtual vCPU/메모리 구성에서 사용할 수 있습니다. 따라서 ASA Virtual 고객은 다양한 VM 리소스 사용 공간에서 실행할 수 있습니다.

ASA Virtual의 vCPU 사용량

ASA virtual vCPU 사용량에서는 데이터 경로, 제어 지점, 외부 프로세스에 사용된 vCPU 양을 확인할 수 있습니다.

vSphere에서 보고하는 vCPU 사용량에는 앞서 설명한 ASA virtual 사용량과 함께 다음 항목도 포함되어 있습니다.

- ASA virtual 유휴 시간
- ASA 가상 머신에 사용된 %SYS 오버헤드
- vSwitch, vNIC, pNIC 간 패킷 이동의 오버헤드. 이 오버헤드가 상당히 클 수 있습니다.

CPU 사용량의 예

show cpu usage 명령을 사용하여 CPU 사용률 통계를 표시할 수 있습니다.

예

```
Ciscoasa#show cpu usage

CPU utilization for 5 seconds = 1%; 1 minute: 2%; 5 minutes: 1%
```

다음은 보고된 vCPU 사용량이 상당한 차이를 보이는 예입니다.

- ASA Virtual 보고서: 40%
- DP: 35%
- 외부 프로세스: 5%
- ASA(ASA Virtual 보고서): 40%
- ASA 유휴 폴링: 10%
- 오버헤드: 45%

이 오버헤드는 하이퍼바이저 기능을 수행하고 vSwitch를 사용하여 NIC와 vNIC 간에 패킷을 이동하는 데 사용됩니다.

KVM CPU 사용량 보고

제공

```
virsh cpu-stats domain --total start count
```

명령은 지정된 게스트 가상 머신에 대한 CPU 통계 정보를 제공합니다. 기본적으로 모든 CPU에 대한 통계와 총계를 표시합니다. `--total` 옵션은 전체 통계만 표시합니다. `--count` 옵션은 카운트 CPU에 대한 통계만 표시합니다.

OProfile과 top 등의 툴은 하이퍼바이저와 VM 모두의 CPU 사용량을 포함하는 특정 KVM VM의 총 CPU 사용량을 제공합니다. 마찬가지로 Xen VMM 전용인 XenMon 같은 도구는 Xen 하이퍼바이저(예: Dom 0)의 총 CPU 사용량을 제공하지만 VM당 하이퍼바이저 사용량으로 분리하지는 않습니다.

또한 OpenNebula와 같은 클라우드 컴퓨팅 프레임워크에는 VM에서 사용하는 가상 CPU의 백분율 정보만 제공하는 특정 툴이 있습니다.

ASA Virtual 및 KVM 그래프

ASA Virtual과 KVM의 CPU % 수치가 다릅니다.

- KVM 그래프 수치가 항상 ASA Virtual 수치보다 높습니다.
- KVM에서는 이를 %CPU usage, ASA Virtual에서는 %CPU utilization이라고 부릅니다.

용어 "%CPU utilization"과 "%CPU usage"의 의미는 서로 다릅니다.

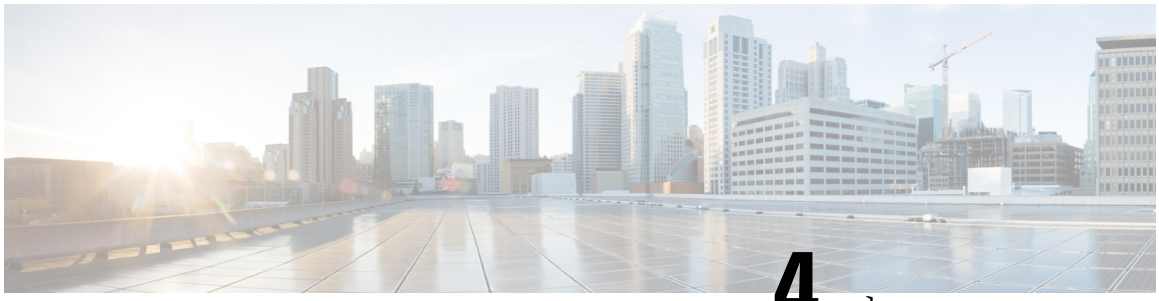
- CPU utilization은 물리적 CPU의 통계를 제공합니다.
- CPU usage는 논리적 CPU의 통계로서 CPU 하이퍼스레딩을 기반으로 합니다. 그러나 단 하나의 vCPU가 사용되므로 하이퍼스레딩은 켜져 있지 않습니다.

KVM는 %CPU usage를 다음과 같이 계산합니다.

활발하게 사용 중인 가상 CPU의 양 - 총 가용 CPU 기준 백분율로 표시

이 계산은 게스트 운영 체제가 아닌 호스트의 관점에서 본 CPU 사용량입니다. 그리고 가상 머신에 있는 사용 가능한 모든 가상 CPU의 평균 CPU 사용률입니다.

예를 들어, 가상 CPU 1개를 사용하는 가상 시스템이 4개의 물리적 CPU를 가진 호스트에서 실행되는 중이고 CPU usage가 100%라면 가상 머신에서 하나의 물리적 CPU를 온전히 사용하는 것입니다. 가상 CPU 사용량 계산: 사용량(MHz) / 가상 CPU 수 x 코어 주파수



4 장

AWS 클라우드에 ASA 가상 구축

AWS(Amazon Web Sources) 클라우드에 ASA 가상을 구축할 수 있습니다.

중요 9.13(1)부터 모든 ASA 가상 라이선스는 지원되는 모든 ASA 가상 vCPU/메모리 구성에서 사용할 수 있습니다. 따라서 ASA 가상 고객은 다양한 VM 리소스 사용 공간에서 실행할 수 있습니다. 또한 지원되는 AWS 인스턴스 유형의 수가 증가합니다.

AWS Cloud에 ASA 가상 구축 정보

ASA 가상은 물리적 ASA 와 동일한 소프트웨어를 실행하여 가상 폼 팩터에서 검증된 보안 기능을 제공합니다. ASA 가상은 퍼블릭 AWS 클라우드에 구축할 수 있습니다. 그러면 시간이 경과함에 따라 해당 위치를 확장, 축소 또는 이동하는 가상 및 물리적 데이터 센터 워크로드를 보호하기 위한 구성이 가능하게 됩니다.

ASA 가상은 다음 AWS 인스턴스 유형을 지원합니다.

표 ***13: AWS*** 지원 인스턴스 유형

Instance	특성		인터페이스
	vCPUs	메모리**(GB)**	
c5.xlarge	4	8	4
c5.2xlarge	8	16	4

Instance	특성		인터페이스
	vCPUs	메모리(GB)	
c4.large	2	3.75	3
c4.xlarge	4	7.5	4
c4.2xlarge	8	15	4
c3.large	2	3.75	3
c3.xlarge	4	7.5	4
c3.2xlarge	8	15	4
m4.large	2	4	3
m4.xlarge	4	16	4
m4.2xlarge	8	32	4
c5a.large	2	4	3
c5a.xlarge	4	8	4
c5a.2xlarge	8	16	4
c5a.4xlarge	16	32	8
c5ad.large	2	4	3
c5ad.xlarge	4	8	4
c5ad.2xlarge	8	16	4
c5ad.4xlarge	16	32	8
c5d.large	2	4	3
c5d.xlarge	4	8	4
c5d.2xlarge	8	16	4
c5d.4xlarge	16	32	8
g4ad.4xlarge	16	64	3
g4dn.xlarge	4	16	3
g4dn.2xlarge	8	32	3
g4dn.4xlarge	16	64	3
i3en.large	2	16	3
i3en.xlarge	4	32	4
i3en.2xlarge	8	64	4
i3en.3xlarge	12	96	4

Instance	특성		인터페이스
	vCPUs	메모리(GB)	
inf1.xlarge	4	8	4
inf1.2xlarge	8	16	4
m5.large	2	8	3
m5.xlarge	4	16	4
m5.2.xlarge	8	32	4
m5.4xlarge	16	64	8
m5a.large	2	8	3
m5a.xlarge	4	16	4
m5a.2xlarge	8	32	4
m5a.4xlarge	16	64	8
m5ad.large	2	8	3
m5ad.xlarge	4	16	4
m5ad.2xlarge	8	32	4
m5ad.4xlarge	16	64	8
m5d.large	2	8	3
m5d.xlarge	4	16	4
m5d.2xlarge	8	32	4
m5d.4xlarge	16	64	8
m5dn.xlarge	2	8	3
m5dn.xlarge	4	16	4
m5dn.2xlarge	8	32	4
m5dn.4xlarge	16	64	8
m5n.large	2	8	3
m5n.xlarge	4	16	4
m5n.2xlarge	8	32	4
m5n.4xlarge	16	64	8
m5zn.large	2	8	3
m5zn.xlarge	4	16	4
m5zn.2xlarge	8	32	4
m5zn.3xlarge	12	48	8
r5.large	2	16	3

Instance	특성		인터페이스
	vCPUs	메모리(GB)	
r5.xlarge	4	32	4
r5.2.xlarge	8	64	4
r5.4xlarge	16	128	8
r5a.large	2	16	3
r5a.xlarge	4	32	4
r5a.2xlarge	8	64	4
r5a.4xlarge	16	128	8
r5ad.large	2	16	3
r5ad.xlarge	4	32	4
r5ad.2xlarge	8	64	4
r5ad.4xlarge	16	128	8
r5b.large	2	16	3
r5b.xlarge	4	32	4
r5b.2xlarge	8	64	4
r5b.4xlarge	16	128	8
r5d.large	2	16	3
r5d.xlarge	4	32	4
r5d.2xlarge	8	64	4
r5d.4xlarge	16	128	8
r5dn.large	2	16	3
r5dn.xlarge	4	32	4
r5dn.2xlarge	8	64	4
r5dn.4xlarge	16	128	8
r5n.large	2	16	3
r5n.xlarge	4	32	4
r5n.2xlarge	8	64	4
r5n.4xlarge	16	128	8
z1d.large	2	16	3
z1d.xlarge	4	32	4
z1d.2xlarge	8	64	4
z1d.3xlarge	12	96	8

팁 M4 또는 C4 인스턴스 유형을 사용하는 경우, 성능 향상을 위해 Nitro 하이퍼바이저 및 ENA(Elastic Network Adapter) 인터페이스 드라이버를 사용하는 C5 또는 M5 인스턴스 유형으로 마이그레이션하는 것이 좋습니다.

AWS에서 계정을 생성하고, AWS 마법사를 사용하여 ASA 가상을 설정하고, AMI(Amazon Machine Image)를 선택합니다. AMI는 인스턴스 실행에 필요한 소프트웨어 컨피그레이션을 포함한 템플릿입니다.

중요 AMI 이미지는 AWS 환경이 아닌 곳에서 다운로드할 수 없습니다.

ASA 가상 및 AWS 사전 요건

- aws.amazon.com에서 계정을 생성합니다.
- ASA 가상에 라이선스를 부여합니다. ASA 가상 라이선스를 등록하기 전에는 저성능 모드에서 실행됩니다. 이 모드에서는 100개의 연결 및 100Kbps의 처리량만 허용됩니다. ASA 가상에 대한 라이선싱, 1 페이지의 내용을 참조하십시오.
- 인터페이스 요건:
  - 관리 인터페이스
  - 내부 및 외부 인터페이스
  - (선택 사항) 추가 서브넷(DMZ)
- 통신 경로:
  - 관리 인터페이스 - ASA 가상을 ASDM에 연결할 때 사용합니다. 통과 트래픽에는 사용할 수 없습니다.
  - 내부 인터페이스(필수)—ASA 가상를 내부 호스트에 연결하는 데 사용합니다.
  - 외부 인터페이스(필수)—ASA 가상를 공용 네트워크에 연결하는 데 사용합니다.
  - DMZ 인터페이스(선택 사항)—c3.xlarge 인터페이스 사용 시 ASA 가상을 DMZ 네트워크에 연결하는 데 사용합니다.
- ASA 가상 시스템 요구 사항은 Cisco Secure Firewall ASA호환성을 참조하십시오.

ASA 가상 및 AWS에 대한 지침과 제한 사항

지원 기능

AWS의 ASA 가상은 다음 기능을 지원합니다.

- Amazon EC2 컴퓨팅 최적화 인스턴스 제품군의 차세대 버전인 Amazon EC2 C5 인스턴스를 지원합니다.
- VPC(Virtual Private Cloud)에 구축
- 확장 네트워킹(SR-IOV) - 사용 가능한 경우
- Amazon Marketplace에서 구축
- L3 네트워크의 사용자 구축
- 라우팅 모드(기본값)
- Amazon CloudWatch

지원되지 않는 기능

AWS의 ASA 가상은 다음을 지원하지 않습니다.

- 콘솔 액세스(네트워크 인터페이스를 통해 SSH 또는 ASDM을 사용하여 관리 수행)
- VLAN
- 프로미스큐어스 모드(스니핑 또는 투명 모드 방화벽 지원 없음)
- 다중 컨텍스트 모드
- 클러스터링
- ASA 가상 기본 HA
- EtherChannel은 직접 물리적 인터페이스에서만 지원됨
- VM 가져오기/내보내기
- 하이퍼바이저 독립적 패키징
- VMware ESXi
- 브로드캐스트/멀티캐스트 메시지

  이러한 메시지는 AWS 내에서 전파되지 않으므로 브로드캐스트/멀티캐스트가 필요한 라우팅 프로토콜은 AWS에서 정상적으로 작동하지 않게 됩니다. VXLAN은 고정 피어에서만 작동할 수 있습니다.
- 불필요한/원치 않는 ARP

이러한 ARPS는 AWS 내에서 허용되지 않으므로 불필요한 ARP 또는 원치 않는 ARP가 필요한 NAT 구성은 정상적으로 작동하지 않게 됩니다.

- IPv6

구성 마이그레이션 및 SSH 인증

SSH 공개 키 인증 사용 시 업그레이드가 미치는 영향 — SSH 인증에 대한 업데이트 때문에 SSH 공개 키 인증을 활성화하려면 추가 구성이 필요합니다. 따라서 공개 키 인증을 사용하는 기존 SSH 구성은 업그레이드 후 더 이상 작동하지 않습니다. 공개 키 인증은 AWS(Amazon Web Services)에서 ASA 가상에 대한 기본값이므로 AWS 사용자에게 이 문제가 표시됩니다. SSH 연결이 손실되는 것을 방지하기 위해 업그레이드하기 전에 구성을 업데이트할 수 있습니다. 또는 업그레이드 후에 ASDM을 사용하여(ASDM 액세스를 활성화한 경우) 구성을 수정할 수 있습니다.

다음은 "admin" 사용자 이름의 샘플 원본 구성입니다.

```
username admin nopassword privilege 15
username admin attributes
  ssh authentication publickey 55:06:47:eb:13:75:fc:5c:a8:c1:2c:bb:
  07:80:3a:fc:d9:08:a9:1f:34:76:31:ed:ab:bd:3a:9e:03:14:1e:1b hashed
```

ssh authentication 명령을 사용하려면 업그레이드하기 전에 다음 명령을 입력합니다.

```
aaa authentication ssh console LOCAL
username admin password <password> privilege 15
```

nopassword 키워드가 있는 경우 이를 유지하지 않고 사용자 이름에 비밀번호를 설정하는 것이 좋습니다. **nopassword**키워드는 어떤 비밀번호든지 입력할 수 있지만 비밀번호를 비워둘 수는 없다는 뜻입니다. 9.6(2) 이전 버전의 경우 **aaa** 명령이 SSH 공개 키 인증에 필요하지 않았으므로 **nopassword** 키워드가 트리거되지 않았습니다. **aaa** 명령이 필요해짐에 따라 **password**(또는 **nopassword**) 키워드가 있는 경우 자동으로 **username**에 대한 일반 비밀번호 인증도 허용됩니다.

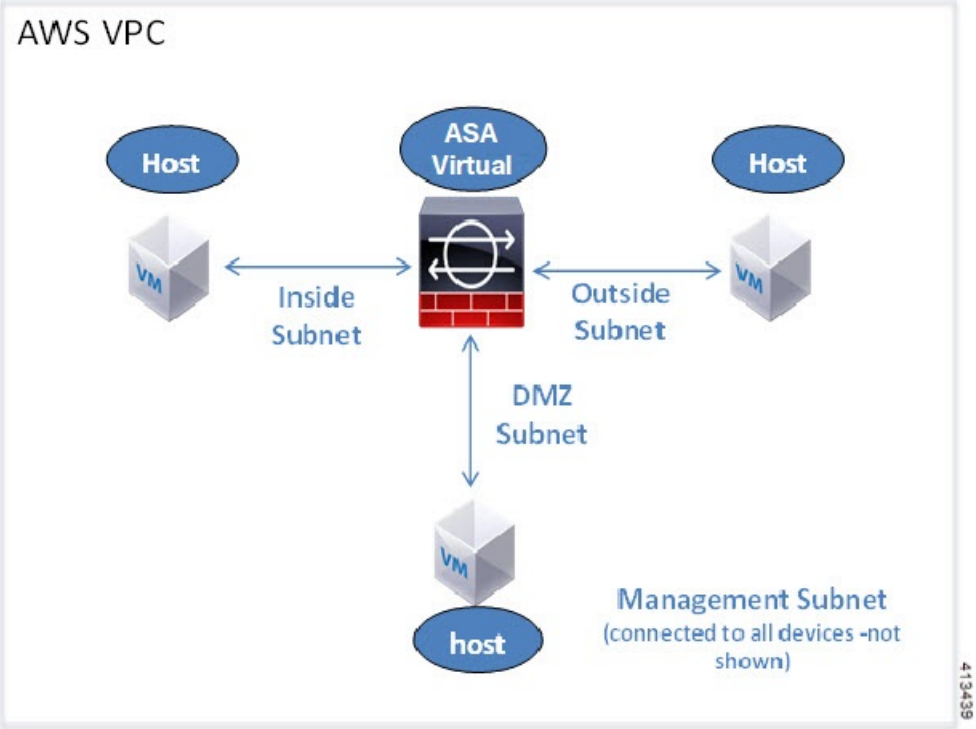
업그레이드를 하고 나면 **username** 명령에는 더 이상 **password** 또는 **nopassword** 키워드가 필요하지 않으므로 사용자가 비밀번호를 입력할 수 없도록 요청할 수 있습니다. 따라서 공개 키 인증만 강제로 적용하려면 **username** 명령을 다시 입력합니다.

```
username admin privilege 15
```

AWS 기반 ASA 가상의 샘플 네트워크 토폴로지

다음 그림에서는 Routed Firewall Mode의 ASA 가상에 대한 권장 토폴로지와 ASA 가상에 대해 AWS에 구성된 4개의 서브넷(관리, 내부, 외부 및 DMZ)을 보여줍니다.

그림 *14: AWS* 구축 기반 샘플 *ASA* 가상



AWS에 ASA 가상 구축

다음 절차는 ASA 가상에 AWS를 설정하는 단계를 간략하게 정리한 것입니다. 자세한 설정 단계는 AWS 시작하기를 참조하십시오.

단계 **1** aws.amazon.com에 로그인하고 지역을 선택합니다.

참고 AWS는 여러 지역으로 나뉘며, 이 지역은 상호 격리되어 있습니다. 화면의 우측 상단에 지역이 표시됩니다. 한 지역의 리소스가 다른 지역에는 나타나지 않습니다. 원하는 지역에 있는지 정기적으로 확인합니다.

단계 **2** **My Account(**내 계정**)** > **AWS Management Console(AWS** 관리 콘솔**)**을 클릭하고 Networking(네트워킹)에서 **VPC** > **Start VPC Wizard(VPC** 마법사 시작**)**를 클릭한 다음 단일 공용 서브넷을 선택하여 VPC를 생성하고 다음과 같이 설정합니다. 달리 표시되지 않는 한 기본 설정을 사용할 수 있습니다.

- 내부 및 외부 서브넷—VPC 및 서브넷의 이름을 입력합니다.
- 인터넷 게이트웨이—인터넷을 통한 직접 연결을 활성화합니다. 인터넷 게이트웨이의 이름을 입력합니다.
- 외부 테이블—인터넷에 대한 아웃바운드 트래픽을 활성화하려면 항목을 추가합니다. 인터넷 게이트웨이에 0.0.0.0/0을 추가합니다.

단계 **3** **My Account(**내 계정**)** > **AWS Management Console(AWS** 관리 콘솔**)** > **EC2**를 클릭한 후, **Create an Instance(**인스턴스 생성**)**를 클릭합니다.

- AMI를 선택합니다(예: Ubuntu Server 14.04 LTS).

  이미지 전달 알림에 식별된 AMI를 사용합니다.

- ASA 가상에서 지원하는 인스턴스 유형(예: c3.large)을 선택합니다.
- 인스턴스를 구성합니다. CPU 및 메모리는 고정되어 있습니다.
- **Advanced Details(**고급 세부 정보**)** 섹션을 확장하고 **User data(**사용자 데이터**)** 필드에 Day 0 컨피그레이션을 입력할 수 있습니다. 이 컨피그레이션은 ASA 가상이 시작될 때 적용된 ASA 가상 컨피그레이션을 포함하는 텍스트 입력입니다. 추가 정보(예: 스마트 라이센싱)를 이용해 Day 0 컨피그레이션에 구성하는 방법과 절차는 Day 0 컨피그레이션 파일 준비를 참조하십시오.
  - **Management interface(**관리 인터페이스**)** - Day 0 컨피그레이션을 제공하도록 선택했다면 DHCP를 사용하도록 구성해야 하는 관리 인터페이스 세부 정보를 반드시 제공해야 합니다.
  - **Data interfaces(**데이터 인터페이스**)** - 데이터 인터페이스의 IP 주소는 Day 0 컨피그레이션의 일부로서 해당 정보를 제공하는 경우에만 할당 및 구성됩니다. DHCP를 사용하도록 데이터 인터페이스를 구성할 수 있습니다. 연결할 네트워크 인터페이스가 이미 생성되었고 IP 주소가 알려진 상태라면 Day 0 컨피그레이션에 IP 세부 정보를 제공해도 됩니다.
  - **Without Day 0 Configuration(Day 0** 컨피그레이션 없음**)** - Day 0 컨피그레이션 없이 ASA 가상을 배포하는 경우, ASA 가상은 ASA 가상 컨피그레이션을 적용하여 AWS 메타데이터 서버에서 연결된 인터페이스의 IP를 가져오며 IP 주소를 할당합니다(데이터 인터페이스에서 IP를 할당하지만 ENI는 다운됩니다). Management0/0 인터페이스가 작동하며 DHCP 주소로 구성한 IP를 가져옵니다. Amazon EC2 및 Amazon VPC IP 주소 지정에 대한 자세한 내용은 VPC에서의 IP 주소 지정을 참조하십시오.
- **Sample Day 0 Configuration(**샘플 **Day 0** 컨피그레이션**)** -

```
! ASA Version 9.x.1.200
!
interface management0/0
management-only
nameif management
security-level 100
ip address dhcp setroute

no shutdown
!
crypto key generate rsa modulus 2048
ssh 0 0 management
ssh ::/0 management
ssh timeout 60
ssh version 2
username admin password Q1w2e3r4 privilege 15
username admin attributes
service-type admin
aaa authentication ssh console LOCAL
!
same-security-traffic permit inter-interface
same-security-traffic permit intra-interface
access-list allow-all extended permit ip any any
access-list allow-all extended permit ip any6 any6
```

```
access-group allow-all global
!
interface G0/0
nameif outside
ip address dhcp setroute

no shutdown
!
interface G0/1
nameif inside
ip address dhcp

no shutdown
!
```

- 스토리지(기본값 적용)
- 태그 인스턴스—다수의 태그를 생성하여 디바이스를 분류할 수 있습니다. 손쉽게 찾을 수 있도록 이름을 지정합니다.
- 보안 그룹—보안 그룹을 생성하고 이름을 지정합니다. 보안 그룹은 인바운드 및 아웃바운드 트래픽을 제어하기 위한 인스턴스에 대한 가상 방화벽입니다.

  기본적으로 보안 그룹은 모든 주소에 개방되어 있습니다. ASA 가상 액세스에 사용하는 주소의 SSH만 허용하도록 규칙을 변경합니다.
- 컨피그레이션을 검토하고 **Launch(**실행**)**를 클릭합니다.

단계 **4** 키 쌍을 생성합니다.

주의 키 쌍을 인식할 수 있는 이름을 지정하고 안전한 곳에 키를 다운로드합니다. 키는 다시 다운로드할 수 없습니다. 키 쌍을 잃어버릴 경우 인스턴스를 삭제하고 다시 구축해야 합니다.

단계 **5** **Launch Instance(**인스턴스 실행**)**를 클릭하여 ASA 가상을 구축합니다.

단계 **6** **My Account(**내 계정**)** > **AWS Management Console(AWS** 관리 콘솔**)** > **EC2** > **Launch an Instance(**인스턴스 실행**)** > **My AMIs(**내 **AMI)**를 클릭합니다.

단계 **7** ASA 가상에 대한 인터페이스 각각에서 Source/Destination Check(소스/대상 확인)가 비활성화되었는지 확인합니다.

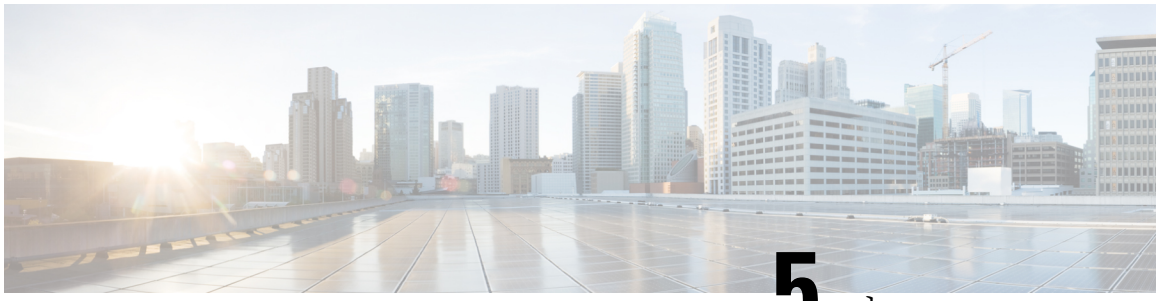
AWS 기본 설정에서는 인스턴스가 관련 IP 주소(IPv4)에 대한 트래픽만 수신하도록 허용하고, 인스턴스가 자체 IP 주소(IPv4)에서 오는 트래픽만 전송하도록 허용합니다. ASA 가상이 라우팅 홉의 역할을 할 수 있으려면 ASA 가상 트래픽 인터페이스(내부, 외부, DMZ) 각각에서 소스/대상 확인을 비활성화해야 합니다.

AWS의 ASA 가상에 대한 성능 조정

VPN 최적화

AWS c5 인스턴스는 이전 c3, c4 및 m4 인스턴스보다 훨씬 뛰어난 성능을 제공합니다. c5 인스턴스 제품군의 대략적인 RA VPN 처리량(AES-CBC 암호화를 통한 450B TCP 트래픽을 사용하는 DTLS)은 다음과 같아야 합니다.

- c5.large에서는 0.5Gbps
- c5.xlarge에서는 1Gbps
- c5.2xlarge에서는 2Gbps



5 장

AWS에 ASA 가상 Auto Scale 솔루션 구축

AWS의 Threat Defense Virtual ASA 가상용 Auto Scale 솔루션

다음 섹션에서는 AWS에서 Auto Scale 솔루션의 구성 요소가 ASA 가상에서 작동하는 방식을 설명합니다.

Auto Scale 솔루션

Cisco는 람다, 자동 확장 그룹, ELB(Elastic Load Balancing), Amazon S3 버킷, SNS 및 CloudWatch를 비롯한 여러 AWS 서비스를 사용하여 ASA 가상 방화벽의 자동 확장 그룹을 구축하기 위한 CloudFormation 템플릿 및 스크립트를 제공합니다.

AWS의 ASA 가상 Auto Scale은 완전한 서버리스 방식으로 구현되는 만큼(즉, 이 기능의 자동화와 관련된 헬퍼 VM 없음) AWS 환경의 ASA 가상 인스턴스에 수평 자동 확장 기능을 추가합니다. 버전 6.4부터 자동 확장 솔루션이 management center에서 관리하는 에서 지원됩니다.

ASA 가상 Auto Scale 솔루션은 다음을 제공하는 CloudFormation 템플릿 기반 구축입니다.

- 확장된 ASA 가상 인스턴스에 자동으로 적용되는 완전히 자동화된 컨피그레이션.
- 로드 밸런서 및 다중 가용성 영역 지원
- Auto Scale 기능 활성화 및 비활성화 지원

사용 사례

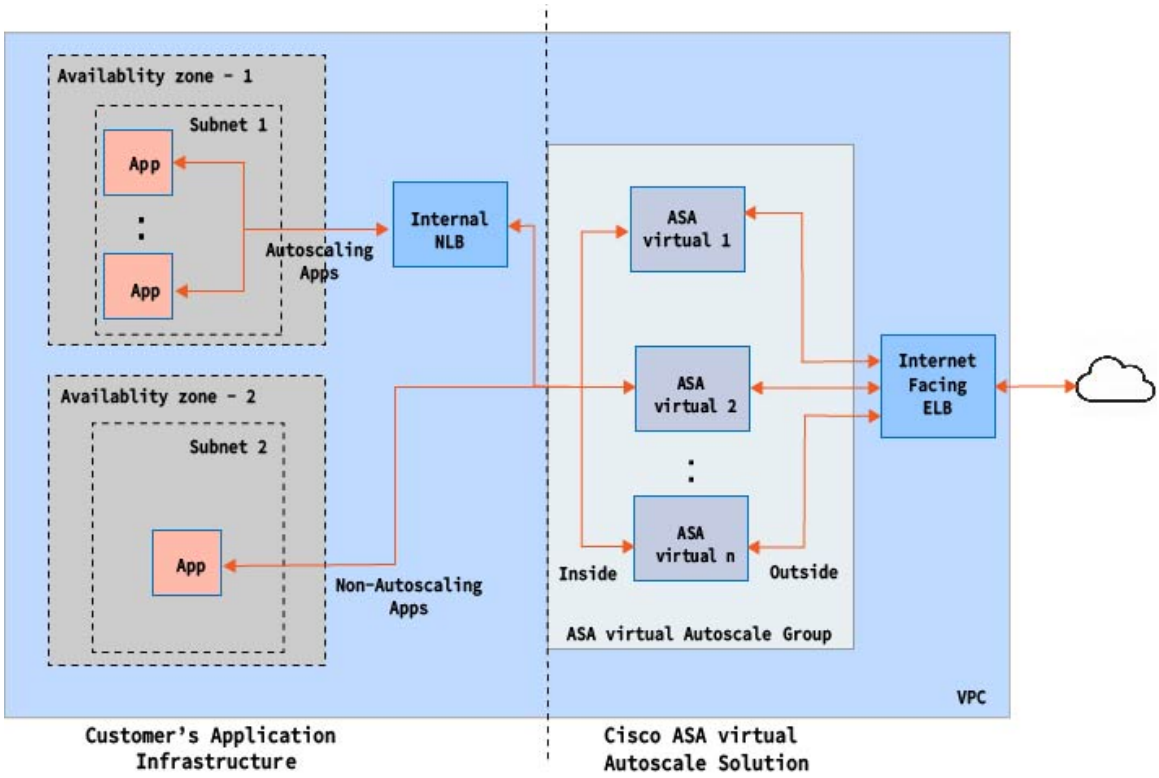
이 ASA 가상 AWS Auto Scale Solution의 사용 사례는 사용 사례 다이어그램에 표시됩니다. AWS 로드 밸런서는 인바운드 시작 연결만 허용하므로 외부에서 생성된 트래픽만 ASA 가상 방화벽을 통해 내부로 전달할 수 있습니다.

참고 보안 포트에는 SSL 서버 인증서, 88 페이지 전제 조건에 설명된 대로 SSL / TLS 인증서가 필요합니다.

인터넷 연결 로드 밸런서는 Network Load Balancer 또는 Application Load Balancer일 수 있습니다. 두 경우 모두 모든 AWS 요건 및 조건이 적용됩니다. 사용 사례 다이어그램에 나와 있는 것처럼 점선의 오른쪽은 ASA 가상 템플릿을 통해 구축됩니다. 왼쪽은 완전히 사용자 정의된 것입니다.

참고 애플리케이션 시작 아웃 바운드 트래픽은 ASA 가상을 통과하지 않습니다.

그림 *15:* 사용 사례 다이어그램



트래픽에 대한 포트 기반 분기가 가능합니다. 이 작업은 NAT 규칙을 통해 수행할 수 있습니다. 예를 들어 인터넷 연결 LB DNS, 포트 80의 트래픽은 Application-1로 라우팅될 수 있습니다. 포트: 88 트래픽을 애플리케이션-2로 라우팅할 수 있습니다.

Auto Scale 솔루션 작동 방식

ASA 가상 인스턴스를 확장 및 축소하기 위해 Auto Scale Manager라는 외부 엔터티가 메트릭을 모니터링하고, ASA 가상 인스턴스를 추가 또는 삭제하도록 자동 확장 그룹에 명령하고, ASA 가상 인스턴스를 구성합니다.

Auto Scale Manager는 AWS 서버리스 아키텍처를 사용하여 구현되고 AWS 리소스 및 ASA 가상과 통신합니다. Cisco는 Auto Scale Manager 구성 요소의 구축을 자동화하기 위해 CloudFormation 템플릿을 제공합니다. 이 템플릿은 전체 솔루션이 작동하는 데 필요한 기타 리소스도 구축합니다.

참고 서버리스 Auto Scale 스크립트는 CloudWatch 이벤트에서만 호출되므로 인스턴스가 시작될 때만 실행됩니다.

Auto Scale 솔루션 구성 요소

다음 구성 요소가 Auto Scale 솔루션을 구성합니다.

CloudFormation 템플릿

CloudFormation 템플릿은 AWS의 Auto Scale 솔루션에 필요한 리소스를 구축하는 데 사용합니다. 템플릿은 다음으로 구성됩니다.

- 자동 확장 그룹, 로드 밸런서, 보안 그룹 및 기타 기타 구성 요소
- 템플릿은 사용자 입력에 따라 구축을 맞춤화합니다.

참고 템플릿에는 사용자 입력을 검증하는 데 제한이 있으므로 구축 중에 입력을 검증하는 것은 사용자의 책임입니다.

람다 함수

Auto Scale 솔루션은 Python으로 개발한 람다 함수의 집합으로서 라이프사이클 후크, SNS, CloudWatch 이벤트/알람 이벤트에서 트리거됩니다. 기본 기능은 다음과 같습니다.

- 인스턴스에 Gig0/0 및 Gig 0/1 인터페이스를 추가/제거합니다.
- 로드 밸런서의 대상 그룹에 Gig0/1 인터페이스를 등록합니다.
- ASA 컨피그레이션 파일을 사용하여 새 ASA 가상를 구성하고 구축합니다.

람다 함수는 Python 패키지 형식으로 고객에게 제공됩니다.

라이프 사이클 후크

- 라이프 사이클 후크는 인스턴스에 대한 라이프 사이클 변경 알림을 가져오는 데 사용됩니다.

- 인스턴스 시작의 경우 라이프 사이클 후크를 사용하여 ASA 가상 인스턴스에 인터페이스를 추가하고 대상 그룹에 외부 인터페이스 IP를 등록할 수 있는 람다 함수를 트리거합니다.
- 인스턴스가 종료되는 경우, 라이프사이클 후크를 사용하여 대상 그룹에서 ASA 가상 인스턴스의 등록을 취소하는 람다 함수를 트리거합니다.

간편 알림 서비스**(SNS)**

- AWS의 SNS(Simple Notification Service)를 사용하여 이벤트를 생성합니다.
- AWS에는 서버리스 람다 함수에 적합한 오케스트레이터가 없다는 제한 때문에, 솔루션은 SNS를 일종의 기능 체인으로 사용하여 이벤트를 기반으로 람다 함수를 오케스트레이션합니다.

Auto Scale 솔루션 사전 요건

구축 파일 다운로드

ASA 가상 Auto Scale for AWS 솔루션을 시작하는 데 필요한 파일을 다운로드합니다. 사용자 ASA 버전의 구축 스크립트 및 템플릿은 GitHub 리포지토리에서 제공됩니다.

주의 Cisco에서 제공하는 자동 확장용 구축 스크립트 및 템플릿은 오픈 소스 예시로 제공되며 일반적인 Cisco TAC 지원 범위에서는 다루지 않습니다. GitHub에서 정기적으로 업데이트 및 ReadMe 지침을 확인하십시오.

인프라 구성

복제/다운로드된 GitHub 리포지토리에서 **infrastructure.yaml** 파일은 템플릿 폴더에 있습니다. 이 CFT는 버킷 정책을 통해 VPC, 서브넷, 경로, ACL, 보안 그룹, VPC 엔드 포인트 및 S3 버킷을 구축하는 데 사용할 수 있습니다. 이 CFT는 요구 사항에 맞게 수정할 수 있습니다.

다음 섹션에서는 이러한 리소스 및 해당 리소스가 Auto Scale에서 사용되는 방법에 대해 자세히 설명합니다. 이러한 리소스를 수동으로 구축하고 Auto Scale에서도 사용할 수 있습니다.

참고 **infrastructure.yaml** 템플릿은 VPC, 서브넷, ACL, 보안 그룹, S3 버킷 및 VPC 엔드 포인트만 구축합니다. SSL 인증서, 람다 레이어 또는 KMS 키 리소스는 생성하지 않습니다.

VPC

애플리케이션 요구 사항에 따라 VPC를 생성해야 합니다. VPC에는 인터넷에 대한 경로가 연결된 하나 이상의 서브넷이 있는 인터넷 게이트웨이가 있어야 합니다. 보안 그룹, 서브넷 등에 대한 요구 사항은 해당 섹션을 참조하십시오.

서브넷

필요할 경우 애플리케이션 요구 사항에 따라 서브넷을 생성할 수 있습니다. 사용 사례에서처럼 작동하려면 ASA 가상 머신에는 3개의 서브넷이 필요합니다.

참고 다중 가용성 영역 지원이 필요한 경우 서브넷은 AWS 클라우드 내의 영역 속성이므로 각 영역에 서브넷이 필요합니다.

외부 서브넷

외부 서브넷에는 인터넷 게이트웨이에 대한 기본 경로가 '0.0.0.0/0'이어야 합니다. 여기에는 ASA 가상의 외부 인터페이스가 포함되며 인터넷 연결 NLB도 이 서브넷에 포함됩니다.

내부 서브넷

이는 NAT/인터넷 게이트웨이가 있거나 없는 애플리케이션 서브넷과 유사할 수 있습니다. ASA 가상 상태 프로브의 경우 포트 80을 통해 AWS 메타데이터 서버(169.254.169.254)에 연결할 수 있어야 합니다.

참고 이 AutoScale 솔루션에서 로드 밸런서 상태 프로브는 inside/Gig0/0 인터페이스를 통해 AWS 메타 데이터 서버로 리디렉션됩니다. 그러나 로드 밸런서에서 ASA 가상로 전송되는 상태 프로브 연결을 제공하는 고유한 애플리케이션을 사용하여 이를 변경할 수 있습니다. 이 경우 상태 프로브 응답을 제공하려면 AWS Metadata Server 개체를 해당 애플리케이션 IP 주소로 교체해야 합니다.

관리 서브넷

이 서브넷에는 ASA 가상 관리 인터페이스가 포함되어 있습니다. 기본 경로를 사용하는 것은 선택 사항입니다.

람다 서브넷

AWS 람다 함수를 사용하려면 NAT 게이트웨이가 기본 게이트웨이인 두 개의 서브넷이 필요합니다. 이렇게 하면 VPC 전용의 람다 함수가 생성됩니다. 람다 서브넷은 다른 서브넷만큼 넓을 필요는 없습니다. 람다 서브넷에 대한 모범 사례는 AWS 설명서를 참조하십시오.

애플리케이션 서브넷

Auto Scale 솔루션에서 이 서브넷에 적용되는 제한은 없지만, 애플리케이션이 VPC 외부에서 아웃 바운드 연결을 필요로 하는 경우 서브넷에 각각의 경로가 구성되어 있어야 합니다. 이는 아웃 바운드에서 시작된 트래픽이 로드 밸런서를 통과하지 않기 때문입니다. AWS Elastic Load Balancing User Guide를 참조하십시오.

보안 그룹

제공된 Auto Scale 그룹 템플릿에서 모든 연결이 허용됩니다. Auto Scale 솔루션이 작동하려면 다음 연결만 필요합니다.

표 ***14:*** 필수 포트

포트	사용	서브넷
상태 프로브 포트 (기본값: 8080)	인터넷 연결 로드 밸런서 상태 프로브	외부, 내부 서브넷
애플리케이션 포트	애플리케이션 데이터 트래픽	외부, 내부 서브넷

Amazon S3 버킷

Amazon Simple Storage Service(Amazon S3)는 업계 최고의 확장성, 데이터 가용성, 보안 및 성능을 제공하는 개체 스토리지 서비스입니다. 방화벽 템플릿 및 애플리케이션 템플릿 모두에 필요한 모든 파일을 S3 버킷에 담을 수 있습니다.

템플릿이 구축되면 S3 버킷의 Zip 파일을 참조하는 람다 함수가 생성됩니다. 따라서 사용자 계정에서 S3 버킷에 액세스할 수 있어야 합니다.

SSL 서버 인증서

인터넷 연결 로드 밸런서가 TLS / SSL을 지원해야 하는 경우 인증서 ARN이 필요합니다. 자세한 내용은 다음 링크를 참조하십시오.

- 서버 인증서 작업
- 테스트를 위한 개인 키 및 자체 서명 인증서 생성
- 자체 서명 SSL 인증서로 AWS ELB 생성(서드 파티 링크)

ARN의 예: arn:aws:iam::[AWS 계정]:server-certificate/[인증서 이름]

람다 레이어

autoscale_layer.zip은 Linux 환경(예: Python 3.9가 설치된 Ubuntu 18.04)에서 생성할 수 있습니다.

```
#!/bin/bash
mkdir -p layer
virtualenv -p /usr/bin/python3.9 ./layer/
source ./layer/bin/activate
pip3 install cffi==1.15.1
pip3 install cryptography==2.9.1
pip3 install paramiko==2.7.1
pip3 install requests==2.23.0
pip3 install scp==0.13.2
pip3 install jsonschema==3.2.0
pip3 install pycryptodome==3.15.0
echo "Copy from ./layer directory to ./python\n"
cp -r ./layer/lib/python3.9/site-packages/* ./python/
zip -r autoscale_layer.zip ./python
```

결과 ***autoscale_layer.zip*** 파일을 *lambda-python-files* 폴더에 복사해야 합니다.

KMS 마스터 키

ASA 가상 비밀번호가 암호화된 형식인 경우 필요합니다. 그렇지 않으면 이 구성 요소가 필요하지 않습니다. 비밀번호는 여기에 제공된 KMS만 사용하여 암호화해야 합니다. KMS ARN이 CFT에 입력된 경우 비밀번호를 암호화해야 합니다. 그렇지 않으면 비밀번호는 일반 텍스트여야 합니다.

마스터 키 및 암호화에 대한 자세한 내용은 AWS 문서 Creating keys 및 AWS CLI Command Reference 에서 비밀번호 암호화 및 KMS에 대한 내용을 참조하십시오.

예:

```
$ aws kms encrypt --key-id <KMS-ARN> --plaintext 'MyC0mplIc@tedProtect1oN'
{
    "KeyId": "KMS-ARN",
    "CiphertextBlob":
"AQICAHgcQFAGtz/hvaxMtJvY/x/rfHnKI3clFPpSXUU7HQRnCAFwfXhXHJAHL8tcVmDqurALAAAAajBoBgkqhki
G9w0BBwagWzBZAgEAMFQGCSqGSIb3DQEHATAeBglghkgBZQMEAS4wEQQM45AIkTqjSekX2mniAgEQgCcOav6Hhol
+wxpWKtXY4y1Z1d0z1P4fx0jTdosfCbPnUExmNJ4zdx8="
}
$
```

CiphertextBlob 키의 값을 비밀번호로 사용해야 합니다.

Python 3 환경

make.py 파일은 복제된 리포지토리의 최상위 디렉토리에 있습니다. 이렇게하면 python 파일을 Zip 파일로 압축하고 대상 폴더에 복사합니다. 이러한 작업을 수행하려면 Python 3 환경을 사용할 수 있어야 합니다.

Auto Scale 구축

준비

애플리케이션이 구축되었거나 구축 계획을 사용할 수 있어야 합니다.

입력 매개변수

다음 입력 매개 변수는 구축 전에 수집해야 합니다.

참고 AWS Gateway Load Balancer(GWLB)의 경우 **LoadBalancerType**, **LoadBalancerSG**, **LoadBalancerPort** 및 **SSLcertificate** 매개변수는 사용할 수 없습니다.

표 ***15: Auto Scale*** 입력 매개 변수

파라미터	허용되는 값 / 유형	설명
PodNumber	문자열 허용되는 패턴: '^\d{1,3}$'	Pod 번호입니다. 이는 Auto Scale 그룹 이름(ASA 가상-Group-Name)의 접미사입니다. 예를 들어 이 값이 '1'인 경우 그룹 이름은 *ASA 가상-Group-Name-1*이 됩니다. 숫자는 한 자릿수 이상 세 자릿수 이하여야 합니다. 기본값: 1
AutoscaleGrpNamePrefix	문자열	Auto Scale 그룹 이름 접두사입니다. Pod 번호는 접미사로 추가됩니다. 최대 문자수: 18자 예: Cisco-ASA 가상-1
NotifyEmailID	문자열	Auto Scale 이벤트가 이 이메일 주소로 전송됩니다. 구독 이메일 요청을 수락해야 합니다. 예: admin@company.com
VpcId	문자열	디바이스를 구축해야 하는 VPC ID입니다. 이는 AWS 요건에 따라 구성해야 합니다. 유형: AWS::EC2::VPC::Id "*infrastructure.yaml*" 파일을 사용하여 인프라를 구축하는 경우 스택의 출력 섹션에이 값이 지정됩니다. 해당 값을 사용하십시오.

파라미터	허용되는 값 / 유형	설명
LambdaSubnets	목록	람다 함수가 구축될 서브넷 유형: List<AWS::EC2::Subnet::Id> "*infrastructure.yaml*" 파일을 사용하여 인프라를 구축하는 경우 스택의 출력 섹션에이 값이 지정됩니다. 해당 값을 사용하십시오.
LambdaSG	목록	람다 함수의 보안 그룹 유형: List<AWS::EC2::SecurityGroup::Id> "*infrastructure.yaml*" 파일을 사용하여 인프라를 구축하는 경우 스택의 출력 섹션에이 값이 지정됩니다. 해당 값을 사용하십시오.
S3BktName	문자열	파일의 S3 버킷 이름입니다. 이는 AWS 요건에 따라 사용자 계정에서 구성해야 합니다. "*infrastructure.yaml*" 파일을 사용하여 인프라를 구축하는 경우 스택의 출력 섹션에이 값이 지정됩니다. 해당 값을 사용하십시오.
LoadBlancerType	문자열	인터넷 연결 로드 밸런서의 유형(“애플리케이션”또는 “네트워크”)입니다. 예: application
LoadBlancerSG	문자열	로드 밸런서의 보안 그룹 네트워크 로드 밸런서의 경우에는 사용되지 않습니다. 그러나 보안 그룹 ID를 제공해야 합니다. 유형: List<AWS::EC2::SecurityGroup::Id> "*infrastructure.yaml*" 파일을 사용하여 인프라를 구축하는 경우 스택의 출력 섹션에이 값이 지정됩니다. 해당 값을 사용하십시오.
LoadBlancerPort	정수	로드 밸런서 포트 이 포트는 선택한 로드 밸런서 유형에 따라 프로토콜로 HTTP / HTTPS 또는 TCP / TLS를 사용하여 LB에서 열립니다. 포트가 유효한 TCP 포트인지 확인합니다. 이 포트는 로드 밸런서 리스너를 생성하는 데 사용됩니다. 기본값: 80

파라미터	허용되는 값 / 유형	설명
SSL인증서	문자열	보안 포트 연결을 위한 SSL 인증서의 ARN입니다. 지정하지 않으면 로드 밸런서에서 열린 포트는 TCP / HTTP가 됩니다. 지정된 경우 로드 밸런서에서 열린 포트는 TLS / HTTPS입니다.
TgHealthPort	정수	이 포트는 상태 프로브의 대상 그룹에서 사용됩니다. ASA 가상에서 이 포트에 도착하는 상태 프로브는 AWS 메타데이터 서버로 라우팅되며 트래픽에 사용해서는 안 됩니다. 유효한 TCP 포트여야 합니다. 애플리케이션 자체가 상태 프로브에 응답하도록 하려면 ASA 가상에 따라 NAT 규칙을 변경할 수 있습니다. 이 경우 애플리케이션이 응답하지 않으면 ASA 가상가 비정상 상태로 표시되고 비정상 인스턴스 임계 값 알람으로 인해 삭제됩니다. 예: 8080
AssignPublicIP	부울	"true"로 선택된 경우 공용 IP가 할당됩니다. BYOL 유형 ASA 가상의 경우 https://tools.cisco.com에 연결해야 합니다. 예: true
ASAv 인스턴스 유형	문자열	AMI(Amazon Machine Image)는 인스턴스의 크기와 필요한 메모리 양을 결정하는 다양한 인스턴스 유형을 지원합니다. ASA 가상을 지원하는 AMI 인스턴스 유형만 사용해야 합니다. 예: c4.2xlarge
ASAv LicenseType	문자열	ASA 가상 라이선스 유형(BYOL 또는 PAYG) 관련 AMI ID가 동일한 라이선싱 유형인지 확인합니다. 예: BYOL
ASAv AmiId	문자열	ASA 가상 AMI ID(유효한 Cisco ASA 가상 AMI ID) 유형: AWS::EC2::Image::Id 영역 및 원하는 이미지 버전에 따라 올바른 AMI ID를 선택하십시오.

파라미터	허용되는 값 / 유형	설명
ConfigFileURL	문자열	ASA 가상 컨피그레이션 파일의 HTTP URL입니다. 각 AZ에 대한 컨피그레이션 파일은 URL에서 사용할 수 있어야 합니다. 람다 함수가 올바른 파일을 선택합니다. HTTP 서버를 구축하여 컨피그레이션 파일을 호스팅하거나 AWS S3 정적 웹 호스팅 기능을 사용할 수 있습니다. 참고 마지막 "/"도 필요한데, 이는 컨피그레이션 파일 이름을 가져올 때 URL에 적용되기 때문입니다. "*infrastructure.yaml*" 파일을 사용하여 인프라를 구축하는 경우 스택의 출력 섹션에이 값이 지정됩니다. 해당 값을 사용하십시오. 예: https : //myserver/asavconfig/asaconfig.txt/
NoOfAZs	정수	ASA 가상이 1~3의 범위에 걸쳐 있어야 하는 가용성 영역의 수입니다. ALB 구축의 경우 AWS에 필요한 최소값은 2입니다. 예: 2
ListOfAzs	쉼표로 구분된 문자열	쉼표로 구분된 영역의 목록(순서대로) 참고 나열되는 순서가 중요합니다. 서브넷 목록은 동일한 순서로 제공되어야 합니다. "*infrastructure.yaml*" 파일을 사용하여 인프라를 구축하는 경우 스택의 출력 섹션에이 값이 지정됩니다. 해당 값을 사용하십시오. 예: us-east-1a, us-east-1b, us-east-1c
ASAvMgmtSubnetId	쉼표로 구분된 목록	쉼표로 구분된 관리 서브넷 ID 목록입니다. 목록은 해당 가용 영역과 동일한 순서여야 합니다. 유형: List<AWS::EC2::SecurityGroup::Id> "*infrastructure.yaml*" 파일을 사용하여 인프라를 구축하는 경우 스택의 출력 섹션에이 값이 지정됩니다. 해당 값을 사용하십시오.

파라미터	허용되는 값 / 유형	설명
ASAvInsideSubnetId	쉼표로 구분된 목록	쉼표로 구분된 inside/Gig0/0 ID 목록입니다. 목록은 해당 가용 영역과 동일한 순서여야 합니다. 유형: List<AWS::EC2::SecurityGroup::Id> "*infrastructure.yaml*" 파일을 사용하여 인프라를 구축하는 경우 스택의 출력 섹션에이 값이 지정됩니다. 해당 값을 사용하십시오.
ASAvOutsideSubnetId	쉼표로 구분된 목록	쉼표로 구분된 outside/Gig0/1 ID 목록입니다. 목록은 해당 가용 영역과 동일한 순서여야 합니다. 유형: List<AWS::EC2::SecurityGroup::Id> "*infrastructure.yaml*" 파일을 사용하여 인프라를 구축하는 경우 스택의 출력 섹션에이 값이 지정됩니다. 해당 값을 사용하십시오.
KmsArn	문자열	기존 KMS의 ARN(대기시 암호화 할 AWS KMS 키) 지정된 경우 ASA 가상 비밀번호를 암호화해야 합니다. 비밀번호 암호화는 지정된 ARN만 사용하여 수행해야 합니다. 암호화된 비밀번호 생성 예 : "aws kmscrypt --key-id<KMS ARN> --plaintext<password> ". 표시된 대로 생성된 비밀번호를 사용하십시오. 예: arn:aws:kms:us-east-1:[AWS 계정]:key/7d586a25-5875-43b1-bb68-a452e2f6468e
CpuThresholds	쉼표로 구분된 정수	CPU 하한 임계값 및 CPU 상한 임계값 최소값은 0이고 최대값은 99입니다. 기본값: 10, 70 하한 임계값은 상한 임계값보다 작아야 합니다. 예: 30, 70

ASA 컨피그레이션 파일 업로드

ASA 컨피그레이션 파일을 준비하고 ASA 가상 인스턴스에서 액세스할 수 있는 http/https 서버에 저장합니다. 표준 ASA 컨피그레이션 파일 형식을 사용합니다. 확장된 ASA 가상에서 컨피그레이션 파일을 다운로드하고 관련 컨피그레이션을 업데이트합니다.

다음 섹션에서는 Auto Scale 솔루션을 대상으로 ASA 컨피그레이션 파일을 수정하는 방법에 대한 예를 확인할 수 있습니다.

개체, 디바이스 그룹, **NAT** 규칙 및 액세스 정책

ASA 가상 컨피그레이션의 로드 밸런서 상태 프로브에 대한 개체, 경로 및 NAT 규칙의 예는 다음을 참조하십시오.

```
! Load Balancer Health probe Configuration
object network aws-metadata-server
host 169.254.169.254
object service aws-health-port
service tcp destination eq 7777
object service aws-metadata-http-port
service tcp destination eq 80
route inside 169.254.169.254 255.255.255.255 10.0.100.1 1
nat (outside,inside) source static any interface destination static interface
aws-metadata-server service aws-health-port aws-metadata-http-port
!
```

참고 위의 상태 프로브 연결은 액세스 정책에서 허용되어야 합니다.

ASA 가상 컨피그레이션의 데이터 플레인 컨피그레이션의 예는 다음을 참조하십시오.

```
! Data Plane Configuration
route inside 10.0.0.0 255.255.0.0 10.0.100.1 1
object network http-server-80
host 10.0.50.40
object network file-server-8000
host 10.0.51.27
object service http-server-80-port
service tcp destination eq 80
nat (outside,inside) source static any interface destination static interface http-server-80
 service http-server-80-port http-server-80-port
object service file-server-8000-port
service tcp destination eq 8000
nat (outside,inside) source static any interface destination static interface file-server-8000
 service file-server-8000-port file-server-8000-port
object service https-server-443-port
service tcp destination eq 443
nat (outside,inside) source static any interface destination static interface http-server-80
 service https-server-443-port http-server-80-port
!
```

컨피그레이션 파일 업데이트

ASA 가상 컨피그레이션은 *az1-connfiguration.txt*, *az2-configuration.txt* 및 *az3-configuration.txt* 파일에서 업데이트해야 합니다.

참고 3개의 컨피그레이션 파일이 있으면 가용성 영역(AZ)에 따라 컨피그레이션을 수정할 수 있습니다. 예를 들어 aws-metadata-server에 대한 고정 경로는 각 AZ에서 다른 게이트웨이를 갖습니다.

템플릿 업데이트

deploy_autoscale.yaml 템플릿은 신중하게 수정해야 합니다. ***LaunchTemplate***의 *UserData* 필드를 수정해야 합니다. 필요에 따라 *UserData*를 업데이트할 수 있습니다. 그에 따라 *name-server*를 업데이트해야 합니다. 예를 들어 VPC DNS IP일 수 있습니다. 라이선싱이 BYOL인 경우 라이선싱 *idtoken*을 공유해야 합니다.

```
!
dns domain-lookup management
DNS server-group DefaultDNS
name-server <VPC DNS IP>
!
! License configuration
        call-home
        profile License
        destination transport-method http
        destination address http <url>
        license smart
        feature tier standard
        throughput level <entitlement>
        license smart register idtoken <token>
```

Amazon Simple Storage Service(S3)로 파일 업로드

대상 디렉토리의 모든 파일을 Amazon S3 버킷에 업로드해야 합니다. 원할 경우 CLI를 사용하여 대상 디렉토리의 모든 파일을 Amazon S3 버킷에 업로드할 수 있습니다.

```
$ cd ./target
$ aws s3 cp . s3://<bucket-name> --recursive
```

스택 구축

구축을 위한 모든 전제 조건이 완료되면 AWS CloudFormation 스택을 생성할 수 있습니다.

대상 디렉토리의 *deploy_autoscale.yaml* 파일을 사용합니다.

Geneve Autoscale의 대상 디렉토리의 *deploy_ngfw_autoscale_with_gwlb.yaml* 파일을 사용합니다.

참고 *deploy_ngfw_autoscale_with_gwlb.yaml* 파일을 구축하기 전에 AWS GWLB Auto Scale 솔루션용 **Infrastructure_gwlb.yaml** 파일을 구축해야 합니다.

deploy_autoscale_with_gwlb.yaml 템플릿 구축 중에 생성되는 GWLB를 선택하여 GWLB-E(Gateway Loadbalancer Endpoint)를 생성해야 합니다. GWLBe를 생성한 후에는 Application Subnet(애플리케이션 서브넷) 및 default Route(기본 경로) 테이블에 GWLBe를 사용하도록 기본 경로를 업데이트해야 합니다.

자세한 내용은 https://docs.amazonaws.cn/en_us/vpc/latest/privatelink/create-endpoint-service-gwlbe.html를 참고하십시오.

입력 매개변수, 90 페이지에 수집된 매개 변수를 제공합니다.

구축 검증

템플릿 구축이 완료되면, 람다 함수 및 CloudWatch 이벤트가 생성되었는지 검증해야 합니다. 기본적으로 Auto Scale 그룹에는 최소 및 최대 인스턴스 수가 0입니다. 원하는 인스턴스 수를 사용하여 AWS EC2 콘솔에서 Auto Scale그룹을 편집해야 합니다. 그러면 새 ASA 가상 인스턴스가 트리거됩니다.

인스턴스를 하나만 실행하고 그 워크플로우를 확인하고 예상대로 작동하는지에 대한 동작을 검증하는 것이 좋습니다. ASA 가상의 실제 요구 사항을 구축한 후에는 동작에 대해 확인할 수도 있습니다. AWS Scaling 정책에서 ASA 가상 인스턴스가 제거되지 않도록 최소 인스턴스 수를 축소 보호로 표시할 수 있습니다.

Auto Scale 유지 보수 작업

확장 프로세스

이 항목에서는 Auto Scale 그룹에 대해 하나 이상의 확장 프로세스를 일시 중지한 다음 다시 시작하는 방법을 설명합니다.

확장 작업 시작 및 중지

스케일 아웃/인 작업을 시작하고 중지하려면 다음 단계를 수행합니다.

- AWS Dynamic Scaling의 경우 - 다음 링크를 참조하여 스케일 아웃 작업을 활성화하거나 비활성화할 수 있습니다.

  확장 프로세스 일시 중단 및 다시 시작

상태 모니터

CloudWatch Cron 작업은 60분마다 Health Doctor 모듈의 Auto Scale Manager 람다를 트리거합니다.

- 유효한 ASA 가상 VM에 속한 비정상적인 IP가 있는 경우 ASA 가상가 1시간 이상 경과하면 해당 인스턴스가 삭제됩니다.
- 해당 IP가 유효한 ASA 가상 머신에 있지 않으면 대상 그룹에서 IP만 제거됩니다.

상태 모니터 비활성화

상태 모니터를 비활성화하려면 ***constant.py***에서 상수를 "True"로 지정합니다.

상태 모니터 활성화

상태 모니터를 활성화하려면 ***consist.py***에서 상수를 "False"로 지정합니다.

라이프 사이클 후크 비활성화

라이프 사이클 후크를 비활성화해야 하는 경우는 드물지만 비활성화되면 인스턴스에 인터페이스를 추가하지 않습니다. 또한 일련의 ASA 가상 인스턴스 구축이 실패할 수 있습니다.

Auto Scale Manager 비활성화

Auto Scale Manager를 비활성화하려면 각 CloudWatch 이벤트 "notify-instance-launch" 및 "notify-instance-terminate"를 비활성화해야 합니다. 이 기능을 비활성화하면 새 이벤트에 대해 람다가 트리거되지 않습니다. 그러나 이미 실행중인 람다 작업은 계속 진행됩니다. Auto Scale Manager는 갑자기 중지되지 않습니다. 스택 삭제 또는 리소스 삭제로 인해 갑자기 중지하려고 시도하면 무한 상태가 발생할 수 있습니다.

로드 밸런서 대상

AWS로드 밸런서는 둘 이상의 네트워크 인터페이스가 있는 인스턴스에 대해 인스턴스 유형 대상을 허용하지 않으므로 Gigabit0/1 인터페이스 IP는 대상 그룹에서 대상으로 구성됩니다. 그러나 현재 AWS Auto Scale 상태 확인은 IP가 아닌 인스턴스 유형 대상에 대해서만 작동합니다. 또한 이러한 IP는 대상 그룹에서 자동으로 추가되거나 제거되지 않습니다. 따라서 Auto Scale 솔루션은 이러한 두 작업을 모두 프로그래밍 방식으로 처리합니다. 그러나 유지 보수 또는 문제 해결의 경우에는 수동으로 수행해야 하는 상황이 있을 수 있습니다.

대상 그룹에 대상 등록

로드 밸런서에 ASA 가상 인스턴스를 등록하려면 Gigabit0/1 인스턴스 IP(외부 서브넷)를 대상 그룹의 대상으로 추가해야 합니다. Register or Deregister Targets by IP Address를 참조하십시오.

대상 그룹에서 대상 등록 취소

로드 밸런서에 ASA 가상 인스턴스를 등록 해제하려면 Gigabit0/1 인스턴스 IP(외부 서브넷)를 대상 그룹에서 삭제해야 합니다. Register or Deregister Targets by IP Address를 참조하십시오.

인스턴스 스탠바이

AWS는 Auto Scale 그룹에서 인스턴스 재부팅을 허용하지 않지만 사용자가 인스턴스를 스탠바이 상태로 설정하고 이러한 작업을 수행할 수 있도록 허용합니다. 그러나 이는 로드 밸런서 대상이 인스턴스 유형인 경우에 가장 적합합니다. 그러나 복수의 네트워크 인터페이스 때문에 ASA 가상 머신은 인스턴스 유형 대상으로 구성할 수 없습니다.

인스턴스를 스탠바이 상태로 설정

인스턴스가 스탠바이 상태가 되면 대상 그룹의 해당 IP는 상태 프로브가 실패할 때까지 계속 동일한 상태로 유지됩니다. 따라서 인스턴스를 스탠바이 상태로 설정하기 전에 대상 그룹에서 각 IP를 등록 취소하는 것이 좋습니다. 자세한 내용은 대상 그룹에서 대상 등록 취소, 98 페이지를 참조하십시오.

IP가 제거되면 Temporarily Removing Instances from Your Auto Scaling Group을 참조하십시오.

스탠바이에서 인스턴스 제거

마찬가지로 인스턴스를 스탠바이 상태에서 실행 중 상태로 이동할 수 있습니다. 스탠바이 상태에서 제거한 후에는 인스턴스의 IP를 대상 그룹 대상에 등록해야 합니다. 대상 그룹에 대상 등록, 98 페이지의 내용을 참조하십시오.

문제 해결 또는 유지 보수를 위해 인스턴스를 스탠바이 상태로 설정하는 방법에 대한 자세한 내용은 AWS 뉴스 블로그를 참조하십시오.

Auto Scale 그룹에서 인스턴스 제거/분리

Auto Scale 그룹에서 인스턴스를 제거하려면 먼저 스탠바이 상태로 이동해야 합니다. "Put Instances on Stand-by"를 참조하십시오. 인스턴스가 스탠바이 상태가 되면 제거하거나 분리할 수 있습니다. Detach EC2 Instances from Your Auto Scaling Group을 참조하십시오.

인스턴스 종료

인스턴스를 종료하려면 스탠바이 상태로 설정해야 합니다. 인스턴스 스탠바이, 98 페이지을 참조하십시오. 인스턴스가 스탠바이 상태가 되면 종료를 진행할 수 있습니다.

인스턴스 축소 보호

Auto Scale 그룹에서 특정 인스턴스가 실수로 제거되는 것을 방지하기 위해 축소(scale in) 보호로 설정할 수 있습니다. 인스턴스가 축소(Scale-In) 보호 상태에 있을 경우 해당 인스턴스는 축소 이벤트로 인해 종료되지 않습니다.

인스턴스를 축소 보호 상태로 전환하려면 다음 링크를 참조하십시오.

https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-instance-termination.html

중요 상태가 양호한 최소 인스턴스 수(대상 IP는 EC2 인스턴스가 아니라 정상이어야 함)를 축소 보호하는 것이 좋습니다.

자격 증명 및 등록 ID 변경

컨피그레이션의 변경 사항은 이미 실행중인 인스턴스에 자동으로 반영되지 않습니다. 변경 사항은 향후 디바이스에만 반영됩니다. 이러한 변경 사항은 기존 디바이스에 수동으로 푸시해야 합니다.

ASA Virtual 관리자 비밀번호 변경

ASA 가상 비밀번호를 변경하려면 사용자가 실행중인 인스턴스에 대해 각 디바이스에서 비밀번호를 수동으로 변경해야 합니다. 새 ASA 가상 디바이스를 온보딩할 경우, ASA 가상 비밀번호는 람다 환경 변수에서 가져옵니다. Using AWS Lambda Environment Variables을 확인하십시오.

AWS 리소스 변경

Auto Post Group, Launch Configuration(컨피그레이션 시작), CloudWatch 이벤트, 확장 정책 등 AWS 사후 구축에서 여러 가지 사항을 변경할 수 있습니다. 리소스를 CloudFormation 스택으로 가져 오거나 기존 리소스에서 새 스택을 생성할 수 있습니다.

AWS 리소스에서 수행되는 변경 사항을 관리하는 방법에 대한 자세한 내용은 Bringing Existing Resources Into CloudFormation Management를 참조하십시오.

CloudWatch 로그 수집 및 분석

CloudWatch 로그를 내보내려면 Export Log Data to Amazon S3 Using the AWS CLI를 참조하십시오.

Auto Scale 문제 해결 및 디버깅

AWS CloudFormation 콘솔

AWS CloudFormation 콘솔에서 CloudFormation 스택에 대한 입력 매개 변수를 확인할 수 있습니다. 그러면 웹 브라우저에서 직접 스택을 생성, 모니터링, 업데이트 및 삭제할 수 있습니다.

필요한 스택으로 이동하여 매개 변수 탭을 확인합니다. 또한 람다 함수 환경 변수 탭에서 람다 함수에 대한 입력을 확인할 수도 있습니다.

AWS CloudFormation 콘솔에 대한 자세한 내용은 *AWS CloudFormation User Guide*를 참조하십시오.

Amazon Cloudwatch 로그

개별적인 람다 함수의 로그를 볼 수 있습니다. AWS 람다는 사용자를 대신하여 람다 함수를 자동으로 모니터링하며 Amazon CloudWatch를 통해 메트릭을 보고합니다. 함수에서 장애를 해결하는 데 도움이 되도록 람다는 함수에서 처리한 모든 요청을 기록하고, 코드에서 생성된 로그를 Amazon CloudWatch Logs를 통해 자동으로 저장합니다.

람다 콘솔, CloudWatch 콘솔, AWS CLI 또는 CloudWatch API를 사용하여 람다에 대한 로그를 볼 수 있습니다. CloudWatch 콘솔을 통해 로그 그룹에 액세스하고 액세스하는 방법에 대한 자세한 내용은 *Amazon CloudWatch User Guide*의 모니터링 시스템, 애플리케이션 및 맞춤형 로그 파일을 참조하십시오.

로드 밸런서 상태 확인 실패

로드 밸런서 상태 확인에는 프로토콜, ping 포트, ping 경로, 응답 시간 초과, 상태 확인 간격 등의 정보가 포함됩니다. 상태 확인 간격 내에 200 응답 코드를 반환하는 인스턴스는 정상 상태로 간주됩니다.

일부 또는 모든 인스턴스의 현재 상태가 `OutOfService`이고 설명 필드에 `인스턴스가 최소한 비정상 상태 임계 횟수 이상 실패했다`는 메시지가 표시되면 인스턴스가 로드 밸런서 상태 검사에 실패한 것입니다.

ASA 컨피그레이션에서 상태 프로브 NAT 규칙을 확인해야 합니다. 자세한 내용은 Troubleshoot a Classic Load Balancer: Health checks를 참조하십시오.

트래픽 문제

ASA 가상 인스턴스의 트래픽 문제를 해결하려면 로드 밸런서 규칙, NAT 규칙 및 ASA 가상 인스턴스에 구성된 고정 경로를 확인해야 합니다.

또한 보안 그룹 규칙 등 구축 템플릿에 제공된 AWS 가상 네트워크 / 서브넷 / 게이트웨이 세부 정보도 확인해야 합니다. Troubleshooting EC2 instances와 같은 AWS 설명서를 참조할 수도 있습니다.

ASA 가상 구성 실패

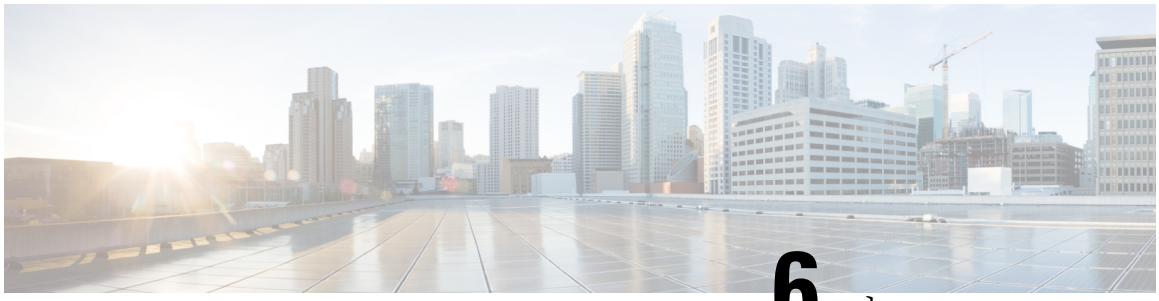
ASA 가상를 구성에 실패한다면 Amazon S3 고정 HTTP 웹서버 호스팅 컨피그레이션에 대한 연결을 확인해야 합니다. 자세한 내용은 Hosting a static website on Amazon S3을 참조하십시오.

ASA 가상 라이선스 실패

ASA 가상이 라이선스에 실패하면 CSSM 서버에 대한 연결을 확인하고 ASA 가상 보안 그룹 컨피그레이션을 확인한 다음 액세스 제어 목록을 확인해야 합니다.

ASA Virtual에 SSH를 할 수 없음

SSH를 ASA 가상로 연결할 수 없는 경우 템플릿을 통해 복잡한 비밀번호가 ASA 가상에 전달되었는지 확인합니다.



6 장

Microsoft Azure 클라우드에 ASA 가상 구축

Microsoft Azure 클라우드에 ASA 가상를 구축할 수 있습니다.

중요 9.13(1)부터 모든 ASA 가상 라이선스는 지원되는 모든 ASA 가상 vCPU/메모리 구성에서 사용할 수 있습니다. 따라서 ASA 가상 고객은 다양한 VM 리소스 사용 공간에서 실행할 수 있습니다. 또한 지원되는 Azure 인스턴스 유형의 수가 증가합니다.

Microsoft Azure 클라우드에 ASA 가상 구축

ASA 가상 필요에 맞는 Azure 가상 머신 유형 및 크기를 선택합니다. 모든 ASA 가상 라이선스는 지원되는 ASA 가상 vCPU/메모리 설정에서 사용할 수 있습니다. 이렇게 하면 다양한 Azure 인스턴스 유형에서 ASA 가상을 실행할 수 있습니다.

표 ***16: Azure*** 지원 인스턴스 유형

Instance	특성		인터페이스
	vCPUs	메모리**(GB)**	
D3, D3_v2, DS3, DS3_v2	4	14	4

Instance	특성		인터페이스
	vCPUs	메모리(GB)	
D4, D4_v2, DS4, DS4_v2	8	28	8
D5, D5_v2, DS5, DS5_v2	16	56	8
D8_v3	8	32	4
D16_v3	16	64	4
D8s_v3	8	32	4
D16s_v3	16	64	8
F4, F4s	4	8	4
F8, F8s	8	16	8
F16, F16s	16	32	8
F8s_v2	8	16	4
F16s_v2	16	32	8

Microsoft Azure에 ASA 가상을 구축할 수 있습니다.

- 표준 Azure 퍼블릭 클라우드 및 Azure Government 환경에서 Azure Resource Manager를 사용하여 독립형 방화벽으로 구축합니다.
- Azure Security Center를 사용하여 통합된 파트너 솔루션으로 구축합니다.
- 표준 Azure 퍼블릭 클라우드 환경과 Azure Government 환경에서 Azure Resource Manager를 사용하여 HA(고가용성) 쌍으로 구축합니다.

Azure Resource Manager에서 ASA 가상 구축, 112 페이지의 내용을 참조하십시오. 표준 Azure 퍼블릭 클라우드 및 Azure Government 환경에서 ASA 가상 HA 구성을 구축할 수 있습니다.

ASA 가상 및 Azure의 사전 요건과 시스템 요구 사항

- Azure.com에서 계정을 생성합니다.

  Microsoft Azure에서 계정을 생성한 다음, 로그인하고 Microsoft Azure Marketplace에서 ASA 가상을 선택하여 ASA 가상을 구축합니다.

- ASA 가상에 라이선스를 부여합니다.

  ASA 가상 라이선스를 등록하기 전에는 저성능 모드에서 실행됩니다. 이 모드에서는 100개의 연결 및 100Kbps의 처리량만 허용됩니다. ASA 가상을 위한 Smart Software Licensing을 참조하십시오.

참고 Azure에서 구축될 때 ASA 가상 기본값은 2Gbps 엔타이틀먼트가 됩니다. 100Mbps 및 1Gbps 엔타이틀먼트를 사용할 수 있습니다. 그러나 처리량 수준은 100Mbps 또는 1Gbps 엔타이틀먼트를 사용하도록 명시적으로 구성되어야 합니다.

- 인터페이스 요구 사항:

  4개의 네트워크에서 4개의 인터페이스로 ASA 가상을 구축해야 합니다. 공용 IP 주소를 다른 인터페이스에 할당할 수 있습니다. 공용 IP 주소를 생성, 변경 또는 삭제하는 방법을 비롯하여 퍼블릭 IP와 관련된 Azure 지침은 Public IP addresses(퍼블릭 IP 주소)를 참조하십시오.

- 관리 인터페이스:

  Azure에서는 처음 정의되는 인터페이스가 언제나 을 포함하지 않습니다.

- 통신 경로:
  - 관리 인터페이스—SSH 액세스에 그리고 ASDM에 ASA 가상을 연결하는 데 사용합니다.

참고 Azure 가속 네트워킹은 관리 인터페이스에서 지원되지 않습니다.

  - 내부 인터페이스(필수)—ASA 가상를 내부 호스트에 연결하는 데 사용합니다.
  - 외부 인터페이스(필수)—ASA 가상를 공용 네트워크에 연결하는 데 사용합니다.
  - DMZ 인터페이스(선택 사항)—Standard_D3 인터페이스 사용 시 ASA 가상을 DMZ 네트워크에 연결하는 데 사용합니다.

- ASA 가상 하이퍼바이저 및 가상 플랫폼 지원 관련 정보는 Cisco Secure Firewall ASA 호환성을 참조하십시오.

지침 및 제한 사항

지원 기능

- Microsoft Azure 클라우드에서 구축
- Azure Accelerated Networking(AN)
- 선택한 인스턴스 유형에 따라 최대 16개의 vCPU

참고 Azure는 구성 가능한 L2 vSwitch 기능을 제공하지 않습니다.

- 임의의 인터페이스상의 공용 IP 주소

  공용 IP 주소를 다른 인터페이스에 할당할 수 있습니다. 공용 IP 주소를 생성, 변경 또는 삭제하는 방법을 비롯하여 퍼블릭 IP와 관련된 Azure 지침은 Public IP addresses(퍼블릭 IP 주소)를 참조하십시오.

- 라우팅 방화벽 모드(기본)

참고 라우팅 방화벽 모드의 ASA 가상는 네트워크의 일반 레이어 3 경계입니다. 이 모드에서는 각 인터페이스에 IP 주소가 필요합니다. Azure에서는 VLAN 태깅 인터페이스를 지원하지 않으므로 태그가 지정되지 않은 비 트렁크 인터페이스에서 IP 주소를 구성해야 합니다.

- IPv6

알려진 문제

유휴 시간 제한

Azure의 ASA 가상에는 VM에서 구성 가능한 유휴 시간 제한이 있습니다. 최소 설정은 4분이고 최대 설정은 30분입니다. 그러나 SSH 세션의 경우 최소 설정은 5분이고 최대 설정은 60분입니다.

참고 ASA 가상의 유휴 시간 제한은 항상 SSH 시간 제한을 재정의하며 세션 연결을 끊는다는 사실에 유의하십시오. 세션이 어느 쪽에서도 시간 초과되지 않도록 VM의 유휴 시간 제한을 SSH 시간 초과와 일치시킬 수도 있습니다.

기본 **ASA** 가상에서 스탠바이 **ASA** 가상로의 장애 조치

Azure 구축의 ASA 가상 HA에서 Azure 업그레이드가 발생하면, 기본 ASA 가상에서 스탠바이 ASA 가상로의 페일오버가 발생할 수 있습니다. Azure를 업그레이드하면 기본 ASA 가상이 일시 중지 상태가 됩니다. 기본 ASA 가상이 일시 중지되면 스탠바이 ASA 가상은 hello 패킷을 수신하지 않습니다. 스탠바이 ASA 가상이 장애 조치 보류 시간이 끝날 때까지 hello 패킷을 받지 못하면, 스탠바이 ASA 가상에 대한 페일오버가 발생합니다.

장애 조치 보류 시간이 초과되지 않았지만 페일오버가 발생할 수도 있습니다. 기본 ASA 가상이 일시 중지 상태로 들어간 후 19초 후에 다시 시작되는 상황을 고려해보십시오. 페일오버 보류 시간은 30초입니다. 그러나 스탠바이 ASA 가상은 시계가 2분마다 동기화되므로 올바른 타임스탬프가 포함된 hello 패킷을 수신하지 못합니다. 따라서 기본 ASA 가상에서 스탠바이 ASA 가상로의 페일오버가 발생합니다.

참고 이 기능은 IPv4만 지원하며, IPv6 구성에서는 ASA Virtual HA가 지원되지 않습니다.

지원되지 않는 기능

- 콘솔 액세스(네트워크 인터페이스를 통해 SSH 또는 ASDM을 사용하여 관리 수행)
- 사용자 인스턴스 인터페이스의 VLAN 태깅
- 점보 프레임
- Azure의 관점에서는 디바이스 소유가 아닌 IP 주소에 대한 프록시 ARP
- 프로미스큐어스 모드(스니핑 또는 투명 모드 방화벽 지원 없음)

참고 Azure 정책에서는 ASA 가상이 투명 방화벽 모드에서 작동할 수 없습니다. 이 모드에서는 인터페이스가 프로미스큐어스 모드에서 작동할 수 없기 때문입니다.

- 멀티컨텍스트 모드
- 클러스터링
- ASA 가상 기본 HA
- VM 가져오기/내보내기
- 기본적으로 FIPS 모드는 Azure 클라우드에서 실행 중인 ASA 가상에 대해 활성화되지 않습니다.

참고 FIPS 모드를 활성화할 경우 **ssh key-exchange group dh-group14-sha1** 명령을 사용하여 Diffie-Helman 키 교환 그룹을 더 강력한 키로 변경해야 합니다. Diffie-Helman 그룹을 변경하지 않으면 ASA 가상에 대한 SSH를 사용할 수 없습니다. 이는 초기에 ASA 가상를 관리할 수 있는 유일한 방법입니다.

- IPv6

Azure DDoS 보호 기능

Microsoft Azure의 Azure DDoS Protection은 ASA 가상의 최전선에서 구현되는 추가 기능입니다. 가상 네트워크에서 이 기능을 활성화하면 네트워크 예상 트래픽의 초당 패킷 수에 따라 일반적인 네트워크 레이어 공격으로부터 애플리케이션을 방어할 수 있습니다. 네트워크 트래픽 패턴에 따라 이 기능을 맞춤화할 수 있습니다.

Azure DDoS Protection 기능에 대한 자세한 내용은 Azure DDoS Protection 표준 개요를 참고하십시오.

구축 중에 생성된 리소스

Azure에서 ASA 가상를 구축할 때 다음 리소스가 생성됩니다.

- ASA 가상 머신
- 리소스 그룹(기존 리소스 그룹을 선택하지 않는 한)

  ASA 가상 리소스 그룹은 가상 네트워크 및 스토리지 계정에서 사용하는 것과 동일한 리소스 그룹이어야 합니다.
- 4개의 NIC - vm name-Nic0, vm name-Nic1, vm name-Nic2, vm name-Nic3

  이 NIC는 ASA 가상의 인터페이스인 Management 0/0, GigabitEthernet 0/0, GigabitEthernet 0/1, GigabitEthernet 0/2에 각각 매핑됩니다.

참고 요구 사항에 따라 IPv4 전용 을 사용하여 Vnet을 생성할 수 있습니다.

- vm name-SSH-SecurityGroup이라는 보안 그룹

  보안 그룹이 VM의 Nic0에 매핑되며, 이는 ASA 가상 Management 0/0에 매핑됩니다.

  보안 그룹은 VPN 목적으로 SSH 및 UDP 포트 500, UDP 4500을 허용하는 규칙을 포함합니다. 구축 후에 이 값을 수정할 수 있습니다.
- 공용 IP 주소(구축 중에 선택한 값에 따라 이름이 지정됨)

  공용 IP 주소(IPv4 전용)를 모든 인터페이스에 할당할 수 있습니다.

  공용 IP 주소를 생성, 변경 또는 삭제하는 방법을 비롯하여 퍼블릭 IP와 관련된 Azure 지침은 Public IP addresses(퍼블릭 IP 주소)를 참조하십시오.
- 4개의 서브넷이 있는 가상 네트워크(기존 네트워크를 선택하지 않은 경우)
- 각 서브넷에 대한 라우팅 테이블(이미 있을 경우 업데이트됨)

  이 테이블의 이름은 subnet name-ASAv-RouteTable입니다.

  각 라우팅 테이블에는 다른 3개 서브넷에 대한 경로가 포함되며 ASA 가상 IP 주소가 다음 홉입니다. 트래픽이 다른 서브넷 또는 인터넷에 도달해야 하는 경우 기본 경로 추가를 선택할 수 있습니다.
- 선택된 스토리지 계정의 부팅 진단 파일

  부팅 진단 파일은 Blob(binary large object)에 포함됩니다.
- Blob과 컨테이너 VHD인 *vm name*-disk.vhd 및 *vm name*-<uuid>.status에 속한 선택된 스토리지 계정의 파일 2개
- 스토리지 계정(기존 스토리지 계정을 선택하지 않은 경우)

참고 VM을 삭제할 경우 이 리소스에서 유지할 것을 제외하고 각각을 개별적으로 삭제해야 합니다.

Azure 라우팅

Azure 가상 네트워크의 라우팅은 가상 네트워크의 유효 라우팅 테이블에 따라 결정됩니다. 유효 라우팅 테이블은 기존 시스템 라우팅 테이블과 사용자 정의 라우팅 테이블의 조합입니다.

참고 ASA 가상은 Azure 클라우드 라우팅 특성 때문에 EIGRP나 OSPF 같은 동적 내부 라우팅 프로토콜을 사용할 수 없습니다. 가상 클라이언트에 정적/동적 경로가 구성되었는지에 상관없이, 유효 라우팅 테이블이 따라 다음 홉이 결정됩니다.

현재는 유효 라우팅 테이블과 시스템 라우팅 테이블 중 어느 쪽도 볼 수 없습니다.

사용자 정의 라우팅 테이블은 보고 수정할 수 있습니다. 시스템 테이블과 사용자 정의 테이블의 조합으로 유효 라우팅 테이블이 구성될 경우 가장 구체적인 경로가 선택되며 동등할 때는 사용자 정의 라우팅 테이블이 적용됩니다. 시스템 라우팅 테이블은 Azure의 가상 네트워크 인터넷 게이트웨이를 가리키는 기본 경로(0.0.0.0/0)를 포함합니다. 시스템 라우팅 테이블은 나머지 정의된 서브넷에 대한 경로도 포함하는데, 다음 홉은 Azure의 가상 네트워크 인프라 게이트웨이를 가리킵니다.

ASA 가상 구축 프로세스에서는 ASA 가상을 통과하도록 트래픽을 라우팅하기 위해 각 서브넷에 대한 경로를 다음 홉으로 ASA 가상을 사용 중인 나머지 세 서브넷에 추가합니다. 서브넷의 ASA 가상 인터페이스를 가리키는 기본 경로(0.0.0.0/0)를 추가하려는 경우도 있습니다. 그러면 서브넷의 모든 트래픽이 ASA 가상을 통과합니다. 따라서 이 트래픽 처리를 위해 (아마도 NAT/PAT를 사용하여) 미리 ASA 가상 정책이 구성되어야 할 수도 있습니다.

시스템 라우팅 테이블의 기존 경로 때문에 ASA 가상를 다음 홉으로 가리키는 경로를 사용자 정의 라우팅 테이블에 추가해야 합니다. 그렇지 않으면 사용자 정의 테이블의 기본 경로가 시스템 라우팅 테이블의 더 구체적인 경로에 밀려 트래픽이 ASA 가상를 우회하게 됩니다.

가상 네트워크의 VM을 위한 라우팅 컨피그레이션

Azure 가상 네트워크의 라우팅은 클라이언트의 특정 게이트웨이 설정이 아니라 유효 라우팅 테이블에 따라 달라집니다. 가상 네트워크에서 실행 중인 클라이언트는 DHCP에서 경로를 지정할 수도 있습니다. 이는 해당 서브넷의 1번 주소입니다. 이는 자리 표시자이며 가상 네트워크의 인프라 가상 게이트웨이에 패킷을 보내는 기능만 할 뿐입니다. 패킷이 VM을 떠나면 유효 라우팅 테이블에 따라 (사용자 정의 테이블에서 수정한 대로) 라우팅됩니다. 클라이언트가 .1 또는 ASA 가상 주소로 구성된 경우에도 유효 라우팅 테이블에 따라 다음 홉이 결정됩니다.

Azure VM ARP 테이블에서는 모든 확인된 호스트에 대해 동일한 MAC 주소(1234.5678.9abc)를 표시합니다. 그러면 Azure VM을 떠나는 모든 패킷이 Azure 게이트웨이에 도달하며, 여기서 유효 라우팅 테이블을 사용하여 패킷의 경로를 결정합니다.

참고 ASA 가상은 Azure 클라우드 라우팅 특성 때문에 EIGRP나 OSPF 같은 동적 내부 라우팅 프로토콜을 사용할 수 없습니다. 가상 클라이언트에 정적/동적 경로가 구성되었는지에 상관없이, 유효 라우팅 테이블이 따라 다음 홉이 결정됩니다.

IP 주소

다음 정보가 Azure의 IP 네트워크에 적용됩니다.

- DHCP를 사용하여 ASA 가상 인터페이스의 IP 주소를 설정해야 합니다.

  Azure 인프라는 Azure에서 설정된 IP 주소가 ASA 가상 인터페이스에 지정되게 합니다.

- Management 0/0는 연결된 서브넷에서 전용 IP 주소를 받습니다.

  공용 IP 주소를 이 전용 IP 주소와 연결할 수 있으며 Azure Internet 게이트웨이에서 NAT 변환을 처리합니다.

- 모든 인터페이스에 공용 IP 주소를 할당할 수 있습니다.
- 동적 공용 IP 주소는 Azure 중지/시작 사이클에 변경될 수 있습니다. 그러나 Azure가 재시작하고 ASA 가상가 다시 로드될 때는 유지됩니다.
- 고정 공용 IP 주소는 Azure에서 변경하지 않는 한 바뀌지 않습니다.

DNS

모든 Azure 가상 네트워크는 내장된 DNS 서버인 168.63.129.16에 액세스할 수 있으며, 이는 다음과 같이 사용 가능합니다.

```
configure terminal
dns domain-lookup management
dns server-group DefaultDNS
 name-server 168.63.129.16
end
```

스마트 라이센싱을 구성할 때 자체 DNS 서버가 설정되지 않은 경우 이 컨피그레이션을 사용할 수 있습니다.

Accelerated Networking(AN)

Azure의 AN(Accelerated Networking) 기능은 VM에 대한 SR-IOV(Single Root I / O Virtualization)를 활성화합니다. 그러면 VM NIC가 하이퍼바이저를 우회하여 PCIe 카드로 바로 이동할 수 있습니다. AN은 VM의 처리량 성능을 크게 향상시키며 추가 코어(예: 더 큰 VM)로 확장됩니다.

기본적으로 비활성화되어 있습니다. Azure는 사전 프로비저닝된 가상 머신에서 AN 활성화를 지원합니다. Azure에서 VM을 중지하고 네트워크 카드 속성을 업데이트하여 *enableAcceleratedNetworking* 매

개 변수를 true로 설정하기만 하면 됩니다. Microsoft 문서 Enable accelerated networking on existing VMs를 참조하십시오. 그런 다음 VM을 다시 시작합니다.

Mellanox 하드웨어 지원

Microsoft Azure 클라우드에는 AN 기능을 지원하는 두 가지 유형의 하드웨어인 Mellanox 4(MLX4)와 Mellanox 5(MLX5)가 있습니다. ASA 가상은 릴리스 9.15부터 다음 인스턴스에 대해 Mellanox 하드웨어용 AN을 지원합니다.

- D3, D3_v2, DS3, DS3_v2
- D4, D4_v2, DS4, DS4_v2
- D5, D5_v2, DS5, DS5_v2
- D8_v3, D8s_v3
- D16_v3, D16s_v3
- F4, F4s
- F8, F8s, F8s_v2
- F16, F16s, F16s_v2

참고 MLX4(Mellanox 4)는 connectx3 = cx3이라고도 하며, MLX5(Mellanox 5)는 connectx4 = cx4라고도 합니다.

VM 구축을 위해 NIC Azure에서 MLX4와 MLX5 중 한쪽을 사용하도록 지정할 수는 없습니다. 가속화된 네트워킹 기능을 사용하려면 ASA 가상을 9.15 이상 버전으로 업그레이드하는 것이 좋습니다.

Microsoft Azure에 ASA 가상 구축

Microsoft Azure에 ASA 가상을 구축할 수 있습니다.

- 표준 Azure 퍼블릭 클라우드 및 Azure Government 환경에서 Azure Resource Manager를 사용하여 ASA 가상을 독립형 방화벽으로 구축합니다. Azure Resource Manager에서 ASA 가상 구축을 참조하십시오.
- Azure Security Center를 사용하여 Azure 내에서 통합된 파트너 솔루션으로 ASA 가상을 구축합니다. 보안에 민감한 고객에게는 Azure 워크로드를 보호하기 위한 방화벽 옵션으로 ASA 가상이 제공됩니다. 보안 및 상태 이벤트는 단일한 통합 대시보드에서 모니터링됩니다. Azure Security Center에서 ASA 가상 구축을 참조하십시오.
- Azure Resource Manager를 사용하여 ASA 가상 고가용성 쌍을 구축합니다. 리던던시를 보장하려면 액티브/백업 고가용성(HA) 구성으로 ASA 가상를 구축하면 됩니다. 퍼블릭 클라우드에서 HA는 액티브 ASA 가상 장애 때문에 백업 ASA 가상로 시스템의 자동 페일오버를 트리거하게 만들

수 있는 스테이트리스 액티브/백업 솔루션을 구현합니다. Azure Resource Manager에서 고가용성을 위한 ASA 가상 구축, 115 페이지의 내용을 참조하십시오.

- VHD(cisco.com에서 사용 가능)의 매니지드 이미지를 사용하여 맞춤형 템플릿과 함께 ASA 가상 또는 ASA 가상 고가용성 쌍을 구축합니다. Cisco에서는 Azure에 업로드하여 ASA 가상 구축 프로세스를 간소화할 수 있는 압축된 VHD(Virtual Hard Disk)를 제공합니다. 매니지드 이미지와 두 개의 JSON 파일(템플릿 파일 및 매개 변수 파일)을 사용하여, 조율된 단일 작업으로 ASA 가상을 위한 모든 리소스를 구축하고 프로비저닝할 수 있습니다. 맞춤형 템플릿을 사용하려면 VHD 및 리소스 템플릿을 사용해서 Azure에서 ASA 가상 구축, 117 페이지를 참조하십시오.

Azure Resource Manager에서 ASA 가상 구축

다음 절차는 ASA 가상에 Microsoft Azure를 설정하는 단계를 간략하게 정리한 것입니다. 자세한 Azure 설정 단계에 대한 자세한 내용은 Azure 시작하기를 참조하십시오.

Azure에서 ASA 가상를 구축할 경우 리소스, 공용 IP 주소, 경로 테이블과 같은 다양한 컨피그레이션이 자동으로 생성됩니다. 구축 후에 이 컨피그레이션을 추가로 관리할 수 있습니다. 이를테면 유휴 시간 초과 값을 낮게 설정된 기본값에서 변경할 수 있습니다.

단계 **1** ARM(Azure Resource Manager) 포털에 로그인합니다.

Azure 포털에서는 데이터 센터 위치와 상관없이 현재 계정 및 서브스크립션의 가상 요소를 보여줍니다.

단계 **2** 마켓플레이스에서 Cisco ASAv를 검색한 다음 구축하려는 ASA 가상을 클릭합니다.

단계 **3** 기본 설정을 구성합니다.

a) 가상 시스템의 이름을 입력합니다. 이 이름은 Azure 서브스크립션 내에서 고유해야 합니다.

중요 이름이 고유하지 않거나 기존 이름을 재사용하면 구축이 실패하게 됩니다.

b) 사용자 이름을 입력합니다.

c) 권한 부여 유형을 비밀번호 또는 **SSH** 공용 키 중 하나로 선택합니다.

비밀번호를 선택할 경우 비밀번호를 입력하고 확인합니다.

d) 서브스크립션 유형을 선택합니다.

e) **Resource group(**리소스 그룹**)**을 선택합니다.

리소스 그룹은 가상 네트워크의 리소스 그룹과 동일해야 합니다.

f) 위치를 선택합니다.

이 위치는 네트워크 및 리소스 그룹과 동일해야 합니다.

g) **OK(**확인**)**를 클릭합니다.

단계 **4** ASA 가상 설정을 구성합니다.

a) 가상 머신 크기를 선택합니다.

b) 스토리지 계정을 선택합니다.

기존 스토리지 계정을 사용하거나 새로 만들 수 있습니다. 스토리지 계정의 위치가 네트워크 및 가상 시스템에 대한 위치와 동일해야 합니다.

c) Name(이름) 필드에 IP 주소에 대한 레이블을 입력하여 공용 IP 주소를 요청한 다음 **OK(**확인**)**를 클릭합니다.

Azure는 기본적으로 동적 공용 IP를 생성합니다. 이는 VM이 중지하고 재시작할 때 변경될 수 있습니다. 고정 IP 주소를 선호할 경우 포털에서 public-ip를 열고 동적 주소에서 고정 주소로 변경할 수 있습니다.

d) 필요하다면 DNS 레이블을 추가합니다.

FQDN(fully qualified domain name)은 DNS 레이블 + Azure URL(<*dnslabel*>.<*location*>.*cloupapp.azure.com*)입니다.

e) 기존 가상 네트워크를 선택하거나 새로 만듭니다.

f) ASA 가상 구축 대상이 될 4개의 서브넷을 구성하고 **OK(**확인**)**를 클릭합니다.

중요 각 인터페이스가 고유한 서브넷에 연결되어야 합니다.

g) **OK(**확인**)**를 클릭합니다.

단계 **5** 컨피그레이션 요약을 본 다음 **OK(**확인**)**를 클릭합니다.

단계 **6** 이용 약관을 보고 **Create(**생성**)**를 클릭합니다.

다음에 수행할 작업

- SSH를 통해 입력 가능한 CLI 명령을 사용하여 컨피그레이션을 계속하거나 ASDM을 사용합니다. ASDM 액세스에 대한 지침은 ASDM 시작을 참조하십시오.

Azure Security Center에서 ASA 가상 구축

Microsoft Azure Security Center는 고객이 클라우드 구축에 대해 보호를 수행하고 보안 위험을 감지 및 완화하는 데 활용할 수 있게 해주는 Azure용 보안 솔루션입니다. Security Center 대시보드에서 고객은 보안 정책을 설정하고 보안 구성을 모니터링하며 보안 경고를 볼 수 있습니다.

Security Center는 잠재적인 보안 취약점을 식별하기 위해 Azure 리소스의 보안 상태를 분석합니다. 권장 사항 목록은 필요한 제어 기능을 구성하는 프로세스를 고객에게 안내합니다. 이 제어 기능에는 Azure 고객에게 ASA 가상을 방화벽 솔루션으로 배포하는 작업이 포함될 수 있습니다.

Security Center에서 통합 솔루션 역할을 하기 때문에, 몇 번만 클릭하면 ASA 가상을 신속하게 구축한 다음, 단일 대시보드에서 보안 및 상태 이벤트를 모니터링할 수 있습니다. 다음 절차는 Security Center에서 ASA 가상을 구축하는 단계를 간략하게 정리한 것입니다. 자세한 내용은 Azure Security Center를 참조하십시오.

단계 **1** Azure 포털에 로그인합니다.

Azure 포털에서는 데이터 센터 위치와 상관없이 현재 계정 및 서브스크립션의 가상 요소를 보여줍니다.

단계 **2** Microsoft Azure 메뉴에서 **Security Center**를 선택합니다.

처음으로 Security Center에 액세스하는 경우 **Welcome(**시작**)** 블레이드가 열립니다. **Yes! I want to Launch Azure Security Center(**예**, Azure Security Center**를 실행하고 싶습니다**.)**를 선택하여 **Security Center** 블레이드를 열고 데이터 수집을 활성화합니다.

단계 **3** **Security Center** 블레이드에서 **Policy(**정책**)** 타일을 선택합니다.

단계 **4** **Security Center** 블레이드에서 **Prevention policy(**방지 정책**)**를 선택합니다.

단계 **5** **Prevention policy(**방지 정책**)** 블레이드에서 보안 정책의 일부로 보려는 권장 사항을 켭니다.

a) **Next generation firewall(**차세대 방화벽**)**을 **On(**켜기**)**으로 설정합니다. 이렇게 하면 ASA 가상을 Security Center에서 권장 솔루션으로 사용할 수 있습니다.

b) 필요에 따라 다른 권장 사항을 설정합니다.

단계 **6** **Security Center** 블레이드로 돌아가 **Recommendations(**권장 사항**)** 타일로 이동합니다.

Security Center는 Azure 리소스의 보안 상태를 정기적으로 분석합니다. Security Center는 잠재적인 보안 취약점을 식별하고 **Recommendations(**권장 사항**)** 블레이드에서 권장 사항을 표시합니다.

단계 **7** **Recommendations(**권장 사항**)** 블레이드에 있는 **Add a Next Generation Firewall(**차세대 방화벽 추가**)**을 선택하여 자세한 정보를 확인하거나 문제 해결을 위한 조치를 수행합니다.

단계 **8** **Create New(**새로 생성**)** 또는 **Use existing solution(**기존 솔루션 사용**)**을 선택한 다음 구축하려는 ASA 가상을 클릭합니다.

단계 **9** 기본 설정을 구성합니다.

a) 가상 시스템의 이름을 입력합니다. 이 이름은 Azure 서브스크립션 내에서 고유해야 합니다.

중요 이름이 고유하지 않거나 기존 이름을 재사용하면 구축이 실패하게 됩니다.

b) 사용자 이름을 입력합니다.

c) 권한 부여 유형을 비밀번호 또는 SSH 키 중 하나로 선택합니다.

비밀번호를 선택할 경우 비밀번호를 입력하고 커밋합니다.

d) 서브스크립션 유형을 선택합니다.

e) 리소스 그룹을 선택합니다.

리소스 그룹은 가상 네트워크의 리소스 그룹과 동일해야 합니다.

f) 위치를 선택합니다.

이 위치는 네트워크 및 리소스 그룹과 동일해야 합니다.

g) **OK(**확인**)**를 클릭합니다.

단계 **10** ASA 가상 설정을 구성합니다.

a) 가상 머신 크기를 선택합니다.

ASA 가상은 Standard D3 및 Standard D3_v2를 지원합니다.

b) 스토리지 계정을 선택합니다.

기존 스토리지 계정을 사용하거나 새로 만들 수 있습니다. 스토리지 계정의 위치가 네트워크 및 가상 시스템에 대한 위치와 동일해야 합니다.

c) Name(이름) 필드에 IP 주소에 대한 레이블을 입력하여 공용 IP 주소를 요청한 다음 **OK(**확인**)**를 클릭합니다.

Azure는 기본적으로 동적 공용 IP를 생성합니다. 이는 VM이 중지하고 재시작할 때 변경될 수 있습니다. 고정 IP 주소를 선호할 경우 포털에서 public-ip를 열고 동적 주소에서 고정 주소로 변경할 수 있습니다.

d) 필요하다면 DNS 레이블을 추가합니다.

FQDN(fully qualified domain name)은 DNS 레이블 + Azure URL(<*dnslabel*>.<*location*>.*cloupapp.azure.com*)입니다.

e) 기존 가상 네트워크를 선택하거나 새로 만듭니다.

f) ASA 가상 구축 대상이 될 4개의 서브넷을 구성하고 **OK(**확인**)**를 클릭합니다.

중요 각 인터페이스가 고유한 서브넷에 연결되어야 합니다.

g) **OK(**확인**)**를 클릭합니다.

단계 **11** 컨피그레이션 요약을 본 다음 **OK(**확인**)**를 클릭합니다.

단계 **12** 이용 약관을 보고 **Create(**생성**)**를 클릭합니다.

다음에 수행할 작업

- SSH를 통해 입력 가능한 CLI 명령을 사용하여 컨피그레이션을 계속하거나 ASDM을 사용합니다. ASDM 액세스에 대한 지침은 ASDM 시작을 참조하십시오.
- Security Center 도움말의 권장 사항이 Azure 리소스를 보호하는 데 어떻게 도움이 되는지에 대한 자세한 내용은 Security Center에서 사용 가능한 설명서를 참조하십시오.

Azure Resource Manager에서 고가용성을 위한 ASA 가상 구축

다음 절차는 Microsoft Azure에서 HA(고가용성) ASA 가상 쌍을 설정하는 단계를 간략하게 정리한 것입니다.. 자세한 Azure 설정 단계에 대한 자세한 내용은 Azure 시작하기를 참조하십시오.

Azure에서의 ASA 가상 HA는 ASA 가상 2개를 가용성 모음에 구축하며, 리소스와 퍼블릭 IP 조수 및 라우트 테이블 같은 다양한 구성을 자동으로 생성합니다. 구축 후에 이 컨피그레이션을 추가로 관리할 수 있습니다.

단계 **1** Azure 포털에 로그인합니다.

Azure 포털에서는 데이터 센터 위치와 상관없이 현재 계정 및 서브스크립션의 가상 요소를 보여줍니다.

단계 **2** Marketplace에서 **Cisco ASAv**를 검색한 다음 **ASAv 4 NIC HA**를 클릭하여 페일오버 ASA 가상 구성을 구축합니다.

단계 **3** **Basics(**기본**)** 설정을 구성합니다.

a) ASA 가상 머신 이름의 접두사를 입력합니다. ASA 가상 이름은 'prefix'-A 및 'prefix'-B입니다.

중요 기존 접두사를 사용하면 구축이 실패하므로 주의해야 합니다.

b) 사용자 이름을 입력합니다.

두 가상 머신 모두의 관리 사용자 이름이 됩니다.

중요 사용자 이름 **admin**은 Azure에서 허용되지 않습니다.

c) 두 가상 머신의 권한 부여 유형을 비밀번호 또는 **SSH** 공개 키 중 하나로 선택합니다.

비밀번호를 선택할 경우 비밀번호를 입력하고 확인합니다.

d) 서브스크립션 유형을 선택합니다.

e) **Resource group(**리소스 그룹**)**을 선택합니다.

Create new(새로 만들기**)**를 선택하여 새 리소스 그룹을 생성하거나 **Use existing(**기존 리소스 그룹 사용**)**을 선택하여 기존 리소스 그룹을 선택합니다. 기존 리소스 그룹을 사용한다면 비어 있어야 합니다. 비어 있지 않다면 새 리소스 그룹을 생성해야 합니다.

f) 위치를 선택합니다.

이 위치는 네트워크 및 리소스 그룹과 동일해야 합니다.

g) **OK(**확인**)**를 클릭합니다.

단계 **4** **Cisco ASAv settings(Cisco ASAv** 설정**)**를 구성합니다.

a) 가상 머신 크기를 선택합니다.

b) **Managed(**관리형**)** 또는 **Unmanaged OS disk(**언매니지드 **OS** 디스크**)** 스토리지를 선택합니다.

중요 ASA HA 모드는 항상 **Managed(**관리형**)**를 사용합니다.

단계 **5** **ASAv-A** 설정을 구성합니다.

a) (선택 사항) **Create new(**새로 만들기**)**를 선택하고 Name(이름) 필드에 IP 주소에 대한 레이블을 입력하여 공용 IP 주소를 요청하고, **OK(**확인**)**를 클릭합니다. 공용 IP 주소를 사용하지 않으려면 **None(**없음**)**을 선택합니다.

참고 Azure는 기본적으로 동적 공용 IP를 생성합니다. 이는 VM이 중지하고 재시작할 때 변경될 수 있습니다. 고정 IP 주소를 선호할 경우 포털에서 public-ip를 열고 동적 주소에서 고정 주소로 변경할 수 있습니다.

b) 필요하다면 DNS 레이블을 추가합니다.

FQDN(fully qualified domain name)은 DNS 레이블 + Azure URL(<*dnslabel*>.<*location*>.*cloupapp.azure.com*)입니다.

c) ASAv-A 부팅 진단을 위한 스토리지 계정에 필요한 설정을 구성합니다.

단계 **6** **ASAv-B** 설정에 대해 이전 단계를 반복합니다.

단계 **7** 기존 가상 네트워크를 선택하거나 새로 만듭니다.

a) ASA 가상 구축 대상이 될 4개의 서브넷을 구성하고 **OK(**확인**)**를 클릭합니다.

중요 각 인터페이스가 고유한 서브넷에 연결되어야 합니다.

b) **OK(**확인**)**를 클릭합니다.

단계 **8** 컨피그레이션 **Summary(**요약**)**을 본 다음 **OK(**확인**)**를 클릭합니다.

단계 **9** 이용 약관을 보고 **Create(**생성**)**를 클릭합니다.

다음에 수행할 작업

- SSH를 통해 입력 가능한 CLI 명령을 사용하여 컨피그레이션을 계속하거나 ASDM을 사용합니다. ASDM 액세스에 대한 지침은 ASDM 시작을 참조하십시오.
- Azure에서 ASA 가상 HA 컨피그레이션에 대한 자세한 내용은 ASA 컨피그레이션 가이드의 "퍼블릭 클라우드의 고가용성을 위한 페일오버" 장을 참조하십시오.

VHD 및 리소스 템플릿을 사용해서 Azure에서 ASA 가상 구축

이제 Cisco에서 사용 가능한 압축된 VHD 이미지를 사용하여 Azure에서 고유한 맞춤형 ASA 가상 이미지를 생성할 수 있습니다. VHD 이미지를 사용하여 구축하려면 Azure 스토리지 계정에 VHD 이미지를 업로드합니다. 그런 다음, 업로드된 디스크 이미지 및 Azure Resource Manager 템플릿을 사용하여 매니지드 이미지를 생성할 수 있습니다. Azure 템플릿은 리소스 설명 및 파라미터 정의를 포함하는 JSON 파일입니다.

시작하기 전에

- ASA 가상 템플릿 구축을 위한 JSON 템플릿 및 해당 JSON 매개변수 파일이 필요합니다. 다음 GitHub 리포지토리에서 템플릿 파일을 다운로드할 수 있습니다.

  https://github.com/CiscoDevNet/cisco-asav/tree/master/deployment-templates/azure

- 템플릿 및 매개변수 파일을 작성하는 방법은 부록 - Azure 리소스 템플릿 예, 120 페이지를 참조하십시오.
- 이 절차를 수행하려면 Azure의 기존 Linux VM이 필요합니다. 압축된 VHD 이미지를 Azure에 업로드하려면 임시 Linux VM(예: Ubuntu 16.04)을 사용하는 것이 좋습니다. 이 이미지는 압축을 풀 때 약 50G의 스토리지가 필요합니다. 또한 Azure의 Linux VM에서 Azure 스토리지로의 업로드 시간이 더 빨라집니다.

  VM을 생성해야 하는 경우 다음 방법 중 하나를 사용합니다.

  - Azure CLI를 사용하여 Linux 가상 시스템 생성
  - Azure Portal에서 Linux 가상 머신 생성

- Azure 구독에서 ASA 가상를 구축하려는 위치에서 사용 가능한 스토리지 계정이 있어야 합니다.

단계 **1** https://software.cisco.com/download/home페이지에서 ASA 가상 압축 VHD 이미지를 다운로드합니다.

a) **Products(**제품**)** > **Security(**보안**)** > **Firewalls(**방화벽**)** > **ASA(Adaptive Security Appliances)** > **Adaptive Security Appliance (ASA) Software(ASA[Adaptive Security Appliance]** 소프트웨어**)**로 이동합니다.

b) **ASAv(Adaptive Security Virtual Appliance)**를 클릭합니다.

지침에 따라 다운로드합니다.

예: asav9-14-1.vhd.bz2

단계 **2** 압축된 VHD 이미지를 Azure의 Linux VM에 복사합니다.

파일을 Azure로 또는 Azure에서 아래로 이동하는 데 사용할 수 있는 여러 옵션이 있습니다. 이 예에서는 SCP 또는 보안 복사본을 보여줍니다.

```
# scp /username@remotehost.com/dir/asav9-14-1.vhd.bz2 <linux-ip>
```

단계 **3** Azure에서 Linux VM에 로그인하고 압축된 VHD 이미지를 복사한 디렉터리로 이동합니다.

단계 **4** ASA 가상 VHD 이미지의 압축을 풉니다.

파일의 압축을 풀거나 압축을 풀 때 사용할 수 있는 여러 옵션이 있습니다. 이 예에서는 Bzip2 유틸리티를 보여 주지만, 작동하는 Windows 기반 유틸리티도 있습니다.

```
# bunzip2 asav9-14-1.vhd.bz2
```

단계 **5** Azure 스토리지 계정의 컨테이너에 VHD를 업로드합니다. 기존 스토리지 계정을 사용하거나 새로 만들 수 있습니다. 스토리지 계정 이름은 소문자와 숫자만 포함할 수 있습니다.

스토리지 계정에 VHD를 업로드하는 데 사용할 수 있는 여러 옵션(AzCopy, Azure Storage Copy Blob API, Azure Storage Explorer, Azure CLI 또는 Azure Portal)이 있습니다. ASA 가상만큼 큰 파일에는 Azure Portal을 사용하지 않는 것이 좋습니다.

다음 예에서는 Azure CLI를 사용하는 구문을 보여줍니다.

```
azure storage blob upload \
       --file <unzipped vhd> \
       --account-name <azure storage account> \
       --account-key yX7txxxxxxxx1dnQ== \
       --container <container> \
       --blob <desired vhd name in azure> \
       --blobtype page
```

단계 **6** VHD에서 관리되는 이미지 생성:

a) Azure Portal에서 **Images(**이미지**)**를 선택합니다.

b) **Add(**추가**)**를 클릭하여 새 엔트리를 만듭니다.

c) 다음 정보를 제공합니다.

- **Name(**이름**)**-관리되는 이미지의 사용자 정의 이름을 입력합니다.
- **Subscription(**구독**)**-드롭 다운 목록에서 구독을 선택합니다.
- **Resource group(**리소스 그룹**)**-기존 리소스 그룹을 선택하거나 새 리소스 그룹을 생성합니다.
- **OS disk(OS** 디스크**)**-OS 유형으로 Linux를 선택합니다.
- **Storage blob(**스토리지 블롭**)**-스토리지 계정을 찾아 업로드된 VHD를 선택합니다.
- **Account type(**계정 유형**)**-드롭 다운 목록에서 표준(HDD)을 선택합니다.
- **Host caching(**호스트 캐싱**)**-드롭 다운 목록에서 Read/write(읽기/쓰기)를 선택합니다.
- **Data Disk(**데이터 디스크**)**-기본값을 그대로 둡니다. 데이터 디스크를 추가하지 마십시오.

d) **Create(**생성**)**를 클릭합니다.

Notifications(알림**)** 탭 아래에서 정상적으로 생성된 이미지 메시지를 기다립니다.

참고 관리되는 이미지가 생성되면 업로드된 VHD 및 업로드 스토리지 계정을 제거할 수 있습니다.

단계 **7** 새로 생성한 관리 이미지의 리소스 ID를 가져옵니다.

내부적으로 Azure는 모든 리소스를 리소스 ID와 연결합니다. 이 관리되는 이미지에서 새 ASA 가상 방화벽을 구축할 때는 리소스 ID가 필요합니다.

a) Azure Portal에서 **Images(**이미지**)**를 선택합니다.
b) 이전 단계에서 생성한 관리 이미지를 선택합니다.
c) 이미지 속성을 보려면 **Overview(**개요**)**를 클릭합니다.
d) 리소스 **ID**를 클립 보드에 복사합니다.

리소스 **ID**의 형식은 다음과 같습니다.

/subscriptions/<subscription-id>/resourceGroups/<resourceGroup> /providers/Microsoft.Compute/<container>/<vhdname>

단계 **8** 관리되는 이미지 및 리소스 템플릿을 사용하여 ASA 가상 방화벽을 구축합니다.

a) **New(**새로 만들기**)**를 선택하고 옵션에서 선택할 수 있을 때까지 **Template Deployment(**템플릿 구축**)**를 검색합니다.
b) **Create(**생성**)**을 선택합니다.
c) **Build your own template in the editor(**편집기에서 자체 템플릿 구축**)**를 선택합니다.

맞춤화할 수 있는 빈 템플릿이 있습니다. 템플릿을 생성하는 방법의 예는 리소스 템플릿 생성, 121 페이지를 참조하십시오.

d) 맞춤화된 JSON 템플릿 코드를 창에 붙여넣은 다음 **Save(**저장**)**를 클릭합니다.
e) 드롭 다운 목록에서 **Subscription(**구독**)**을 선택합니다.
f) 기존 **Resource group(**리소스 그룹**)**을 선택하거나 새 리소스 그룹을 생성합니다.
g) 드롭다운 목록에서 **Location(**위치**)**를 선택합니다.
h) 이전 단계의 관리 이미지 리소스 **ID**를 **Vm** 관리 이미지 **ID** 필드에 붙여 넣습니다.

단계 **9** **Custom deployment(**맞춤형 구축**)** 페이지 상단에서 **Edit parameters(**매개 변수 수정**)**를 클릭합니다. 맞춤화할 수 있는 매개변수 템플릿이 있습니다.

a) **Load file(**파일 로드**)**을 클릭하고 사용자 맞춤화된 ASA 가상 매개변수 파일을 찾습니다. 매개변수 템플릿을 생성하는 방법의 예는 매개변수 파일 생성, 130 페이지를 참조하십시오.
b) 사용자 맞춤화된 JSON 매개변수 코드를 창에 붙여 넣은 다음 **Save(**저장**)**를 클릭합니다.

단계 **10** 맞춤형 구축 세부 정보를 검토합니다. **Basics(**기본**)** 및 **Settings(**설정**)**의 정보가 리소스 **ID**를 포함하여 예상되는 구축 컨피그레이션과 일치하는지 확인합니다.

단계 **11** 약관을 검토하고 위에 명시된 약관에 동의합니다 확인란을 선택합니다.

단계 **12** 관리 이미지 및 맞춤형 템플릿을 사용하여 ASA 가상 방화벽을 구축하려면 **Purchase(**구매**)**를 클릭합니다.

템플릿 및 매개변수 파일에 충돌이 없는 경우 구축이 성공적으로 이루어지게 됩니다.

Managed Image(관리 이미지)는 동일한 구독 및 지역 내의 여러 구축에 사용할 수 있습니다.

다음에 수행할 작업

- SSH를 통해 입력 가능한 CLI 명령을 사용하여 컨피그레이션을 계속하거나 ASDM을 사용합니다. ASDM 액세스에 대한 지침은 ASDM 시작를 참조하십시오.

부록 - Azure 리소스 템플릿 예

이 섹션에서는 ASA 가상 구축에 사용할 수 있는 Azure Resource Manager 템플릿의 구조에 대해 설명합니다. Azure 리소스 템플릿은 JSON 파일입니다. 필요한 모든 리소스의 구축을 간소화하기 위해 이 예에는 JSON 파일 두 개가 포함되어 있습니다.

- **Template File(**템플릿 파일**)** - 리소스 그룹 내의 모든 구성 요소를 구축하는 기본 리소스 파일입니다.
- **Parameter File(**매개변수 파일**)** - 이 파일에는 ASA 가상을 성공적으로 구축하는 데 필요한 매개변수가 포함되어 있습니다. 여기에는 서브넷 정보, 가상 머신 계층 및 크기, ASA 가상의 사용자 이름 및 비밀번호, 스토리지 컨테이너의 이름 등의 세부 정보가 포함됩니다. Azure Stack Hub 구축 환경에 맞게 이 파일을 사용자 지정할 수 있습니다.

템플릿 파일 형식

이 섹션에서는 Azure Resource Manager 템플릿 파일 구조에 대해 설명합니다. 다음 예에서는 템플릿 파일의 축소 보기와 템플릿의 다양한 섹션을 확인할 수 있습니다.

Azure Resource Manager JSON 템플릿 파일

```
{
    "$schema":
"http://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
    "contentVersion": "",
    "parameters": { },
    "variables": { },
    "resources": [ ],
    "outputs": { }
}
```

템플릿은 ASA 가상 구축을 위한 값을 구성하는 데 사용할 수 있는 JSON 및 식으로 구성됩니다. 가장 간단한 구조에서 템플릿에는 다음 요소가 포함됩니다.

표 *17: Azure Resource Manager JSON* 템플릿 파일 요소 정의됨

요소	필수	Description
$schema	예	템플릿 언어의 버전을 설명하는 JSON 스키마 파일의 위치입니다. 위의 그림에 표시된 URL을 사용합니다.

요소	필수	**Description**
contentVersion	예	템플릿의 버전(예: 1.0.0.0)입니다. 이 요소에 임의의 값을 입력할 수 있습니다. 템플릿을 사용하여 리소스를 구축할 때 이 값을 사용하여 올바른 템플릿이 사용되고 있는지 확인할 수 있습니다.
parameters	아니요	리소스 구축을 사용자 지정하기 위해 구축을 실행할 때 제공되는 값입니다. 매개변수를 사용하면 구축할 때 값을 입력할 수 있습니다. 반드시 필요하지만 않지만, 이 값이 없으면 JSON 템플릿은 매번 동일한 매개변수를 사용하여 리소스를 구축합니다.
변수	아니요	템플릿 언어 식을 간소화하기 위해 템플릿에서 JSON 프래그먼트로 사용하는 값입니다.
리소스	예	리소스 그룹에서 구축되거나 업데이트된 리소스 유형입니다.
출력	아니요	구축 후 반환되는 값입니다.

JSON 템플릿을 사용하면 구축할 리소스 유형뿐만 아니라 관련 구성 매개변수도 선언할 수 있습니다. 다음 예에서는 새 ASA 가상을 구축하는 템플릿을 확인할 수 있습니다.

리소스 템플릿 생성

아래 예를 참고하여 텍스트 편집기를 이용해 자체 구축 템플릿을 생성할 수 있습니다.

단계 **1** 다음 예의 텍스트를 복사합니다.

예제**:**

```
{
    "$schema": "http://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
    "contentVersion": "1.0.0.0",
    "parameters": {
        "vmName": {
            "type": "string",
            "defaultValue": "ngfw",
            "metadata": {
                "description": "Name of the NGFW VM"
            }
        },
        "vmManagedImageId": {
            "type": "string",
            "defaultValue":
"/subscriptions/{subscription-id}/resourceGroups/myresourcegroup1/providers/Microsoft.Compute/images/myImage",

            "metadata": {
                "description": "The ID of the managed image used for deployment.
/subscriptions/{subscription-id}/resourceGroups/myresourcegroup1/providers/Microsoft.Compute/images/myImage"
```

```
            }
        },
        "adminUsername": {
            "type": "string",
            "defaultValue": "",
            "metadata": {
               "description": "Username for the Virtual Machine. admin, Administrator among other values
 are disallowed - see Azure docs"
            }
        },
        "adminPassword": {
            "type": "securestring",
            "defaultValue" : "",
            "metadata": {
                "description": "Password for the Virtual Machine. Passwords must be 12 to 72 chars and
have at least 3 of the following: Lowercase, uppercase, numbers, special chars"
            }
        },
        "vmStorageAccount": {
            "type": "string",
            "defaultValue": "",
            "metadata": {
                "description": "A storage account name (boot diags require a storage account). Between
3 and 24 characters. Lowercase letters and numbers only"
            }
        },
        "virtualNetworkResourceGroup": {
            "type": "string",
            "defaultValue": "",
            "metadata": {
                "description": "Name of the virtual network's Resource Group"
            }
        },
        "virtualNetworkName": {
            "type": "string",
            "defaultValue": "",
            "metadata": {
                "description": "Name of the virtual network"
            }
        },
        "mgmtSubnetName": {
            "type": "string",
            "defaultValue": "",
            "metadata": {
                "description": "The FTDv management interface will attach to this subnet"
            }
        },
        "mgmtSubnetIP": {
            "type": "string",
            "defaultValue": "",
            "metadata": {
                "description": "NGFW IP on the mgmt interface (example: 192.168.0.10)"
            }
        },
        "diagSubnetName": {
            "type": "string",
            "defaultValue": "",
            "metadata": {
                "description": "The FTDv diagnostic0/0 interface will attach to this subnet"
            }
        },
        "diagSubnetIP": {
            "type": "string",
            "defaultValue": "",
```

```
            "metadata": {
                "description": "NGFW IP on the diag interface (example: 192.168.1.10)"
            }
        },
        "gig00SubnetName": {
            "type": "string",
            "defaultValue": "",
            "metadata": {
                "description": "The FTDv Gigabit 0/0 interface will attach to this subnet"
            }
        },
        "gig00SubnetIP": {
            "type": "string",
            "defaultValue": "",
            "metadata": {
                "description": "The IP on the Gigabit 0/0 interface (example: 192.168.2.10)"
            }
        },
        "gig01SubnetName": {
            "type": "string",
            "defaultValue": "",
            "metadata": {
                "description": "The FTDv Gigabit 0/1 interface will attach to this subnet"
            }
        },
        "gig01SubnetIP": {
            "type": "string",
            "defaultValue": "",
            "metadata": {
                "description": "The IP on the Gigabit 0/1 interface (example: 192.168.3.5)"
            }
        },
        "VmSize": {
            "type": "string",
            "defaultValue": "Standard_D3_v2",
            "allowedValues": [ "Standard_D3_v2" , "Standard_D3" ],
            "metadata": {
                "description": "NGFW VM Size (Standard_D3_v2 or Standard_D3)"
            }
        }
    },
    "variables": {

        "virtualNetworkID":
"[resourceId(parameters('virtualNetworkResourceGroup'),'Microsoft.Network/virtualNetworks',
parameters('virtualNetworkName'))]",

        "vmNic0Name":"[concat(parameters('vmName'),'-nic0')]",
        "vmNic1Name":"[concat(parameters('vmName'),'-nic1')]",
        "vmNic2Name":"[concat(parameters('vmName'),'-nic2')]",
        "vmNic3Name":"[concat(parameters('vmName'),'-nic3')]",

        "vmNic0NsgName":"[concat(variables('vmNic0Name'),'-NSG')]",

        "vmMgmtPublicIPAddressName": "[concat(parameters('vmName'),'nic0-ip')]",
        "vmMgmtPublicIPAddressType": "Static",
        "vmMgmtPublicIPAddressDnsName": "[variables('vmMgmtPublicIPAddressName')]"
    },
    "resources": [
        {
            "apiVersion": "2017-03-01",
            "type": "Microsoft.Network/publicIPAddresses",
            "name": "[variables('vmMgmtPublicIPAddressName')]",
            "location": "[resourceGroup().location]",
```

```
    "properties": {
      "publicIPAllocationMethod": "[variables('vmMgmtPublicIpAddressType')]",
      "dnsSettings": {
        "domainNameLabel": "[variables('vmMgmtPublicIPAddressDnsName')]"
      }
    }
},
{
    "apiVersion": "2015-06-15",
    "type": "Microsoft.Network/networkSecurityGroups",
    "name": "[variables('vmNic0NsgName')]",
    "location": "[resourceGroup().location]",
    "properties": {
        "securityRules": [
            {
                "name": "SSH-Rule",
                "properties": {
                    "description": "Allow SSH",
                    "protocol": "Tcp",
                    "sourcePortRange": "*",
                    "destinationPortRange": "22",
                    "sourceAddressPrefix": "Internet",
                    "destinationAddressPrefix": "*",
                    "access": "Allow",
                    "priority": 100,
                    "direction": "Inbound"
                }
            },
            {
                "name": "SFtunnel-Rule",
                "properties": {
                    "description": "Allow tcp 8305",
                    "protocol": "Tcp",
                    "sourcePortRange": "*",
                    "destinationPortRange": "8305",
                    "sourceAddressPrefix": "Internet",
                    "destinationAddressPrefix": "*",
                    "access": "Allow",
                    "priority": 101,
                    "direction": "Inbound"
                }
            }
        ]
    }
},
{
    "apiVersion": "2017-03-01",
    "type": "Microsoft.Network/networkInterfaces",
    "name": "[variables('vmNic0Name')]",
    "location": "[resourceGroup().location]",
    "dependsOn": [
        "[concat('Microsoft.Network/networkSecurityGroups/',variables('vmNic0NsgName'))]",
      "[concat('Microsoft.Network/publicIPAddresses/', variables('vmMgmtPublicIPAddressName'))]"

    ],
    "properties": {
        "ipConfigurations": [
            {
                "name": "ipconfig1",
                "properties": {
                    "privateIPAllocationMethod": "Static",
                    "privateIPAddress" : "[parameters('mgmtSubnetIP')]",
                    "subnet": {
                        "id": "[concat(variables('virtualNetworkID'),'/subnets/',
```

```
parameters('mgmtSubnetName'))]"
                            },
                            "publicIPAddress":{
                                "id": "[resourceId('Microsoft.Network/publicIPAddresses/',
variables('vmMgmtPublicIPAddressName'))]"
                            }
                        }
                    }
                ],
                "networkSecurityGroup": {
                    "id": "[resourceId('Microsoft.Network/networkSecurityGroups',
variables('vmNic0NsgName'))]"
                },
                "enableIPForwarding": true
            }
        },
        {
            "apiVersion": "2017-03-01",
            "type": "Microsoft.Network/networkInterfaces",
            "name": "[variables('vmNic1Name')]",
            "location": "[resourceGroup().location]",
            "dependsOn": [
            ],
            "properties": {
                "ipConfigurations": [
                    {
                        "name": "ipconfig1",
                        "properties": {
                            "privateIPAllocationMethod": "Static",
                            "privateIPAddress" : "[parameters('diagSubnetIP')]",
                            "subnet": {
                                "id": "[concat(variables('virtualNetworkID'),'/subnets/',
parameters('diagSubnetName'))]"
                            }                        }
                    }
                ],
                "enableIPForwarding": true
            }
        },
        {
            "apiVersion": "2017-03-01",
            "type": "Microsoft.Network/networkInterfaces",
            "name": "[variables('vmNic2Name')]",
            "location": "[resourceGroup().location]",
            "dependsOn": [
            ],
            "properties": {
                "ipConfigurations": [
                    {
                        "name": "ipconfig1",
                        "properties": {
                            "privateIPAllocationMethod": "Static",
                            "privateIPAddress" : "[parameters('gig00SubnetIP')]",
                            "subnet": {
                                "id": "[concat(variables('virtualNetworkID'),'/subnets/',
parameters('gig00SubnetName'))]"
                            }                        }
                    }
                ],
                "enableIPForwarding": true
            }
        },
        {
            "apiVersion": "2017-03-01",
```

```
            "type": "Microsoft.Network/networkInterfaces",
            "name": "[variables('vmNic3Name')]",
            "location": "[resourceGroup().location]",
            "dependsOn": [
            ],
            "properties": {
                "ipConfigurations": [
                    {
                        "name": "ipconfig1",
                        "properties": {
                            "privateIPAllocationMethod": "Static",
                            "privateIPAddress" : "[parameters('gig01SubnetIP')]",
                            "subnet": {
                                "id": "[concat(variables('virtualNetworkID'),'/subnets/',
parameters('gig01SubnetName'))]"

                            }                        }
                    }
                ],
                "enableIPForwarding": true
            }
        },
        {
            "type": "Microsoft.Storage/storageAccounts",
            "name": "[concat(parameters('vmStorageAccount'))]",
            "apiVersion": "2015-06-15",
            "location": "[resourceGroup().location]",
            "properties": {
              "accountType": "Standard_LRS"
            }
        },
        {
            "apiVersion": "2017-12-01",
            "type": "Microsoft.Compute/virtualMachines",
            "name": "[parameters('vmName')]",
            "location": "[resourceGroup().location]",
            "dependsOn": [
                "[concat('Microsoft.Storage/storageAccounts/', parameters('vmStorageAccount'))]",
                "[concat('Microsoft.Network/networkInterfaces/',variables('vmNic0Name'))]",
                "[concat('Microsoft.Network/networkInterfaces/',variables('vmNic1Name'))]",
                "[concat('Microsoft.Network/networkInterfaces/',variables('vmNic2Name'))]",
                "[concat('Microsoft.Network/networkInterfaces/',variables('vmNic3Name'))]"
            ],
            "properties": {
                "hardwareProfile": {
                    "vmSize": "[parameters('vmSize')]"
                },
                "osProfile": {
                    "computername": "[parameters('vmName')]",
                    "adminUsername": "[parameters('AdminUsername')]",
                    "adminPassword": "[parameters('AdminPassword')]"
                },
                "storageProfile": {
                    "imageReference": {
                        "id": "[parameters('vmManagedImageId')]"
                    },
                    "osDisk": {
                        "osType": "Linux",
                        "caching": "ReadWrite",
                        "createOption": "FromImage"
                    }
                },
                "networkProfile": {
                    "networkInterfaces": [
```

```
                        {
                            "properties": {
                                "primary": true
                            },
                            "id": "[resourceId('Microsoft.Network/networkInterfaces',
variables('vmNic0Name'))]"
                        },
                        {
                            "properties": {
                                "primary": false
                            },
                            "id": "[resourceId('Microsoft.Network/networkInterfaces',
variables('vmNic1Name'))]"
                        },
                        {
                            "properties": {
                                "primary": false
                            },
                            "id": "[resourceId('Microsoft.Network/networkInterfaces',
variables('vmNic2Name'))]"
                        },
                        {
                            "properties": {
                                "primary": false
                            },
                            "id": "[resourceId('Microsoft.Network/networkInterfaces',
variables('vmNic3Name'))]"
                        }
                    ]
                },
                "diagnosticsProfile": {
                    "bootDiagnostics": {
                        "enabled": true,
                        "storageUri":
"[concat('http://',parameters('vmStorageAccount'),'.blob.core.windows.net')]"
                    }
                }
            }
        }
    ],
    "outputs": {  }
}
```

단계 **2** 파일을 JSON 파일로 로컬에 저장합니다(예: **azureDeploy.json**).

단계 **3** 파일을 편집하여 구축 파라미터에 맞게 템플릿을 생성합니다.

단계 **4** 이 템플릿을 사용하여 VHD 및 리소스 템플릿을 사용해서 Azure에서 ASA 가상 구축, 117 페이지에 설명된 대로 ASA 가상을 구축합니다.

매개변수 파일 형식

새 구축을 시작하면 리소스 템플릿에 매개변수가 정의됩니다. 구축을 시작하려면 매개변수를 입력해야 합니다. 리소스 템플릿에 정의한 매개변수를 수동으로 입력하거나, 템플릿 매개변수 JSON 파일에 매개변수를 추가할 수 있습니다.

매개변수 파일에는 매개변수 파일 생성, 130 페이지의 매개변수 예에 나와 있는 각 매개변수에 대한 값이 포함되어 있습니다. 이러한 값은 구축 중에 자동으로 템플릿에 전달됩니다. 다양한 구축 시나리오를 위한 여러 매개변수 파일을 생성할 수 있습니다.

이 예에 나오는 ASA 가상 템플릿의 경우 매개변수 파일에 다음 매개변수가 정의되어 있어야 합니다.

표 ***18: ASA*** 가상 매개변수 정의

필드	**Description**	예
vmName	ASA 가상 머신의 Azure에서 의 이름입니다.	cisco-asav
vmManagedImageId	구축에 사용하는 관리형 이미지의 ID입니다. 내부적으로 Azure는 모든 리소스를 리소스 ID와 연결합니다.	/subscriptions/73d2537e-ca44-46aa-beb2-74ff1dd61b41/resourceGroups/ewManagedImages-rg/providers/Microsoft.Compute/images/ASAv910-Managed-I mage
adminUsername	ASA 가상에 로그인하기 위한 사용자 이름입니다. 예약된 이름인 'admin'은 사용할 수 없습니다.	jdoe
adminPassword	관리자 비밀번호입니다. 길이는 12~72자여야 하며 소문자 1개, 대문자 1개, 숫자 1개, 특수 문자 1개 중에서 3개를 포함해야 합니다.	Pw0987654321
vmStorageAccount	Azure 스토리지 계정입니다. 기존 스토리지 계정을 사용하거나 새로 만들 수 있습니다. 스토리지 계정 이름은 3~24자이고 소문자와 숫자만 포함할 수 있습니다.	ciscoasavstorage
virtualNetworkResourceGroup	가상 네트워크 리소스 그룹의 이름입니다. ASA 가상는 항상 새 리소스 그룹에 구축됩니다.	ew-west8-rg
virtualNetworkName	가상 네트워크의 이름입니다.	ew-west8-vnet
mgmtSubnetName	관리 인터페이스가 이 서브넷에 연결됩니다. 첫 번째 서브넷인 Nic0에 매핑됩니다. 기존 네트워크에 조인하는 경우 기존 서브넷 이름과 일치해야 합니다.	mgmt
mgmtSubnetIP	관리 인터페이스 IP 주소입니다.	10.8.0.55

필드	Description	예
gig00SubnetName	GigabitEthernet 0/0 인터페이스가 이 서브넷에 연결됩니다. 두 번째 서브넷인 Nic1에 매핑됩니다. 기존 네트워크에 조인하는 경우 기존 서브넷 이름과 일치해야 합니다.	내부
gig00SubnetIP	GigabitEthernet 0/0 인터페이스 IP 주소입니다. ASA 가상의 첫 번째 데이터 인터페이스에 사용합니다.	10.8.2.55
gig01SubnetName	GigabitEthernet 0/1 인터페이스가 이 서브넷에 연결됩니다. 세 번째 서브넷인 Nic2에 매핑됩니다. 기존 네트워크에 조인하는 경우 기존 서브넷 이름과 일치해야 합니다.	외부
gig01SubnetIP	GigabitEthernet 0/1 인터페이스 IP 주소입니다. ASA 가상의 두 번째 데이터 인터페이스에 사용합니다.	10.8.3.55
gig02SubnetName	GigabitEthernet 0/2 인터페이스가 이 서브넷에 연결됩니다. 네 번째 서브넷인 Nic3에 매핑됩니다. 기존 네트워크에 조인하는 경우 기존 서브넷 이름과 일치해야 합니다.	dmz
gig02SubnetIP	GigabitEthernet 0/2 인터페이스 IP 주소입니다. ASA 가상의 세 번째 데이터 인터페이스에 사용합니다.	10.8.4.55
vmSize	ASA 가상 VM에 사용할 VM 크기입니다. Standard_D3_V2 및 Standard_D3가 지원됩니다. Standard_D3_V2가 기본값입니다.	Standard_D3_V2 또는 Standard_D3

매개변수 파일 생성

아래 예를 참고하여 텍스트 편집기를 사용하여 자체 매개변수 파일을 생성할 수 있습니다.

참고 다음 예는 IPV4에만 해당됩니다.

단계 **1** 다음 예의 텍스트를 복사합니다.

예제:

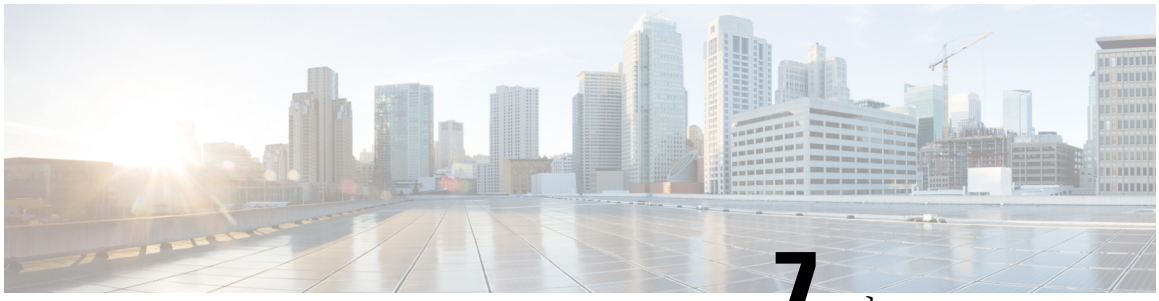
```
{
  "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentParameters.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "vmName": {
      "value": "cisco-asav1"
    },
    "vmManagedImageId": {
      "value":
"/subscriptions/33d2517e-ca88-46aa-beb2-74ff1dd61b41/resourceGroups/ewManagedImages-rg/providers/Microsoft.Compute/images/ASAv-9.10.1-81-Managed-Image"

    },
    "adminUsername": {
      "value": "jdoe"
    },
    "adminPassword": {
      "value": "Pw0987654321"
    },
    "vmStorageAccount": {
      "value": "ciscoasavstorage"
    },
    "virtualNetworkResourceGroup": {
      "value": "ew-west8-rg"
    },
    "virtualNetworkName": {
      "value": "ew-west8-vn"
    },
    "mgmtSubnetName": {
      "value": "mgmt"
    },
    "mgmtSubnetIP": {
      "value": "10.8.3.77"
    },
    "gig00SubnetName": {
      "value": "inside"
    },
    "gig00SubnetIP": {
      "value": "10.8.2.77"
    },
    "gig01SubnetName": {
      "value": "outside"
    },
    "gig01SubnetIP": {
      "value": "10.8.1.77"
    },
    "gig02SubnetName": {
      "value": "dmz"
```

```
    },
    "gig02SubnetIP": {
      "value": "10.8.0.77"
    },
    "VmSize": {
      "value": "Standard_D3_v2"
    }
  }
}
```

단계 **2** 파일을 JSON 파일로 로컬에 저장합니다(예: **azureParameters.json**).

단계 **3** 파일을 편집하여 구축 파라미터에 맞게 템플릿을 생성합니다.

단계 **4** 이 매개변수 템플릿을 사용하여 VHD 및 리소스 템플릿을 사용해서 Azure에서 ASA 가상 구축, 117 페이지에 설명된 대로 ASA 가상을 구축합니다.



7 장

Microsoft Azure에서 ASA 가상 Auto Scale 솔루션 구축

Azure의 ASA 가상용 Auto Scale 솔루션

Auto Scale 솔루션

ASA 가상 Auto Scale for Azure는 Azure에서 제공하는 서버리스 인프라(논리 앱, Azure 기능, 로드 밸런서, 보안 그룹, 가상 시스템 확장 집합 등)를 사용하며 완벽환 서버리스 방식으로 구현됩니다.

Azure용 ASA 가상 Auto Scale 구현의 몇 가지 주요 기능은 다음과 같습니다.

- ARM(Azure Resource Manager) 템플릿 기반 구축
- CPU 기반의 메트릭 확장을 지원합니다.

참고 자세한 내용은 Auto Scale 논리, 158 페이지를 참조하십시오.

- ASA 가상 구축 및 다중 가용성 영역 지원

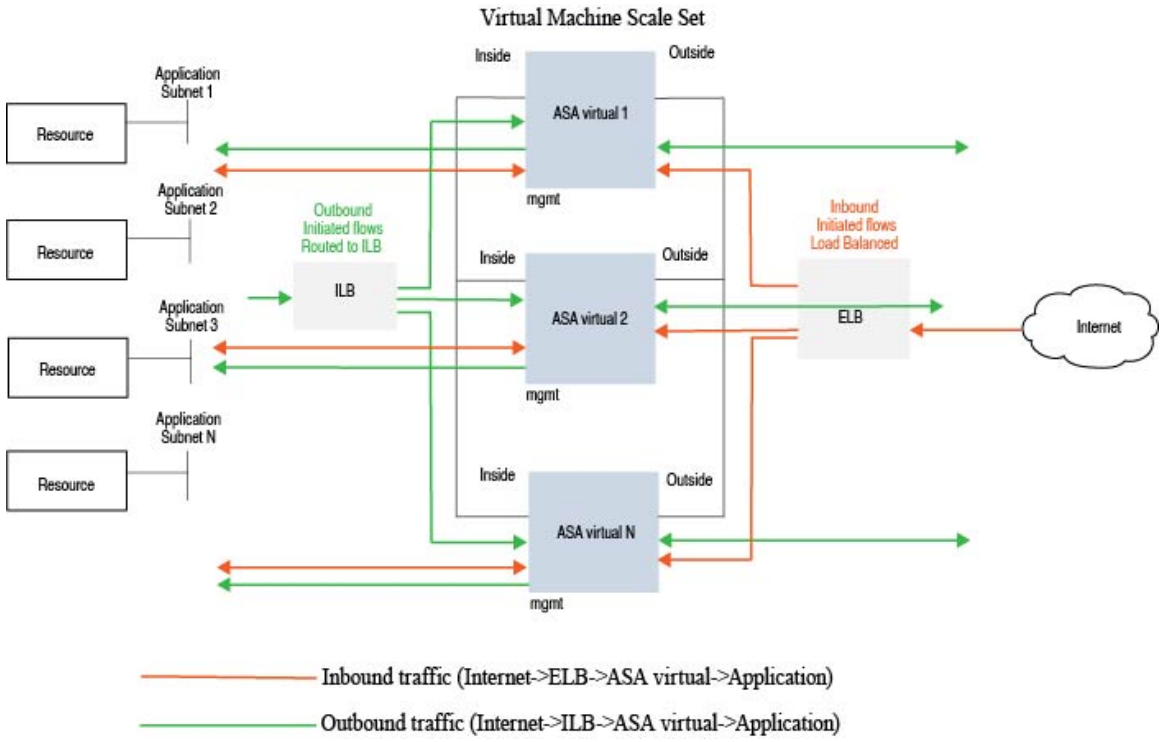
- 확장된 ASA 가상 인스턴스에 자동으로 적용되는 완전히 자동화된 컨피그레이션.
- 로드 밸런서 및 다중 가용성 영역 지원
- Auto Scale 기능 활성화 및 비활성화 지원
- Cisco에서는 구축을 쉽게 수행할 수 있도록 Azure용 Auto Scale 구축 패키지를 제공합니다.

사용 사례

Azure용 ASA 가상 Auto Scale은 Azure 내부 로드 밸런서(ILB)와 Azure 외부 로드 밸런서(ELB) 사이에 ASA 가상 확장 집합을 배치하는 자동화된 수평 확장 솔루션입니다.

- ELB는 확장 집합에서 인터넷에서 ASA 가상 인스턴스로 트래픽을 분산합니다. 그러면 방화벽이 애플리케이션에 트래픽을 전달합니다.
- ILB는 애플리케이션의 아웃 바운드 인터넷 트래픽을 확장 집합의 ASA 가상 인스턴스로 분산합니다. 그러면 방화벽이 트래픽을 인터넷으로 전달합니다.
- 네트워크 패킷은 단일 연결에서 내부 및 외부 로드 밸런서를 모두 통과하지 않습니다.
- 확장 집합의 ASA 가상 인스턴스 수는 로드 조건에 따라 자동으로 조정 및 구성됩니다.

그림 ***16:*** 사용 사례



범위

이 문서에서는 ASA 가상 Auto Scale for Azure 솔루션의 서버리스 구성 요소를 구축하는 자세한 절차를 설명합니다.

중요

- 구축을 시작하기 전에 전체 문서를 읽어보십시오.
- 구축을 시작하기 전에 전제 조건이 충족되었는지 확인합니다.
- 여기에 설명된 대로 단계 및 실행 순서를 따라야 합니다.

구축 패키지 다운로드

ASA 가상 Auto Scale for Azure 솔루션은 Azure에서 제공하는 서버리스 인프라(논리 앱, Azure Functions, 로드 밸런서, 가상 시스템 확장 집합 등)를 활용하는 ARM(Azure Resource Manager) 템플릿 기반 구축입니다.

Azure용 ASA 가상 Auto Scale 솔루션을 시작하는 데 필요한 파일을 다운로드합니다. 사용자 버전에 맞는 구축 스크립트 및 템플릿은 GitHub 리포지토리에서 제공됩니다.

주의

Cisco에서 제공하는 자동 확장용 구축 스크립트 및 템플릿은 오픈 소스 예시로 제공되며 일반적인 Cisco TAC 지원 범위에서는 다루지 않습니다. GitHub에서 정기적으로 업데이트 및 ReadMe 지침을 확인하십시오.

ASM_Function.zip 패키지를 구축하는 방법에 대한 지침은 소스 코드로 Azure 기능 빌드, 160 페이지를 참고하십시오.

Auto Scale 솔루션 구성 요소

다음 구성 요소는 Azure용 ASA 가상 Auto Scale 솔루션을 구성합니다.

Azure Functions(Function 앱)

Function 앱은 Azure 함수의 집합입니다. 기본 기능은 다음과 같습니다.

- Azure 메트릭을 주기적으로 통신/프로브합니다.
- ASA 가상 로드를 모니터링하고 축소/확장(Scale In/Scale Out 작업을 트리거합니다.

이들 함수는 압축된 Zip 패키지 형식으로 제공됩니다(Azure Function 앱 패키지 빌드, 139 페이지 참조). 함수는 특정 작업을 수행하기 위해 가능한 한 개별적으로 유지되며, 개선 사항 및 새로운 릴리스 지원을 위해 필요에 따라 업그레이드할 수 있습니다.

오케스트레이터(Logic 앱)

Auto Scale Logic App은 하나의 워크플로우, 즉 시퀀스 단계 모음입니다. Azure 함수는 독립적인 엔터티이므로 서로 통신할 수 없습니다. 이 오케스트레이터는 이러한 함수의 실행을 시퀀싱하고 함수간 정보를 교환합니다.

- Logic App은 Auto Scale Azure 함수 간에 정보를 오케스트레이션하고 전달하는 데 사용됩니다.
- 각 단계는 Auto Scale Azure 함수 또는 기본 제공 표준 논리를 나타냅니다.
- Logic 앱은 JSON 파일로 제공됩니다.
- Logic 앱은 GUI 또는 JSON 파일을 통해 맞춤화할 수 있습니다.

VMSS(Virtual Machine Scale Set)

VMSS는 ASA 가상 디바이스와 같은 균일한 가상 시스템의 모음입니다.

- VMSS는 해당 집합에 동일한 새 VM을 추가할 수 있습니다.
- VMSS에 추가된 새 VM은 로드 밸런서, 보안 그룹 및 네트워크 인터페이스에 자동으로 연결됩니다.
- VMSS에는 Azure ASA 가상용으로 사용하지 않도록 설정된 Auto Scale 기능이 내장되어 있습니다.
- VMSS에서 ASA 가상 인스턴스를 수동으로 추가하거나 삭제해서는 안 됩니다.

ARM(Azure Resource Manager) 템플릿

ARM 템플릿은 Azure용 ASA 가상 Auto Scale 솔루션에 필요한 리소스를 구축하는 데 사용됩니다.

ARM 템플릿은 다음을 포함하여 Auto Scale Manager 구성 요소에 대한 입력을 제공합니다.

- Azure Function 앱
- Azure Logic 앱
- VMSS(Virtual Machine Scale Set)
- 내부/외부 로드 밸런서
- 구축에 필요한 보안 그룹 및 기타 기타 구성 요소

중요 ARM 템플릿은 사용자 입력 검증과 관련하여 제한 사항이 있으므로 구축 중에 입력을 검증해야 합니다.

Auto Scale 솔루션 사전 요건

Azure 리소스

리소스 그룹

이 솔루션의 모든 구성 요소를 구축하려면 기존 또는 새로 생성 된 리소스 그룹이 필요합니다.

참고 나중에 사용할 수 있도록 리소스 그룹 이름, 리소스 그룹이 생성된 지역 및 Azure 구독 ID를 기록합니다.

네트워킹

가상 네트워크가 사용 가능하거나 생성되었는지 확인합니다. 를 이용한 Auto Scale 구축은 네트워킹 리소스를 생성, 변경 또는 관리하지 않습니다.

ASA 가상에는 3개 네트워크 인터페이스가 필요하므로 가상 네트워크에는 3개 서브넷이 필요합니다.

1. 관리 트래픽
2. 내부 트래픽
3. 외부 트래픽

서브넷이 연결된 네트워크 보안 그룹에서 다음 포트를 열어야 합니다.

- SSH(TCP/22)

  로드 밸런서와 ASA 가상 사이의 상태 프로브에 필요합니다.

  서버리스 함수와 ASA 가상 간의 통신에 필요합니다.

- 애플리케이션별 프로토콜/포트

  모든 사용자 애플리케이션(예: TCP/80)에 필요합니다.

참고 가상 네트워크 이름, 가상 네트워크 CIDR, 3개 서브넷의 이름, 외부 및 내부 서브넷의 게이트웨이 IP 주소를 기록합니다.

ASA 컨피그레이션 파일 준비

ASA 가상 구성 파일을 준비하고 ASA 가상 인스턴스에서 액세스할 수 있는 http/https 서버에 저장합니다. 표준 ASA 컨피그레이션 파일 형식을 사용합니다. 확장된 ASA 가상에서 이 파일을 다운로드하고 관련 컨피그레이션을 업데이트합니다.

ASA 컨피그레이션 파일에는 최소한 다음이 포함되어야 합니다.

- 모든 인터페이스에 DHCP IP 할당을 설정합니다.
- GigabitEthernet0/1은 '내부' 인터페이스여야 합니다.
- GigabitEthernet0/0은 '외부' 인터페이스여야 합니다.
- 게이트웨이를 내부 및 외부 인터페이스로 설정합니다.
- (상태 프로브를 위해) Azure 유틸리티 IP의 내부 및 외부 인터페이스에서 SSH를 활성화합니다.
- 외부에서 내부 인터페이스로 트래픽을 전달하도록 NAT 구성을 생성합니다.
- 원하는 트래픽을 허용하는 액세스 정책을 생성합니다.
- 컨피그레이션에 라이선스를 부여합니다. PAYG 청구는 지원되지 않습니다.

참고 관리 인터페이스를 별도로 구성하지 않아도 됩니다.

다음은 .

```
ASA Version 9.13(1)
!
interface GigabitEthernet0/1
nameif inside
security-level 100
ip address dhcp setroute
!
interface GigabitEthernet0/0
nameif outside
security-level 0
ip address dhcp setroute
!
route outside 0.0.0.0 0.0.0.0 10.12.3.1 2
!
route inside 0.0.0.0 0.0.0.0 10.12.2.1 3
!
ssh 168.63.129.0 255.255.255.0 outside
!
ssh 168.63.129.0 255.255.255.0 inside
!
object network webserver
host 10.12.2.5
object service myport
service tcp source range 1 65535 destination range 1 65535
access-list outowebaccess extended permit object myport any any log disable
access-group outowebaccess in interface outside
object service app
```

```
service tcp source eq www
nat (inside,outside) source static webserver interface destination static interface any
service app app
object network obj-any
subnet 0.0.0.0 0.0.0.0
nat (inside,outside) source dynamic obj-any interface destination static obj-any obj-any
configure terminal
dns domain-lookup management
policy-map global_policy
class inspection_default
inspect icmp
call-home
profile License
destination transport-method http
destination address http https://tools.cisco.com/its/service/oddce/services/DDCEService
license smart
feature tier standard
throughput level 2G
license smart register idtoken <TOKEN>
: end
```

Azure Function 앱 패키지 빌드

ASA 가상 Auto Scale 솔루션은 아카이브 파일인 *ASM_Function.zip* 빌드가 필요하며, 이 파일은 압축된 ZIP 패키지로 개별 Azure 기능을 제공합니다.

ASM_Function.zip 패키지를 구축하는 방법에 대한 지침은 소스 코드로 Azure 기능 빌드, 160 페이지를 참고하십시오.

이러한 기능은 특정 작업을 수행하기 위해 가능한 한 개별적이며, 개선 사항 및 새로운 릴리스 지원을 위해 필요에 따라 업그레이드할 수 있습니다.

입력 매개변수

다음 표에서는 템플릿 매개 변수를 정의하고 일 예를 제공합니다. 이러한 값을 결정하고 나면 Azure 구독에 ARM 템플릿을 구축할 때 이러한 매개 변수를 사용하여 ASA 가상 디바이스를 생성할 수 있습니다. Auto Scale ARM 템플릿 구축, 143 페이지의 내용을 참조하십시오.

표 ***19:*** 템플릿 매개변수

매개 변수 이름	허용되는 값 / 유형	설명	리소스 생성 유형
resourceNamePrefix	문자열*(3~10자)	모든 리소스는 이 접두사를 포함하는 이름으로 생성됩니다. 참고: 소문자만 사용하십시오. 예: asav	New
virtualNetworkRg	문자열	가상 네트워크 리소스 그룹 이름입니다. 예: cisco-virtualnet-rg	기존

매개 변수 이름	허용되는 값 / 유형	설명	리소스 생성 유형
virtualNetworkName	문자열	가상 네트워크 이름(이미 생성됨) 예: cisco-virtualnet	기존
mgmtSubnet	문자열	관리 서브넷 이름(이미 생성됨) 예: cisco-mgmt-subnet	기존
insideSubnet	문자열	내부 서브넷 이름(이미 생성됨) 예: cisco-inside-subnet	기존
internalLbIp	문자열	내부 서브넷(이미 생성됨)의 내부 로드 밸런서 IP 주소입니다. 예: 1.2.3.4.	기존
outsideSubnet	문자열	외부 서브넷 이름(이미 생성됨) 예: cisco-outside-subnet	기존
softwareVersion	문자열	ASA 가상 버전(구축 중 드롭 다운에서 선택) 기본값: 914.1.0 허용됨: 914.1.0, 913.1.0	기존
vmSize	문자열	ASA 가상 인스턴스의 크기(구축 중 드롭 다운에서 선택).	해당 없음
asaAdminUserName	문자열*	ASA 가상 '관리자' 사용자의 사용자 이름입니다. 비밀번호는 12 ~ 72자여야 하며 소문자, 대문자, 숫자 및 특수 문자를 포함해야 합니다. 같은 문자를 세 번 이상 반복해서 사용할 수 없습니다. 이는 관리자가 될 수 없습니다. Azure for VM administrator user name guidelines를 참조하십시오. 참고 템플릿에는 이에 대한 규정 준수 확인이 없습니다.	New

매개 변수 이름	허용되는 값 / 유형	설명	리소스 생성 유형
asaAdminUserPassword	문자열*	ASA 가상 관리자 사용자의 비밀번호입니다. 비밀번호는 12 ~ 72자여야 하며 소문자, 대문자, 숫자 및 특수 문자를 포함해야 합니다. 같은 문자를 세 번 이상 반복해서 사용할 수 없습니다. 참고 템플릿에는 이에 대한 규정 준수 확인이 없습니다.	New
scalingPolicy	POLICY-1 / POLICY-2	**POLICY-1**: 어떤 ASA 가상 평균 로드가 구성된 기간 동안 확장 임계값을 초과하면 확장(Scale-Out)이 트리거됩니다. **POLICY-2**: 자동 확장 그룹 내의 모든 ASA 가상 평균로드가 구성된 기간 동안 확장 임계값을 초과하면 확장(Scale-Out)이 트리거됩니다. 두 경우 모두 축소(Scale-In) 논리는 동일하게 유지됩니다. 모든 ASA 가상 디바이스의 평균로드가 구성된 기간 동안 축소 임계값 미만이 되면 축소가 트리거됩니다.	해당 없음
scalingMetricsList	문자열	스케일링 결정을 내리는 데 사용되는 메트릭입니다. 허용됨: CPU 기본값: CPU	해당 없음
scaleInThreshold	문자열	에 대한 축소 임계값입니다. 기본값: 10 ASA 가상 메트릭이 이 값보다 작으면 축소(Scale-In)가 트리거됩니다. Auto Scale 논리, 158 페이지의 내용을 참조하십시오.	해당 없음

매개 변수 이름	허용되는 값 / 유형	설명	리소스 생성 유형
scaleOutThreshold	문자열	의 확장 임계값입니다. 기본값: 80 메트릭이이 값을 초과하면 스케일 아웃이 트리거됩니다.ASA 가상 'scaleOutThreshold'는 항상 'scaleInThreshold' 보다 커야합니다. Auto Scale 논리, 158 페이지의 내용을 참조하십시오.	해당 없음
minAsaCount	정수	지정된 시간에 설정된 확장 집합에서 사용 가능한 최소 ASA 가상 인스턴스. 예: 2	해당 없음
maxAsaCount	정수	확장 집합에서 허용되는 최대 ASA 가상 인스턴스 수입니다. 예: 10 참고 Auto Scale 논리는이 변수의 범위를 확인하지 않으므로 신중하게 입력하십시오.	해당 없음
metricsAverageDuration	정수	드롭다운에서 선택 이 숫자는 메트릭이 평균화되는 시간(분)을 나타냅니다. 이 변수의 값이 5(즉, 5)인 경우, Auto Scale Manager가 예약되면 메트릭의 지난 5분 평균을 확인하고 이를 기반으로 하여 확장 결정을 내립니다. 참고 Azure 제한으로 인해 숫자 1, 5, 15, 30만 유효합니다.	해당 없음

매개 변수 이름	허용되는 값 / 유형	설명	리소스 생성 유형
initDeploymentMode	일괄(BULK) / 단계별(STEP)	기본적으로 첫 번째 구축 또는 확장 집합에 ASA 가상 인스턴스가 포함되지 않은 경우에 적용됩니다. 일괄(BULK): Aucto Scale Manager가 한 번에 ' minAsaCount'개의 ASA 가상 인스턴스 수를 동시에 구축하려고 시도합니다. 단계별(STEP): Auto Scale Manager는 예약된 간격마다 하나씩 ' minAsaCount'개의 ASA 가상 디바이스를 구축합니다.	
configurationFile	문자열	ASA 가상 컨피그레이션 파일의 전체 경로. 예: https : //myserver/asavconfig/asaconfig.txt	해당 없음
* Azure에는 새 리소스의 명명 규칙에 제한 사항이 있습니다. 제한 사항을 검토하거나 간단히 모두 소문자를 사용하십시오. 공백이나 특수 문자는 사용하지 마십시오.			

Auto Scale 구축

Auto Scale ARM 템플릿 구축

ASA 가상 Auto Scale for Azure에서 요구하는 리소스를 ARM 템플릿을 사용하여 구축합니다. 지정된 리소스 그룹 내에서 ARM 템플릿 구축은 다음을 생성합니다.

- VMSS(Virtual Machine Scale Set)
- 외부 로드 밸런서
- 내부 로드 밸런서
- Azure Function 앱
- Logic 앱
- 보안 그룹 (데이터 및 관리 인터페이스용)

시작하기 전에

- GitHub 리포지토리(https://github.com/CiscoDevNet/cisco-asav)에서 ARM 템플릿 *azure_ asav _autoscale.json*을 다운로드합니다.

단계 **1** 여러 Azure 영역에서 ASA 가상 인스턴스를 구축해야 하는 경우 구축 영역에서 사용 가능한 영역을 기준으로 하여 ARM 템플릿을 편집합니다.

예제:

```
"zones": [
    "1",
    "2",
    "3"
],
```

이 예에서는 3개의 영역이 있는"Central US" 지역을 보여줍니다.

단계 **2** 외부 로드 밸런서에 필요한 트래픽 규칙을 수정합니다. 이 'json' 어레이를 확장하여 원하는 수의 규칙을 추가할 수 있습니다.

예제:

```
{
 "type": "Microsoft.Network/loadBalancers",
 "name": "[variables('elbName')]",
 "location": "[resourceGroup().location]",
 "apiVersion": "2018-06-01",
 "sku": {
   "name": "Standard"
 },
 "dependsOn": [
   "[concat('Microsoft.Network/publicIPAddresses/', variables('elbPublicIpName'))]"

 ],
 "properties": {
   "frontendIPConfigurations": [
     {
       "name": "LoadBalancerFrontEnd",
         "properties": {
           "publicIPAddress": {
             "id": "[resourceId('Microsoft.Network/publicIPAddresses/',
variables('elbPublicIpName'))]"
           }
         }
     }
   ],
   "backendAddressPools": [
     {
       "name": "backendPool"
     }
   ],
   "loadBalancingRules": [
     {
       "properties": {
         "frontendIPConfiguration": {
           "Id": "[concat(resourceId('Microsoft.Network/loadBalancers', variables('elbName')),
 '/frontendIpConfigurations/LoadBalancerFrontend')]"
```

```
                },
                "backendAddressPool": {
                  "Id": "[concat(resourceId('Microsoft.Network/loadBalancers', variables('elbName')),
'/backendAddressPools/BackendPool')]"
                },
                "probe": {
                  "Id": "[concat(resourceId('Microsoft.Network/loadBalancers', variables('elbName')),
'/probes/lbprobe')]"
                },
                "protocol": "TCP",
                "frontendPort": "80",
                "backendPort": "80",
                "idleTimeoutInMinutes": "[variables('idleTimeoutInMinutes')]"
              },
              "Name": "lbrule"
            }
          ],
```

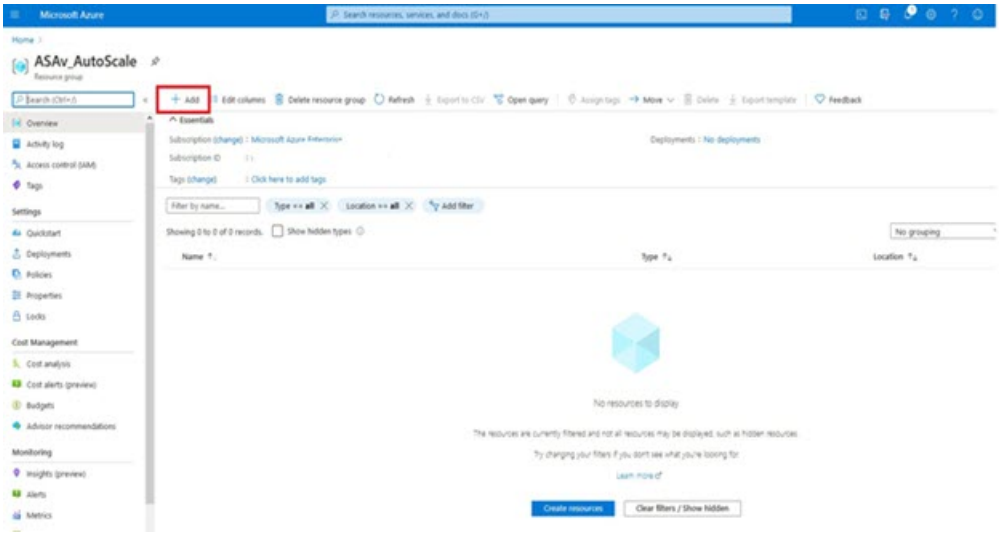
참고 이 파일을 편집하지 않으려는 경우 Azure 포털 구축후(post-deployment)에서 이를 편집할 수도 있습니다.

단계 **3** Microsoft 계정 사용자 이름 및 비밀번호를 사용하여 Microsoft Azure 포털에 로그인합니다.

단계 **4** 서비스 메뉴에서 **Resource groups(**리소스 그룹**)**를 클릭하여 리소스 그룹 블레이드에 액세스합니다. 블레이드에 나열된 구독의 모든 리소스 그룹이 표시됩니다.

새 리소스 그룹을 생성하거나 기존의 빈 리소스 그룹을 선택합니다(예:*ASA 가상 _AutoScale*).

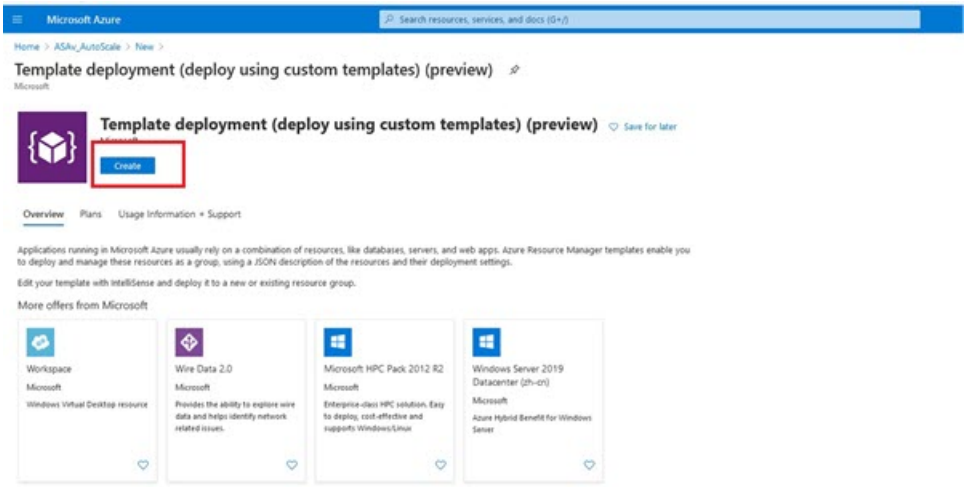
그림 ***17: Azure Portal***



단계 **5** **Create a resource(**리소스 생성**)(+)**를 클릭하여 템플릿 구축을 위한 새 리소스를 생성합니다. Create Resource Group(리소스 그룹 생성) 블레이드가 나타납니다.

단계 **6** **Search the Marketplace(**마켓플레이스 검색**)**에서 **Template deployment(**구축 **(**맞춤형 템플릿 사용**))**를 입력한 다음 **Enter** 키를 누릅니다.

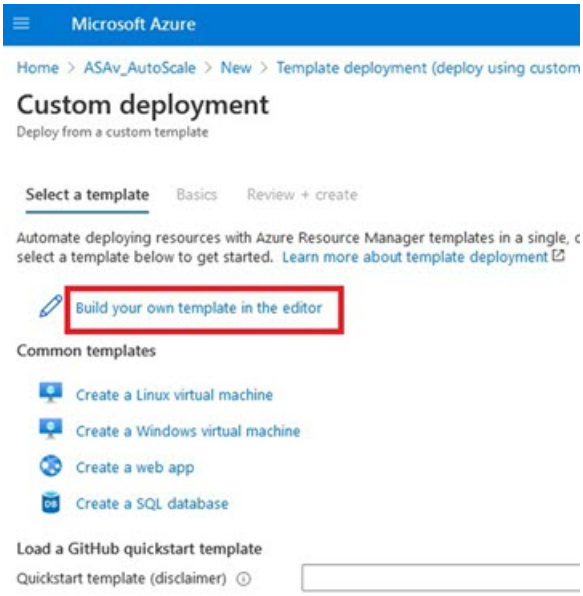
그림 *18:* 맞춤형 템플릿 구축



단계 **7** **Create(**생성**)**를 클릭합니다.

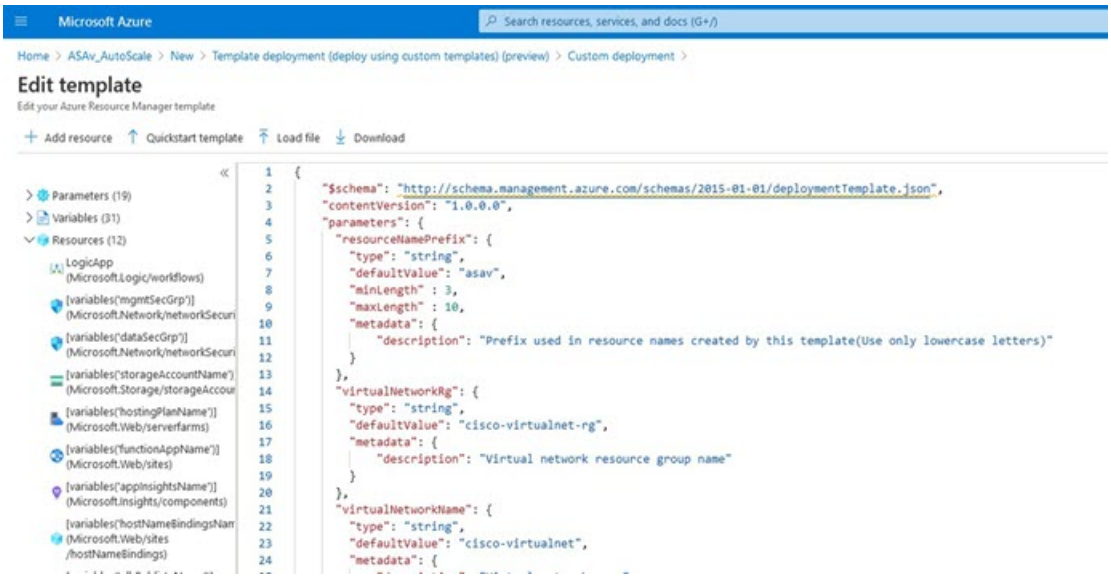
단계 **8** 템플릿을 생성하기 위한 몇 가지 옵션이 있습니다. **Build your own template in editor(**편집기에서 자체 템플릿 구축**)**를 선택합니다.

그림 *19:* 자체 템플릿 만들기



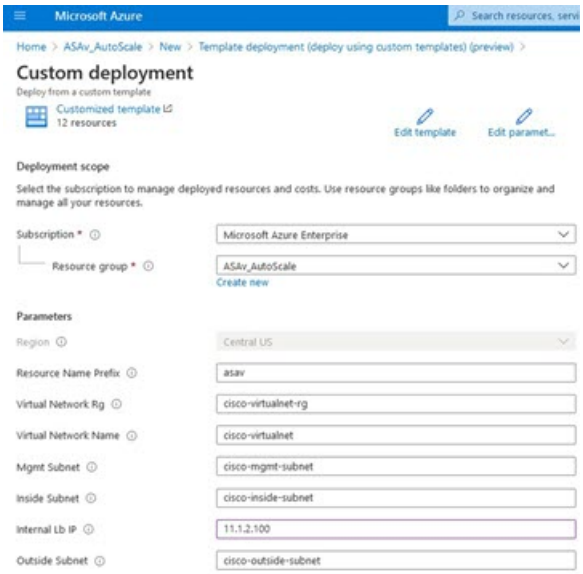
단계 **9** **Edit template(**템플릿 편집**)** 창에서 모든 기본 콘텐츠를 삭제하고 업데이트된 *azure_ asav _autoscale.json*에서 콘텐츠를 복사하고 **Save(**저장**)**를 클릭합니다.

그림 ***20:*** 템플릿 수정



단계 **10** 다음 섹션에서 모든 매개변수를 입력합니다. 각 매개변수에 대한 자세한 내용은 입력 매개변수, 139 페이지를 참조한 다음 **Purchase(**구매**)**를 클릭하십시오.

그림 ***21: ARM*** 템플릿 매개변수

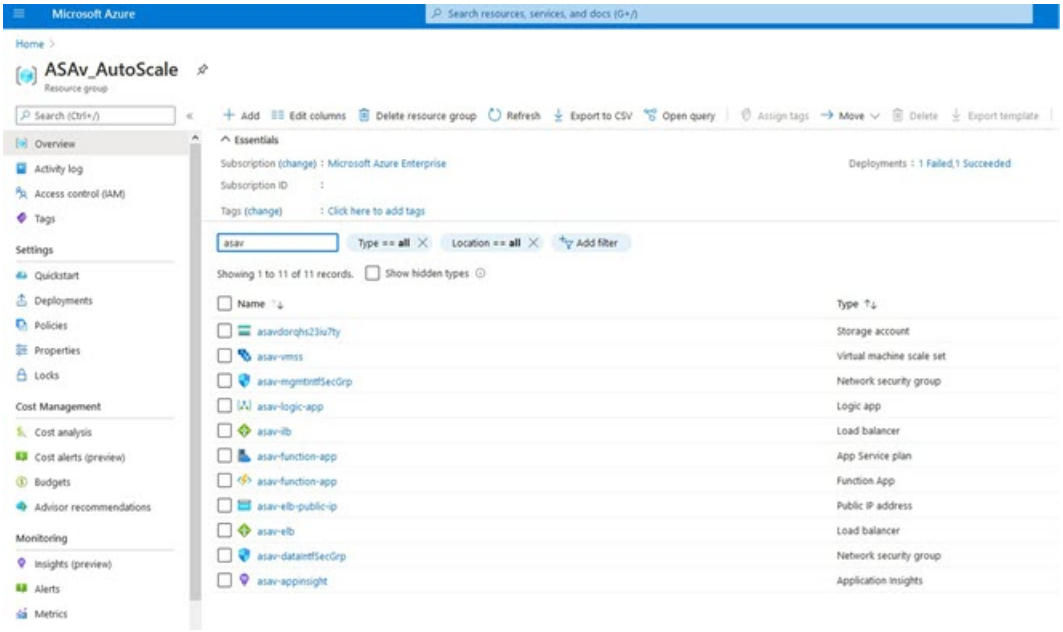


참고 **Edit Parameters(**매개변수 편집**)**를 클릭하고 JSON 파일을 편집하거나 미리 채워진 내용을 업로드할 수도 있습니다.

ARM 템플릿은 입력 검증 기능이 제한되어 있으므로 올바른 입력 검증을 제공하는 것은 사용자의 책임입니다.

단계 **11** 템플릿 구축이 성공하면 Azure용 ASA 가상 Auto Scale 솔루션에 필요한 모든 리소스가 생성됩니다. 다음 그림의 리소스를 참조하십시오. Type(유형) 열은 논리 앱, VMSS, 로드 밸런서, 공용 IP 주소 등 각 리소스에 대해 설명합니다.

그림 ***22: ASA Virtual Auto Scale Template*** 구축



Azure Function 앱 구축

ARM 템플릿을 구축할 때 Azure는 기본 Function 앱을 생성합니다. 그러면 Auto Scale Manager 논리에 필요한 함수를 사용하여 수동으로 업데이트하고 구성해야 합니다.

시작하기 전에

- *ASM_Function.zip* 패키지를 빌드합니다. 소스 코드로 Azure 기능 빌드, 160 페이지의 내용을 참조하십시오.

단계 **1** ARM 템플릿을 구축할 때 생성한 Function 앱으로 이동하여 함수가 없는지 확인합니다. 브라우저에서 다음 URL로 이동합니다.

https://<Function App Name>.scm.azurewebsites.net/DebugConsole

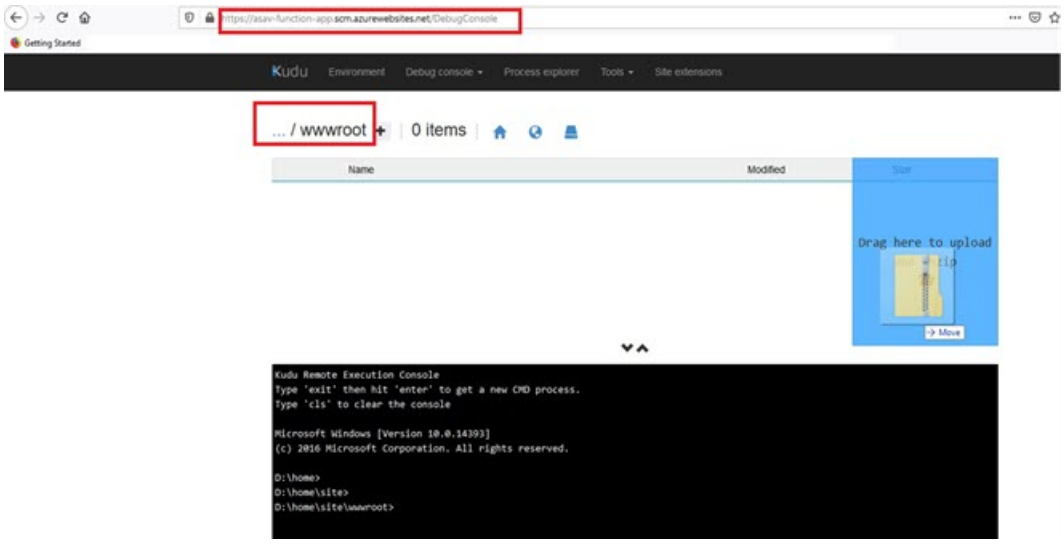
Auto Scale ARM 템플릿 구축, 143 페이지의 예:

https://asav-function-app.scm.azurewebsites.net/DebugConsole

단계 **2** 파일 탐색기에서 **site/wwwroot**로 이동합니다.

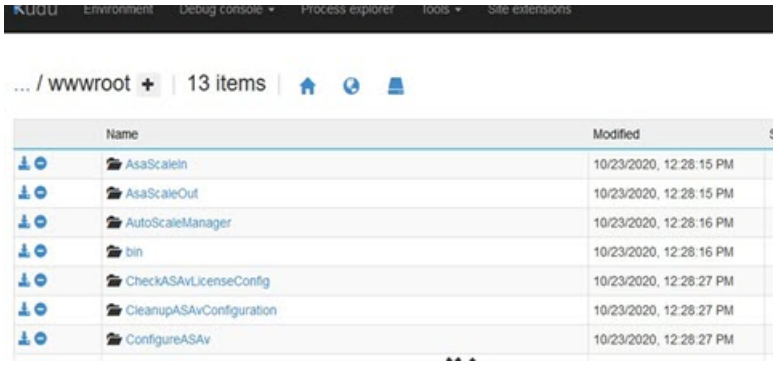
단계 **3** **ASM_Function.zip**을 파일 탐색기의 오른쪽 모서리로 끌어다 놓습니다.

그림 *23: ASA Virtual Auto Scale* 기능 업로드



단계 **4** 업로드에 성공하면 모든 서버리스 함수가 표시됩니다.

그림 *24: ASA* 가상 서버리스 기능

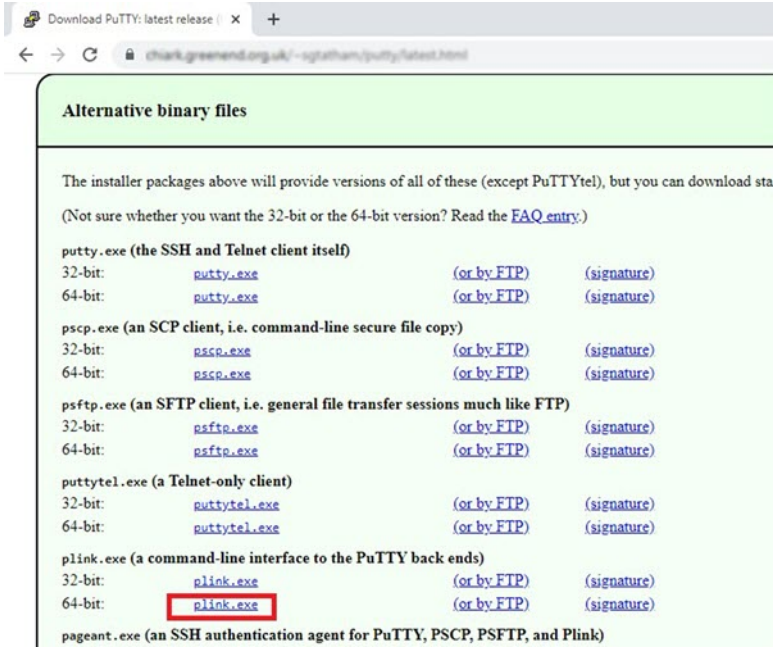


단계 **5** PuTTY SSH 클라이언트를 다운로드합니다.

Azure 함수는 SSH 연결을 통해 ASA 가상에 액세스해야 합니다. 그러나 서버리스 코드에서 사용되는 오픈 소스 라이브러리는 ASA 가상에서 사용하는 SSH 키 교환 알고리즘을 지원하지 않습니다. 따라서 사전 구축된 SSH 클라이언트를 다운로드해야 합니다.

www.putty.org에서 PuTTY 명령줄 인터페이스를 PuTTY 백엔드(*plink.exe*)에 다운로드합니다.

그림 ***25: PuTTY*** 다운로드



단계 **6** SSH 클라이언트 실행 파일의 이름 **plink.exe**를 **asassh.exe**로 변경합니다.

단계 **7** 파일 탐색기의 오른쪽 모서리, 즉 이전 단계에서 **ASM_Function.zip**이 업로드된 위치에 **asassh.exe**를 끌어다 놓습니다.

단계 **8** SSH 클라이언트에 해당 함수 애플리케이션이 있는지 확인합니다. 필요한 경우 페이지를 새로 고칩니다.

컨피그레이션 조정

Auto Scale Manager를 조정하거나 디버깅에 사용할 수 있는 몇 가지 컨피그레이션이 있습니다. 이러한 옵션은 ARM 템플릿에 표시되지 않지만 Function 앱 아래에서 수정할 수 있습니다.

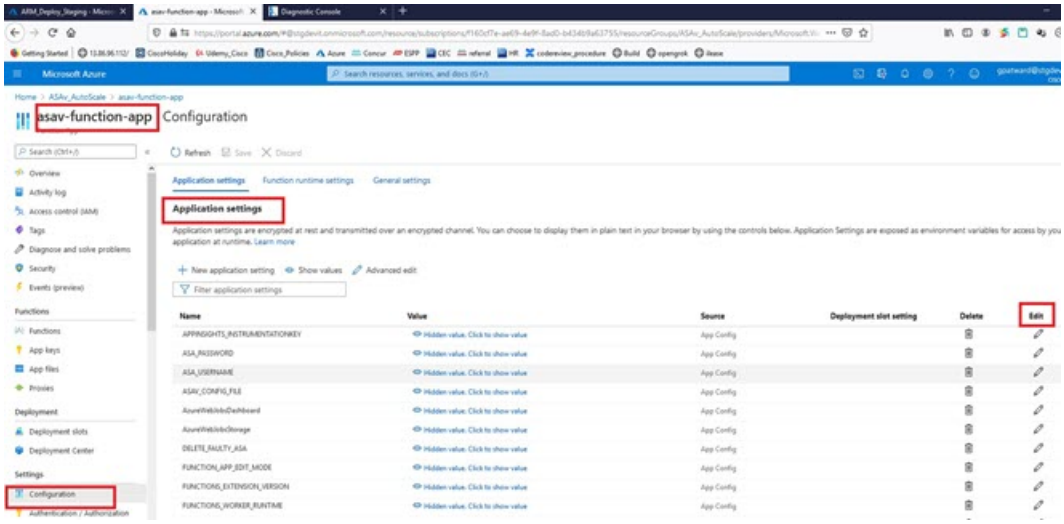
시작하기 전에

참고 이은 언제든지 수정할 수 있습니다. 컨피그레이션을 수정하려면 이 순서를 따릅니다.

- Function 앱을 비활성화합니다.
- 기존의 예약된 작업이 완료 될 때까지 기다립니다.
- 컨피그레이션을 수정하고 저장합니다.
- Function 앱을 활성화합니다.

단계 **1** Azure Portal에서 ASA 가상 함수 애플리케이션을 검색하여 선택합니다.

그림 ***26: ASA*** 가상 기능 애플리케이션



단계 **2** 여기서는 ARM 템플릿을 통해 전달된 컨피그레이션을 수정할 수도 있습니다. 변수 이름은 ARM 템플릿과 다르게 표시될 수 있지만 이러한 변수의 용도를 해당 이름에서 쉽게 식별할 수 있습니다.

대부분의 옵션은 이름에 그 설명을 담고 있습니다. 대표적인 예는 다음과 같습니다.

- 컨피그레이션 이름: “DELETE_FAULTY_ASA”(기본값: YES)

  확장 중에 새 ASA 가상 인스턴스가 시작되고 컨피그레이션 파일을 통해 구성된됩니다. 컨피그레이션이 실패할 경우 이 옵션을 기반으로 Auto Scale Manager는 해당 ASA 가상 인스턴스를 유지하거나 삭제할지 결정합니다. (예: 결함 ASA 가상 삭제 / 아니요: 컨피그레이션이 실패하더라도 ASA 가상 인스턴스를 유지합니다.)

- Function 앱 설정에서는 Azure 구독에 대한 액세스 권한이 있는 사용자가 모든 변수('password'와 같은 보안 문자열을 포함하는 변수 포함)를 일반 텍스트 형식으로 볼 수 있습니다.

  사용자가 이에 대해 보안 문제가 있는 경우(예: 조직 내에서 권한이 낮은 사용자 간에 Azure 구독이 공유되는 경우) 사용자는 Azure의 *Key Vault* 서비스를 사용하여 비밀번호를 보호할 수 있습니다. 이 기능이 구성되면 기능 설정에서 일반 텍스트 '비밀번호'를 제공하는 대신 비밀번호가 저장된 키 저장소에서 생성된 보안 식별자를 제공해야 합니다.

  참고 Azure 문서를 검색하여 애플리케이션 데이터를 보호하는 모범 사례를 찾습니다.

가상 시스템 확장 집합의 IAM 역할 구성

Azure IAM (Identity and Access Management)은 사용자 ID를 관리하고 제어하기 위해 Azure Security and Access Control의 일부로 사용됩니다. Azure 리소스의 관리되는 ID는 Azure Active Directory의 자동으로 관리되는 ID를 Azure 서비스에 제공합니다.

이를 통해 Function 앱은 명시적 인증 자격 증명 없이 VMSS(Virtual Machine Scale Sets)를 제어할 수 있습니다.

단계 **1** Azure 포털에서 VMSS로 이동합니다.

단계 **2** 액세스 제어**(IAM)**를 클릭합니다.

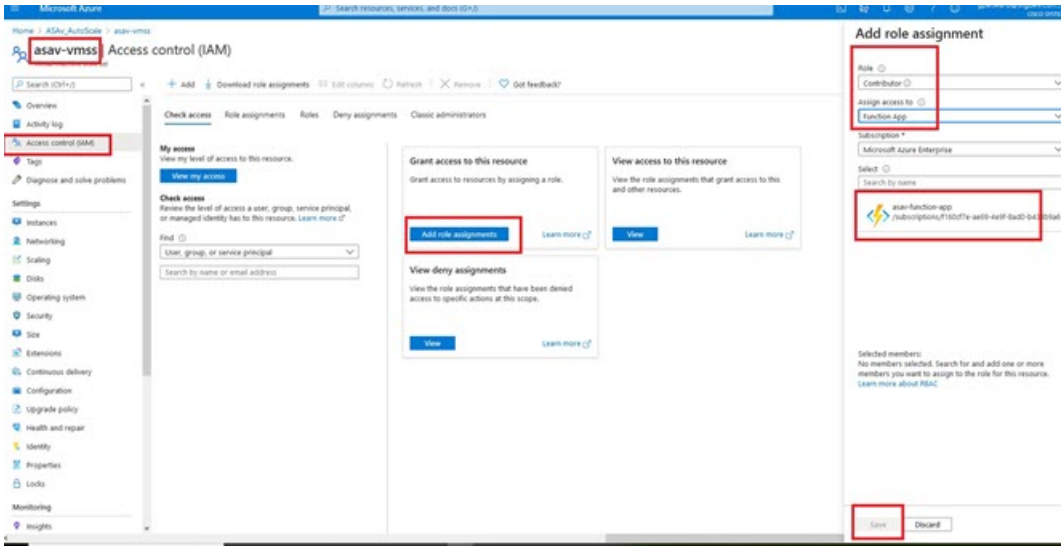
단계 **3** **Add(**추가**)**를 클릭하여 역할 할당을 추가합니다.

단계 **4** **Add role Assignment(**역할 할당 추가**)** 드롭 다운에서 **Contributor(**기여자**)**를 선택합니다.

단계 **5** **Assign access to(**액세스 할당 대상**)** 드롭 다운에서 **Function App(**기능 앱**)**을 선택합니다.

단계 **6** ASA 가상 함수 애플리케이션을 선택합니다.

그림 ***27: AIM*** 역할 할당



단계 **7** **Save(**저장**)**를 클릭합니다.

참고 또한 아직 시작된 ASA 가상 인스턴스가 없는지 확인해야 합니다.

보안 그룹 업데이트

ARM 템플릿은 Management 인터페이스용과 데이터 인터페이스용의 두 가지 보안 그룹을 생성합니다. Management 보안 그룹은 ASA 가상 관리 활동에 필요한 트래픽만 허용합니다. 그러나 데이터 인터페이스 보안 그룹은 모든 트래픽을 허용합니다.

구축의 토폴로지 및 애플리케이션 요구 사항에 따라 보안 그룹 규칙을 세부적으로 조정합니다.

참고 데이터 인터페이스 보안 그룹은 로드 밸런서의 최소 SSH 트래픽을 허용해야 합니다.

Azure Logic 앱 업데이트

Logic 앱은 Autoscale 기능의 오케스트레이터 역할을 합니다. ARM 템플릿은 기본 Logic 앱을 생성합니다. 그러면 Auto Scale 오케스트레이터로 작동하는 데 필요한 정보를 제공할 수 있도록 수동으로 업데이트해야 합니다.

단계 **1** 리포지토리에서 *LogicApp.txt* 파일을 로컬 시스템으로 검색하고 아래 표시된 대로 수정합니다.

중요 계속하기 전에 이 단계를 모두 읽고 숙지하십시오.

이러한 수동 단계는 ARM 템플릿에서 자동화되지 않으므로 나중에 Logic 앱만 독립적으로 업그레이드 할 수 있습니다.

a) 필수: “SUBSCRIPTION_ID”의 모든 어커런스를 찾아서 구독 ID 정보로 교체합니다.
b) 필수: "RG_NAME" 어커런스를 모두 찾아서 리소스 그룹 이름으로 바꿉니다.
c) 필수: "FUNCTIONAPPNAME" 어커런스를 모두 찾아서 함수 앱 이름으로 바꿉니다.

다음 예에서는 *LogicApp.txt* 파일에서 이러한 행 중 일부를 보여줍니다.

```
  "AutoScaleManager": {
      "inputs": {
          "function": {
              "id":
"/subscriptions/SUBSCRIPTION_ID/resourceGroups/RG_NAME/providers/Microsoft.Web/sites/FUNCTIONAPPNAME/functions/AutoScaleManager"

          }
.
.
                      },
                      "Deploy_Changes_to_ASA": {
                          "inputs": {
                              "body": "@body('AutoScaleManager')",
                              "function": {
                                  "id":
"/subscriptions/SUBSCRIPTION_ID/resourceGroups/RG_NAME/providers/Microsoft.Web/sites/FUNCTIONAPPNAME/functions/DeployConfiguration"

                              }
.
.
                      "DeviceDeRegister": {
                          "inputs": {
                              "body": "@body('AutoScaleManager')",
                              "function": {
                                  "id":
"/subscriptions/SUBSCRIPTION_ID/resourceGroups/RG_NAME/providers/Microsoft.Web/sites/FUNCTIONAPPNAME/functions/DeviceDeRegister"

                              }
                          },
                          "runAfter": {
                              "Delay_For_connection_Draining": [
```

d) (선택 사항) 트리거 간격을 수정하거나 기본값(5)을 유지합니다. 이는 Autoscale 기능이 주기적으로 트리거되는 시간 간격입니다. 다음 예는 *LogicApp.txt* 파일에서 이러한 행을 보여줍니다.

```
"triggers": {
    "Recurrence": {
        "conditions": [],
        "inputs": {},
        "recurrence": {
            "frequency": "Minute",
            "interval": 5
        },
```

e) (선택 사항) 드레인 시간을 수정하거나 기본값(5)을 유지합니다. 이는 축소(Scale-In) 작업 중에 디바이스를 삭제하기 전에 ASA 가상에서 기존 연결을 드레 이닝하는 시간 간격입니다. 다음 예는 *LogicApp.txt* 파일에서 이러한 행을 보여줍니다.

```
"actions": {
     "Branch_based_on_Scale-In_or_Scale-Out_condition": {
         "actions": {
             "Delay_For_connection_Draining": {
                 "inputs": {
                     "interval": {
                         "count": 5,
                         "unit": "Minute"
                     }
```

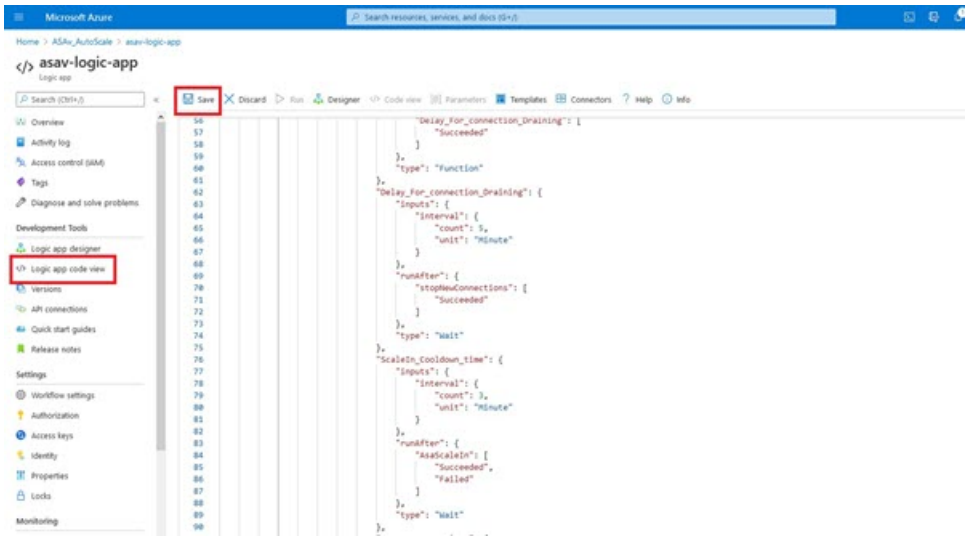
f) (선택 사항) 냉각 시간을 수정하거나 기본값(10)을 유지합니다. 이 시간은 확장(Scale-Out)이 완료된 후 작업 없음을 유지하는 시간입니다. 다음 예는 *LogicApp.txt* 파일에서 이러한 행을 보여줍니다.

```
"actions": {
    "Branch_based_on_Scale-Out_or_Invalid_condition": {
        "actions": {
            "Cooldown_time": {
                "inputs": {
                    "interval": {
                        "count": 10,
                        "unit": "Second"
                    }
```

참고 이러한 단계는 Azure 포털에서도 수행할 수 있습니다. 자세한 내용은 Azure 문서를 참조하십시오.

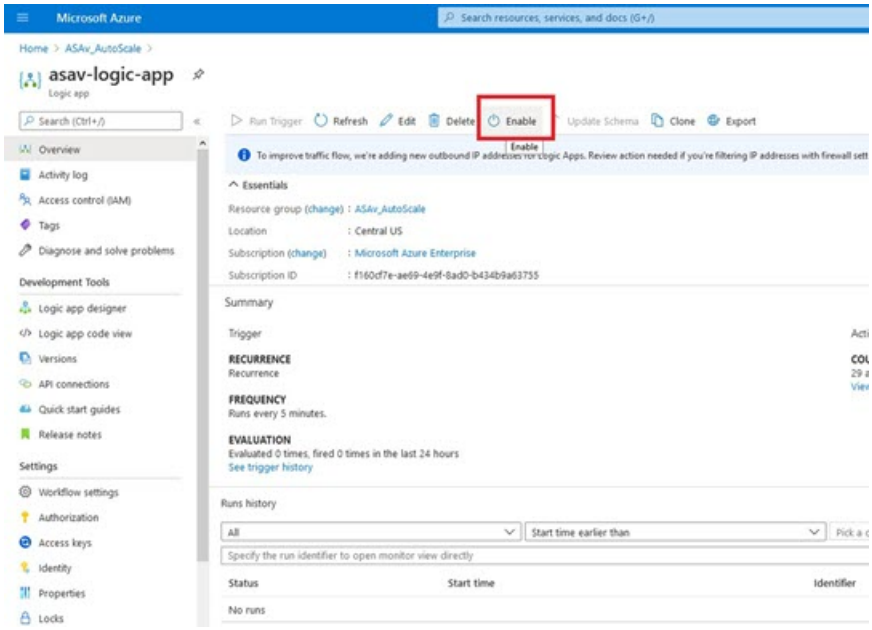
단계 **2** **Logic** 앱 코드 보기로 이동하여 기본 콘텐츠를 삭제하고 수정된 *LogicApp.txt* 파일에서 콘텐츠를 붙여넣고 **Save(**저장**)**을 클릭합니다.

그림 ***28: Logic*** 앱 코드 보기



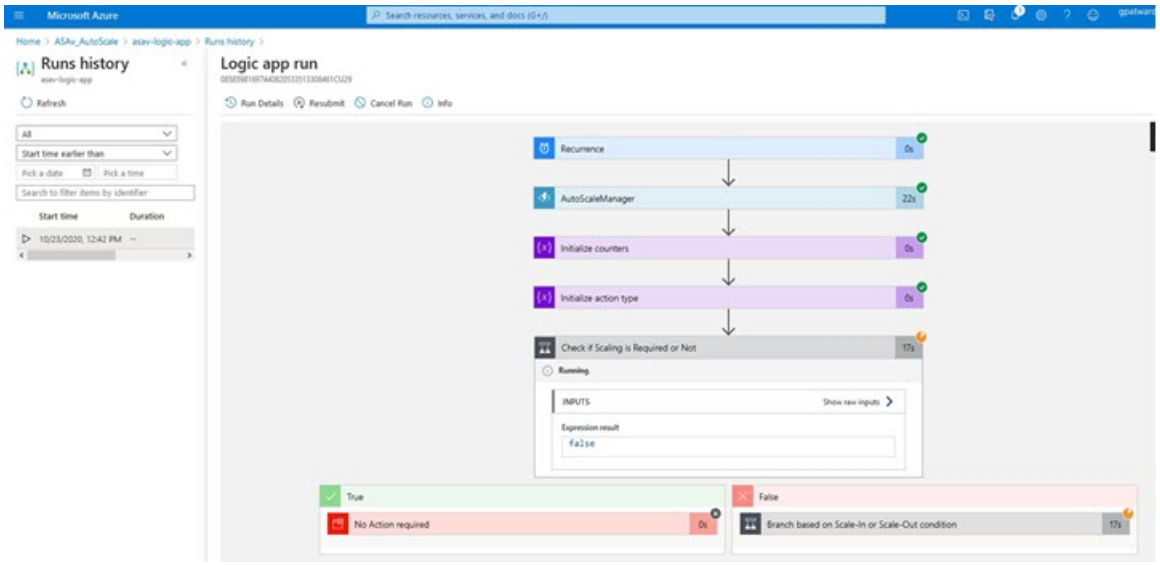
단계 **3** Logic 앱을 저장하면 '비활성화' 상태가 됩니다. Auto Scale Manager를 시작하려면 **Enable(**활성화**)**을 클릭합니다.

그림 ***29: Logic*** 앱 활성화



단계 **4** 활성화되면 작업이 실행되기 시작합니다. 활동을 보려면 '실행 중' 상태를 클릭하십시오.

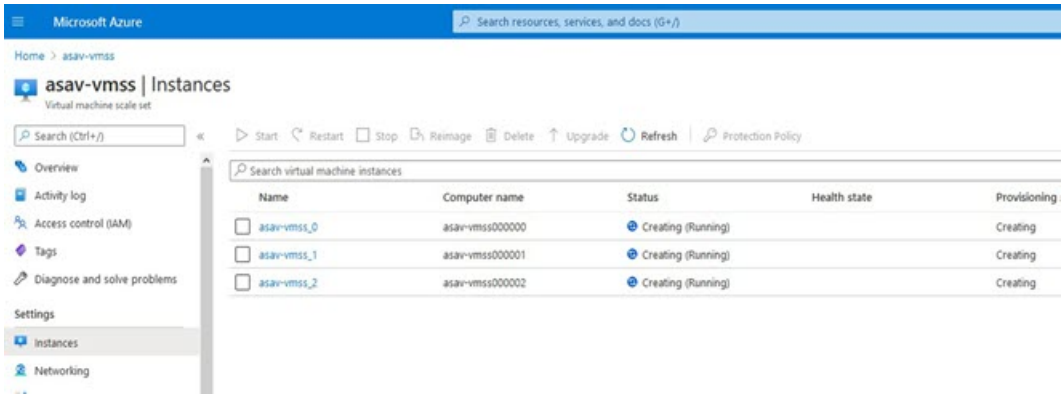
그림 ***30: Logic*** 앱 실행 상태



단계 **5** Logic 앱이 시작되면 모든 구축 관련 단계가 완료됩니다.

단계 **6** VMSS에서 ASA 가상 인스턴스가 생성되고 있는지 확인합니다.

그림 ***31:*** 실행 중인 ***ASA*** 가상 인스턴스



이 예에서는 ARM 템플릿 구축에서 ' 'minAsaCount'가 '3'으로, 'initDeploymentMode'가 'BULK'로 설정되었으므로 3개의 ASA 가상 인스턴스가 시작됩니다.

Threat Defense VirtualASA 가상 업그레이드

ASA 가상 업그레이드는 VMSS(Virtual Machine Scale Set)의 이미지 업그레이드 형식으로만 지원됩니다. 따라서 Azure REST API 인터페이스를 통해 ASA 가상를 업그레이드합니다.

참고 모든 REST 클라이언트를 사용하여 ASA 가상를 업그레이드할 수 있습니다.

시작하기 전에

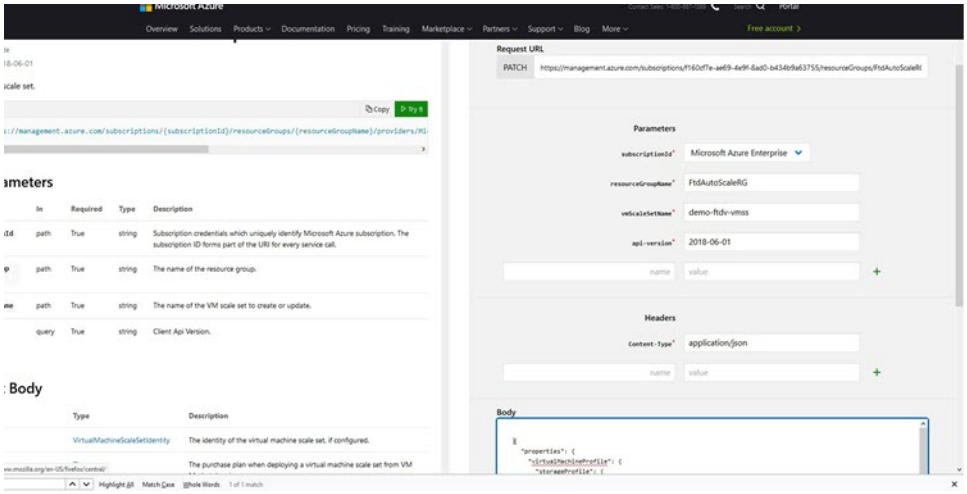
- 마켓플레이스에서 사용 가능한 새 ASA 가상 이미지 버전을 가져옵니다(예: 914.001).
- 원래 스케일 세트를 구축하는 데 사용된 SKU를 가져옵니다(예: asav -azure-byol).
- 리소스 그룹 및 가상 시스템 확장 집합 이름을 가져옵니다.

단계 **1** 브라우저에서 다음 URL로 이동합니다.

https://docs.microsoft.com/en-us/rest/api/compute/virtualmachinescalesets/update#code-try-0

단계 **2** 매개변수 섹션에 세부 사항을 입력합니다.

그림 ***32: Threat Defense VirtualASA*** 가상 업그레이드



단계 **3** 본문 섹션에 새로운 ASA 가상 이미지 버전, SKU 및 트리거 RUN을 포함하는 JSON 입력을 입력합니다.

```
{
 "properties": {
        "virtualMachineProfile": {
              "storageProfile": {
                      "imageReference": {
                           "publisher": "cisco",
                           "offer": "cisco-asav",
                           "sku": "asav-azure-byol",
                            "version": "650.32.0"
                      }
                  },
            }
      }
}
```

단계 **4** Azure의 성공적인 응답은 VMSS가 변경 사항을 수락했음을 의미합니다.

새 ASA 가상 이미지는 새 인스턴스에서 사용되며, 이는 확장 작업의 일부로 시작됩니다.

- 기존 ASA 가상 인스턴스는 확장 집합에 있는 동안 기존 소프트웨어 이미지를 계속 사용합니다.
- 위의 동작을 재정의하고 기존 ASA 가상 인스턴스를 수동으로 업그레이드할 수 있습니다. 이렇게 하려면 VMSS에서 **Upgrade(**업그레이드**)** 버튼을 클릭합니다. 선택한 ASA 가상 인스턴스가 재부팅되고 업그레이드됩니다. 이러한 업그레이드된 ASA 가상 인스턴스를 수동으로 다시 등록하고 재구성해야 합니다. 이 방법은 권장되지 않습니다**.**

Auto Scale 논리

확장 논리

- **POLICY-1**: 어떤 경우든 ASA 가상 평균로드가 구성된 기간 동안 확장 임계값을 초과하면 확장(Scale-Out)이 트리거됩니다.
- **POLICY-2**: 구성된 기간 동안 모든 ASA 가상 디바이스의 평균로드가 확장 임계값을 초과하면 확장(Scale-Out)이 트리거됩니다.

축소 논리

- 모든 ASA 가상 디바이스의 CPU 사용률이 구성된 기간 동안 구성된 축소 임계값 미만인 경우.

Notes(참고**)**

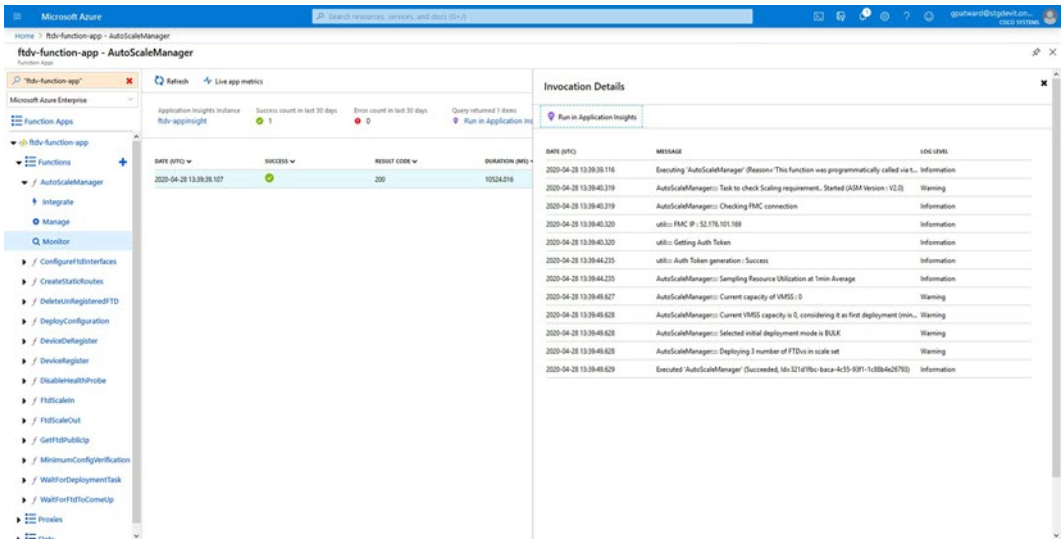
- 축소(Scale-In)/확장(Scale-Out)은 1단계로 수행됩니다(즉, 한 번에 1개 ASA 가상만 축소/확장).
- 위의 논리는 로드 밸런서가 모든 ASA 가상 디바이스에 연결을 동일하게 분산하려고 시도한다는 가정을 기반으로 하며 평균적으로 모든 ASA 가상 디바이스를 동일하게 로드해야 합니다.

Auto Scale 로깅 및 디버깅

서버리스 코드의 각 구성 요소에는 자체 로깅 메커니즘이 있습니다. 또한 로그는 애플리케이션 인사이트에 게시됩니다.

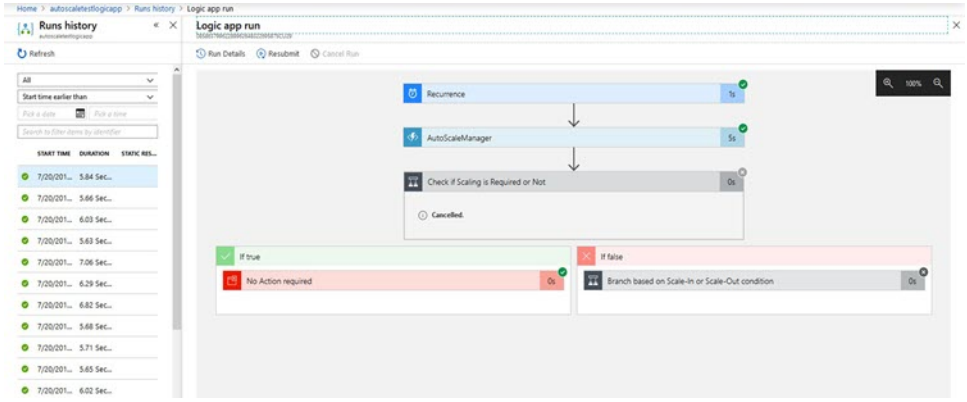
- 개별 Azure 함수의 로그를 볼 수 있습니다.

그림 ***33: Azure Function*** 로그



- 각 Logic App 실행 및 해당 개별 구성 요소에 대한 유사한 로그를 볼 수 있습니다.

  그림 ***34: Logic*** 앱 실행 로그



- 필요한 경우 Logic App에서 실행중인 작업을 언제든지 중지/종료할 수 있습니다. 그러나 현재 실행 중이거나 종료되는 ASA 가상 디바이스는 일관성이 없는 상태로 유지됩니다.
- 각 실행/개별 작업에 소요되는 시간은 Logic 앱에서 확인할 수 있습니다.
- 언제든지 새 zip을 업로드하여 Function 앱을 업그레이드할 수 있습니다. Function 앱을 업그레이드하기 전에 Logic 앱을 중지하고 모든 작업이 완료될 때까지 기다립니다.

Auto Scale 지침 및 제한 사항

ASA 가상 Auto Scale for Azure를 구축할 때 다음 지침 및 제한 사항에 유의하십시오.

- 확장 결정은 CPU 사용률을 기반으로 합니다.

- ASA 가상 Management 인터페이스가 공용 IP 주소를 갖도록 구성되었습니다.
- IPv4만 지원됩니다.
- ARM 템플릿은 입력 검증 기능이 제한되어 있으므로 올바른 입력 검증을 제공하는 것은 사용자의 책임입니다.
- Azure 관리자는 Function 앱 환경 내에서 민감한 데이터(예: 관리자 로그인 자격 증명 및 비밀번호)를 일반 텍스트 형식으로 볼 수 있습니다. *Azure Key Vault* 서비스를 사용하여 민감한 데이터를 보호할 수 있습니다.

Auto Scale 문제 해결

다음은 일반적인 오류 시나리오 및 ASA 가상 Auto Scale for Azure에 대한 디버깅 팁입니다.

- ASA 가상로 SSH할 수 없음 : 템플릿을 통해 복잡한 비밀번호가 ASA 가상에 전달되는지 확인합니다. 보안 그룹에서 SSH 연결을 허용하는지 확인하십시오.
- 로드 밸런서 상태 확인 실패: ASA 가상에서 데이터 인터페이스의 SSH에 응답하는지 확인합니다. 보안 그룹 설정을 확인합니다.
- 트래픽 문제: 로드 밸런서 규칙, NAT 규칙 / ASA 가상에 구성된 고정 경로를 확인합니다. 템플릿 및 보안 그룹 규칙에 제공된 Azure 가상 네트워크 / 서브넷 / 게이트웨이 세부 정보를 확인합니다.
- Logic 앱이 VMSS에 액세스하지 못함: VMSS의 IAM 역할 컨피그레이션이 올바른지 확인하십시오.
- Logic 앱이 매우 오랫동안 실행 됨: 확장된 ASA 가상 디바이스에서 SSH 액세스를 확인합니다. Azure VMSS에서 ASA 가상 디바이스의 상태를 확인합니다.
- 구독 ID와 관련된 오류 발생 Azure Function: 계정에서 기본 구독이 선택되었는지 확인하십시오.
- 축소(Scale-In) 작업 실패: 경우에 따라 Azure에서 인스턴스를 삭제하는 데 시간이 오래 걸리는 경우가 있습니다. 이러한 상황에서는 축소 작업이 시간 초과되고 오류를 보고할 수 있지만 결국엔 인스턴스가 삭제됩니다.
- 컨피그레이션 변경을 수행하기 전에 논리 애플리케이션을 비활성화하고 실행중인 모든 작업이 완료될 때까지 기다리십시오.

소스 코드로 Azure 기능 빌드

시스템 요구 사항

- Microsoft Windows 데스크톱 / 노트북
- Visual Studio(Visual Studio 2019 버전 16.1.3에서 테스트)

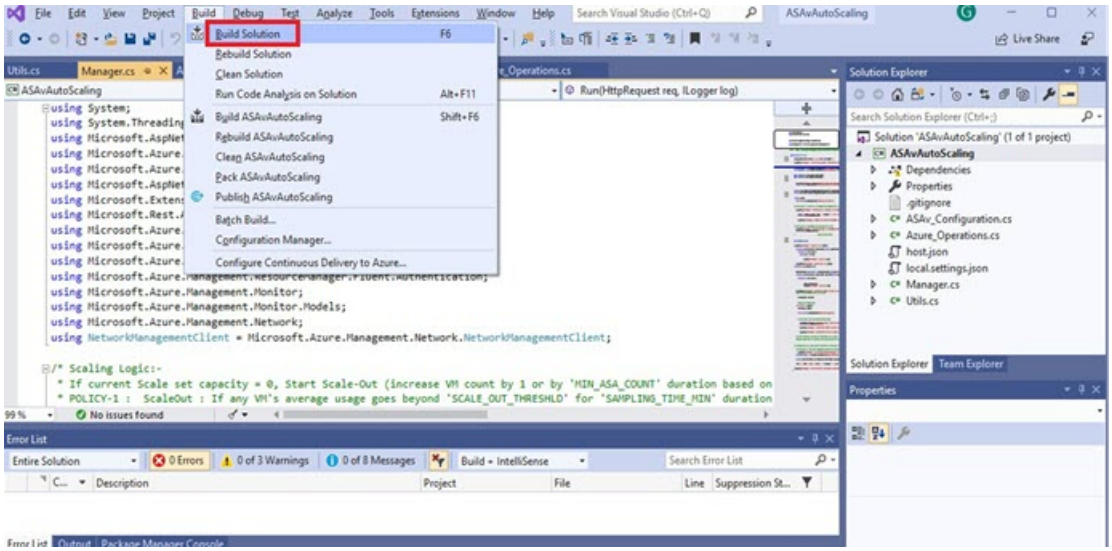
참고 Azure 함수는 C#을 사용하여 작성됩니다.

- "Azure 개발" 워크로드를 Visual Studio에 설치해야 합니다.

Visual Studio로 빌드

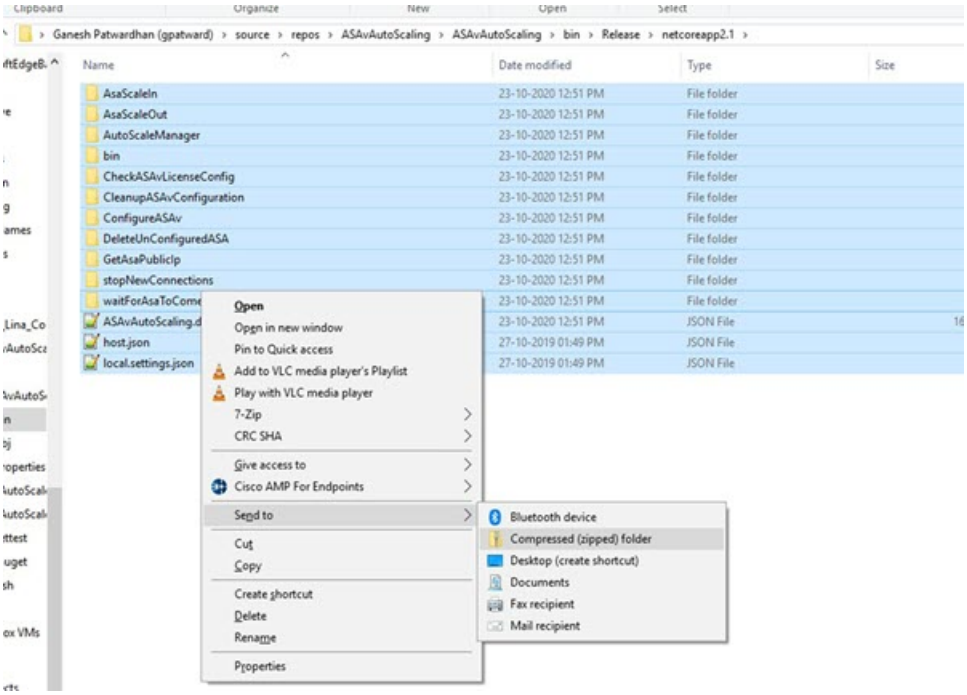
1. 'code' 폴더를 로컬 시스템에 다운로드합니다.
2. ' ASAAutoScaling'폴더로 이동합니다.
3. Visual Studio에서 ' ASAAutoScaling.csproj' 프로젝트 파일을 엽니다.
4. Visual Studio 표준 절차를 사용하여 정리 및 빌드합니다.

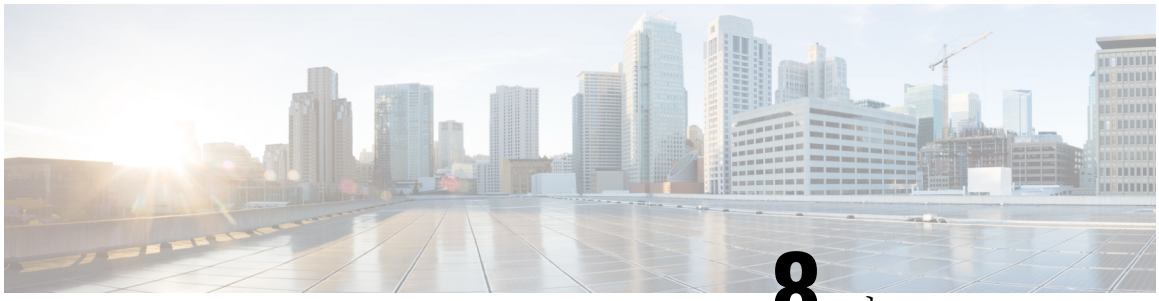
그림 ***35: Visual Studio*** 빌드



5. 빌드가 성공적으로 컴파일되면 **\bin\Release\netcoreapp2.1** 폴더로 이동합니다.
6. 모든 내용을 선택하고 **Send to** > **Compressed(zipped)(**압축 폴더로 전송**)**을 클릭하고 ZIP 파일을 *ASM_Function.zip*으로 저장합니다.

그림 *36: Build ASM_Function.zip*





8 장

Rackspace 클라우드에 ASA 가상을 구축

Rackspace 클라우드에 ASA 가상을 구축할 수 있습니다.

중요 9.13(1)부터 모든 ASA 가상 라이선스는 지원되는 모든 ASA 가상 vCPU/메모리 구성에서 사용할 수 있습니다. 따라서 ASA 가상 고객은 다양한 VM 리소스 사용 공간에서 실행할 수 있습니다.

Rackspace 클라우드에 ASA 가상 구축 정보

Rackspace는 모든 주요 퍼블릭 및 프라이빗 클라우드 기술에서 전문 지식과 관리형드 서비스를 제공하는 최고의 업체입니다. Rackspace 클라우드는 유틸리티 컴퓨팅을 기준으로 청구되는 클라우드 컴퓨팅 제품 및 서비스 집합입니다.

Rackspace 클라우드에서 가상 어플라이언스로 Rackspace용 를 구축할 수 있습니다.ASA 가상 이 장에서는 단일 인스턴스 ASA 가상 어플라이언스를 설치하고 구성하는 방법을 설명합니다.

Rackspace 클라우드의 인스턴스 유형을 버전이라고 합니다. 버전이라는 용어는 서버의 RAM 크기, vCPU, 네트워크 처리량(RXTX 계수) 및 디스크 공간의 조합을 의미합니다. 다음 표에는 ASA 가상 구축에 적합한 Rackspace 버전이 나와 있습니다.

표 ***20: Rackspace*** 지원 버전

Flavor	특성		집계 대역폭
	vCPUs	메모리**(GB)**	
general 1-2	2	2	400 Mbps

Flavor	특성		집계 대역폭
	vCPUs	메모리(GB)	
general 1-4	4	4	800Mbps
general 1-8	8	8	1.6Gbps
compute 1-4	2	3.75	312.5Mbp
compute 1-8	4	7.5	625Mbps
compute 1-15	8	15	1.3Gbps
memory 1-15	2	15	625Mbps
memory 1-15	4	30	1.3Gbps
memory 1-15	8	60	2.5Gbps

Rackspace 유형 정보

Rackspace 가상 클라우드 서버 버전은 다음 클래스로 분류됩니다.

- 범용 **v1**
  - 범용 워크로드에서 고성능 웹사이트에 이르는 다양한 사용 사례에 유용합니다.
  - vCPU는 초과 구독되고 "버스트 가능"합니다. 즉, 물리적 CPU 스레드 수보다 물리적 호스트의 클라우드 서버에 할당된 vCPU가 더 많습니다.
- 컴퓨팅 **v1**
  - 웹 서버, 애플리케이션 서버 및 기타 CPU 집약적 워크로드에 최적화되어 있습니다.
  - vCPU는 "예약"됩니다. 즉, 호스트의 물리적 CPU 스레드 수는 물리적 호스트의 클라우드 서버에 할당된 vCPU를 절대로 초과하지 않습니다.
- 메모리 **v1**
  - 메모리 집약적 워크로드에 권장됩니다.
- **I/O v1**
  - 빠른 디스크 I/O가 도움이 되는 고성능 애플리케이션 및 데이터베이스에 적합합니다.

ASA 가상 및 Rackspace에 대한 사전 요건

- Rackspace 계정을 만듭니다.

모든 Rackspace 퍼블릭 클라우드 계정은 기본적으로 매니지드 인프라 서비스 수준으로 설정됩니다. 클라우드 제어판 내에서 관리형 운영 서비스 수준으로 업그레이드할 수 있습니다. 클라우드 제어판 상단에서 계정 사용자 이름을 클릭하고 Upgrade Service Level(서비스 수준 업그레이드)을 선택합니다.

- ASA 가상에 라이선스를 부여합니다. ASA 가상 라이선스를 등록하기 전에는 저성능 모드에서 실행됩니다. 이 모드에서는 100개의 연결 및 100Kbps의 처리량만 허용됩니다. ASA 가상에 대한 라이선싱, 1 페이지의 내용을 참조하십시오.
- 인터페이스 요건:
  - 관리 인터페이스
  - 내부 및 외부 인터페이스
  - (선택 사항) 추가 서브넷(DMZ)
- 통신 경로:
  - 관리 인터페이스 - ASA 가상을 ASDM에 연결할 때 사용합니다. 통과 트래픽에는 사용할 수 없습니다.
  - 내부 인터페이스(필수)—ASA 가상를 내부 호스트에 연결하는 데 사용합니다.
  - 외부 인터페이스(필수)—ASA 가상를 공용 네트워크에 연결하는 데 사용합니다.
  - DMZ 인터페이스(선택 사항)—ASA 가상을 DMZ 네트워크에 연결하는 데 사용합니다.
- ASA 및 ASA 가상 시스템 호환성과 요구 사항은 Cisco Secure Firewall ASA호환성을 참조하십시오.

Rackspace 클라우드 네트워크

클라우드 구성에는 필요에 따라 적절하게 연결되는 여러 유형의 네트워크가 포함될 수 있습니다. 클라우드 서버의 네트워킹 기능은 다른 네트워크에서와 동일한 방식으로 관리할 수 있습니다. ASA 가상 구축은 Rackspace 클라우드에서 주로 세 가지 유형의 가상 네트워크와 상호 작용합니다.

- PublicNet - 클라우드 서버, 클라우드 로드 밸런서 및 네트워크 어플라이언스 같은 클라우드 인프라 구성 요소를 인터넷에 연결합니다.
  - PublicNet을 사용하여 ASA 가상을 인터넷에 연결합니다.
  - ASA 가상은 Management0/0 인터페이스를 통해 이 네트워크에 연결합니다.
  - PublicNet은 IPv4 및 IPv6에 대해 이중 스택됩니다. PublicNet을 사용하여 서버를 생성하는 경우 서버는 기본적으로 IPv4 주소 및 IPv6 주소를 수신합니다.
- ServiceNet - 각 Rackspace 클라우드 영역 내의 내부 IPv4 전용 다중 테넌트 네트워크입니다.
  - ServiceNet은 구성 내 서버 간 트래픽 전달에 최적화되어 있습니다(east-west 트래픽).

- 서버는 클라우드 파일, 클라우드 로드 밸런서, 클라우드 데이터베이스 및 클라우드 백업 같은 지역화된 서비스에 무료로 액세스할 수 있습니다.
- 네트워크 10.176.0.0/12 및 10.208.0.0/12는 ServiceNet용으로 예약됩니다. ServiceNet에 연결이할 수 있는 모든 서버는 이러한 네트워크 중 하나의 IP 주소로 프로비저닝됩니다.
- ASA 가상은 Gigabit0/0 인터페이스를 통해 이 네트워크에 연결합니다.

- 프라이빗 클라우드 네트워크 - 클라우드 네트워크를 사용하면 클라우드에서 안전하고 격리된 네트워크를 생성하고 관리할 수 있습니다.
  - 이러한 네트워크는 완전한 단일 테넌트이며, 사용자는 네트워크 토폴로지, IP 주소 지정(IPv4 또는 IPv6) 및 연결되는 클라우드 서버를 완벽하게 제어할 수 있습니다.
  - 클라우드 네트워크의 범위는 지역적이며, 지정된 지역의 모든 클라우드 서버에 연결할 수 있습니다.
  - API를 통하거나 Rackspace 클라우드 제어판을 사용하여 클라우드 네트워크를 생성하고 관리할 수 있습니다.

    ASA 가상은 Gigabit0/1~Gigabit0/8 인터페이스를 통해 이러한 네트워크에 연결합니다.

Rackspace Day 0 컨피그레이션

VM이 Rackspace 클라우드에 구축되면 Rackspace 프로비저닝 정보가 포함된 파일이 들어 있는 CD-ROM 디바이스가 VM에 연결됩니다. 프로비저닝 정보에는 다음이 포함됩니다.

- 호스트 이름
- 필수 인터페이스의 IP 주소
- 고정 IP 경로
- 사용자 이름 및 비밀번호(SSH 공개 키(선택 사항))
- DNS 서버
- NTP 서버

이러한 파일을 초기 구축 중에 읽고 ASA 구성이 생성됩니다.

ASA 가상 호스트 이름

기본적으로 ASA 가상 호스트 이름은 ASA 가상 구축을 시작할 때 클라우드 서버에 할당하는 이름입니다.

```
hostname rackspace-asav
```

ASA 호스트 이름 구성은 RFC 1034 및 1101을 준수하는 호스트 이름만 수락합니다.

- 문자 또는 숫자로 시작하고 끝나야 합니다.
- 내부 문자는 문자, 숫자 또는 하이픈이어야 합니다.

참고 ASA 가상은 이러한 규칙을 준수하도록 클라우드 서버 이름을 변경하지만, 원래 클라우드 서버 이름에 최대한 가깝게 만듭니다. 클라우드 서버 이름의 시작과 끝 부분에 있는 특수 문자를 삭제하고 호환되지 않는 내부 문자를 하이픈으로 대체합니다.

예를 들어 이름이 **ASAv-9.13.1.200**인 클라우드 서버의 호스트 이름은 **ASAv-9-13-1-200**입니다.

인터페이스

인터페이스는 다음과 같은 방식으로 구성됩니다.

- Management 0/0
  - PublicNet에 연결되어 있으므로 'outside'로 이름이 지정됩니다.
  - Rackspace는 IPv4 및 IPv6 공용 주소를 PublicNet 인터페이스에 할당합니다.
- Gigabit0/0
  - ServiceNet에 연결되어 있으므로 'management'로 이름이 지정됩니다.
  - Rackspace는 Rackspace 영역에 대한 ServiceNet 서브넷의 IPv4 주소를 할당합니다.
- Gigabit0/1~Gigabit0/8
  - 프라이빗 클라우드 네트워크에 연결되어 있으므로 'inside', 'inside02', 'inside03' 등으로 이름이 지정됩니다.
  - Rackspace는 클라우드 네트워크 서브넷의 IP 주소를 할당합니다.

인터페이스 3개가 있는 ASA 가상의 인터페이스 구성은 다음과 같은 모습입니다.

```
interface GigabitEthernet0/0
 nameif management
 security-level 0
 ip address 10.176.5.71 255.255.192.0
!
interface GigabitEthernet0/1
 nameif inside
 security-level 100
 ip address 172.19.219.7 255.255.255.0
!
interface Management0/0
 nameif outside
 security-level 0
 ip address 162.209.103.109 255.255.255.0
 ipv6 address 2001:4802:7800:1:be76:4eff:fe20:1763/64
```

Static Routes(고정 경로)

Rackspace는 다음 고정 IP 경로를 프로비저닝합니다.

- PublicNet 인터페이스를 통한 기본 IPv4 경로(**outside**).
- PublicNet 인터페이스를 통한 기본 IPv6 경로.
- ServiceNet 인터페이스의 인프라 서브넷 경로(**management**).

```
route outside 0.0.0.0 0.0.0.0 104.130.24.1 1
ipv6 route outside ::/0 fe80::def
route management 10.176.0.0 255.240.0.0 10.176.0.1 1
route management 10.208.0.0 255.240.0.0 10.176.0.1 1
```

로그인 자격 증명

Rackspace에서 생성한 비밀번호를 사용하여 사용자 이름 'admin'을 생성합니다. 클라우드 서버가 Rackspace 공개 키를 사용하여 구축되었다면, 사용자 'admin'에 대한 공개 키가 생성됩니다.

```
username admin password <admin_password> privilege 15
username admin attributes
 ssh authentication publickey <public_key>
```

Day0 SSH 컨피그레이션:

- PublicNet 인터페이스를 통한 SSH**(outside)**는 IPv4 및 IPv6에 대해 활성화됩니다.
- ServiceNet 인터페이스를 통한 SSH**(management)**는 IPv4에 대해 활성화됩니다.
- Rackspace의 요청에 따라 더 강력한 키 교환 그룹을 구성합니다.

```
aaa authentication ssh console LOCAL
ssh 0 0 management
ssh 0 0 outside
ssh ::0/0 outside
ssh version 2
ssh key-exchange group dh-group14-sha1
```

DNS 및 NTP

Rackspace는 DNS 및 NTP에 사용할 IPv4 서비스 주소 2개를 제공합니다.

```
dns domain-lookup outside
dns server-group DefaultDNS
 name-server 69.20.0.164
 name-server 69.20.0.196

ntp server 69.20.0.164
ntp server 69.20.0.196
```

Rackspace 클라우드에 ASA 가상을 구축

Rackspace 클라우드에 ASA 가상을 구축

Rackspace 클라우드에서 ASA 가상을 가상 어플라이언스로서 구축할 수 있습니다. 이 절차에서는 단일 인스턴스 ASA 가상 어플라이언스를 설치하는 방법을 보여줍니다.

시작하기 전에

호스트 이름 요구 사항, 인터페이스 프로비저닝 및 네트워킹 정보를 포함하여 성공적인 ASA 가상 구축을 위해 Rackspace 클라우드가 활성화하는 구성 매개변수에 대한 설명은 Rackspace Day 0 컨피그레이션, 166 페이지 항목을 검토하십시오.

단계 **1** Rackspace mycloud 포털에서 **SERVERS(**서버**)** > **CREATE RESOURCES(**리소스 생성**)** > **Cloud Server(**클라우드 서버**)**로 이동합니다.

단계 **2** **Create Server(**서버 생성**)** 페이지에서 **Server Details(**서버 세부 정보**)**를 입력합니다.

a) **Server Name(**서버 이름**)** 필드에 ASA 가상 머신의 이름을 입력합니다.

b) **Region(**지역**)** 드롭다운 목록에서 지역을 선택합니다.

단계 **3** **Image(**이미지**)**에서 **Linux/Appliances(Linux/**어플라이언스**)** > **ASAv** > **Version(**버전**)**을 선택합니다.

참고 새 ASA 가상을 구축할 때는 대부분 지원되는 최신 버전을 선택합니다.

단계 **4** **Flavor(**버전**)**에서 리소스 요구 사항에 맞는 **Flavor Class(**버전 클래스**)**를 선택합니다. 적절한 VM 목록은 표 20: Rackspace 지원 버전, 163 페이지를 참조하십시오.

중요 9.13(1)부터 ASA 가상의 최소 메모리 요구 사항은 2GB입니다. 2개 이상의 vCPU로 ASA 가상을 구축할 경우 ASA 가상의 최소 메모리 요구 사항은 4GB입니다.

단계 **5** (선택 사항) **Advanced Options(**고급 옵션**)**에서 SSH 키를 구성합니다.

Rackspace 클라우드의 SSH 키에 대한 자세한 내용은 SSH 키로 액세스 관리를 참조하십시오.

단계 **6** ASA 가상에 적용 가능한 **Recommended Installs(**권장 설치**)** 및 **Itemized Charges(**항목화된 요금**)**을 검토하고, **Create Server(**서버 생성**)**를 클릭합니다.

루트 관리자 비밀번호가 표시됩니다. 비밀번호를 복사한 다음 대화 상자를 닫습니다.

단계 **7** 서버가 생성되면 서버 세부 정보 페이지가 표시됩니다. 서버가 활성 상태로 표시될 때까지 기다립니다. 이 작업은 일반적으로 몇 분 정도 걸립니다.

다음에 수행할 작업

- ASA 가상에 연결합니다.
- SSH를 통해 입력 가능한 CLI 명령을 사용하여 컨피그레이션을 계속하거나 ASDM을 사용합니다. ASDM 액세스에 대한 지침은 ASDM 시작, 291 페이지을 참조하십시오.

CPU 사용량 및 보고

CPU Utilization(CPU 사용률) 보고서에는 지정된 시간 내에 사용된 CPU의 백분율이 요약되어 있습니다. 일반적으로 코어는 사용량이 적은 시간에는 총 CPU 용량의 약 30~40%, 사용량이 많은 시간에는 약 60~70%로 작동합니다.

ASA Virtual의 vCPU 사용량

ASA virtual vCPU 사용량에서는 데이터 경로, 제어 지점, 외부 프로세스에 사용된 vCPU 양을 확인할 수 있습니다.

Rackspace에서 보고하는 vCPU 사용량에는 앞서 설명한 ASA virtual 사용량과 함께 다음 항목도 포함되어 있습니다.

- ASA virtual 유휴 시간
- ASA 가상 머신에 사용된 %SYS 오버헤드
- vSwitch, vNIC, pNIC 간 패킷 이동의 오버헤드. 이 오버헤드가 상당히 클 수 있습니다.

CPU 사용량의 예

show cpu usage 명령을 사용하여 CPU 사용률 통계를 표시할 수 있습니다.

예

```
Ciscoasa#show cpu usage

CPU utilization for 5 seconds = 1%; 1 minute: 2%; 5 minutes: 1%
```

다음은 보고된 vCPU 사용량이 상당한 차이를 보이는 예입니다.

- ASA Virtual 보고서: 40%
- DP: 35%
- 외부 프로세스: 5%
- ASA(ASA Virtual 보고서): 40%
- ASA 유휴 폴링: 10%
- 오버헤드: 45%

이 오버헤드는 하이퍼바이저 기능을 수행하고 vSwitch를 사용하여 NIC와 vNIC 간에 패킷을 이동하는 데 사용됩니다.

Rackspace CPU 사용량 보고

사용 가능한 클라우드 서버에 대한 CPU, RAM 및 디스크 공간 컨피그레이션 정보는 물론 디스크, I/O 및 네트워킹 정보도 볼 수 있습니다. 이 정보를 사용하여 요구 사항에 적합한 클라우드 서버를 결정하십시오. 명령줄 nova 클라이언트나 **Cloud Control Panel(**클라우드 제어판**)** 인터페이스를 통해 사용 가능한 서버를 볼 수 있습니다.

명령줄에서 다음 명령을 실행합니다.

```
nova flavor-list
```

사용 가능한 모든 서버 컨피그레이션이 표시됩니다. 목록에는 다음 정보가 포함됩니다.

- ID - 서버 컨피그레이션 ID
- Name - RAM 크기 및 성능 유형별로 레이블이 지정된 컨피그레이션의 이름
- Memory_MB - 컨피그레이션에 대한 RAM 크기
- Disk - GB 단위의 디스크 크기(범용 클라우드 서버의 경우 시스템 디스크의 크기)
- Ephemeral - 데이터 디스크의 크기
- Swap - 스왑 공간의 크기
- VCPUs - 컨피그레이션과 연결된 가상 CPU의 수
- RXTX_Factor - PublicNet 포트, ServiceNet 포트, 서버에 할당된 격리된 네트워크(클라우드 네트워크)에 할당된 대역폭(Mbps)
- Is_Public - 사용하지 않음

ASA Virtual 및 Rackspace 그래프

ASA Virtual과 Rackspace의 CPU % 수치가 다릅니다.

- Rackspace 그래프 수치가 항상 ASA Virtual 수치보다 높습니다.
- Rackspace에서는 이를 %CPU usage, ASA Virtual에서는 %CPU utilization이라고 부릅니다.

용어 “%CPU utilization”과 “%CPU usage”의 의미는 서로 다릅니다.

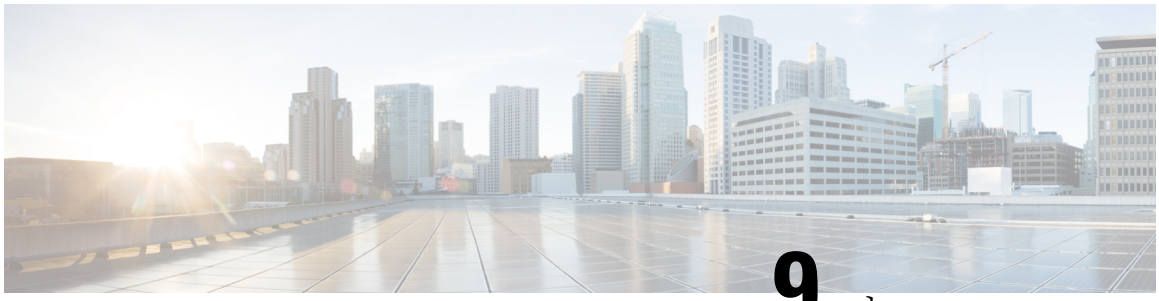
- CPU utilization은 물리적 CPU의 통계를 제공합니다.
- CPU usage는 논리적 CPU의 통계로서 CPU 하이퍼스레딩을 기반으로 합니다. 그러나 단 하나의 vCPU가 사용되므로 하이퍼스레딩은 켜져 있지 않습니다.

Rackspace는 %CPU usage를 다음과 같이 계산합니다.

활발하게 사용 중인 가상 CPU의 양 - 총 가용 CPU 기준 백분율로 표시

이 계산은 게스트 운영 체제가 아닌 호스트의 관점에서 본 CPU 사용량입니다. 그리고 가상 머신에 있는 사용 가능한 모든 가상 CPU의 평균 CPU 사용률입니다.

예를 들어, 가상 CPU 1개를 사용하는 가상 시스템이 4개의 물리적 CPU를 가진 호스트에서 실행되는 중이고 CPU usage가 100%라면 가상 머신에서 하나의 물리적 CPU를 온전히 사용하는 것입니다. 가상 CPU 사용량 계산: 사용량(MHz) / 가상 CPU 수 x 코어 주파수



9 장

Hyper-V를 사용하여 ASA 가상 구축

Microsoft Hyper-V를 사용하여 ASA 가상를 구축할 수 있습니다.

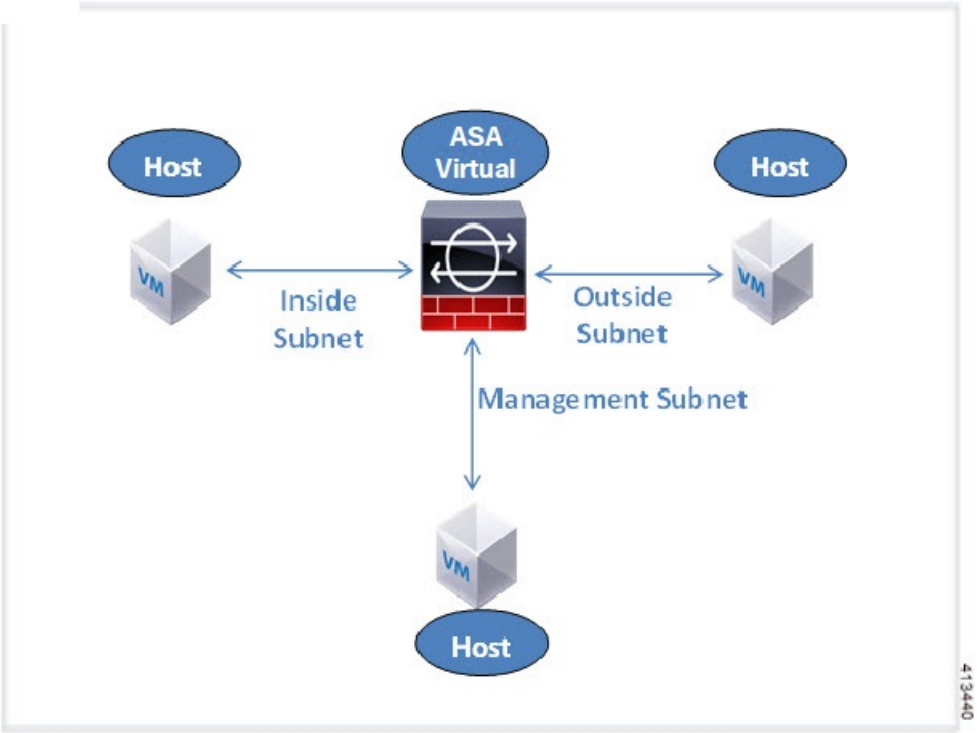
중요 9.13(1)부터 ASA 가상의 최소 메모리 요구 사항은 2GB입니다. 현재 ASA 가상이 2GB 미만의 메모리로 실행되는 경우에는 ASA 가상 머신의 메모리를 늘리지 않고는 이전 버전에서 9.13(1) 이상으로 업그레이드할 수 없습니다. 9.13(1) 버전의 새 ASA 가상 머신을 재구축할 수도 있습니다.

Hyper-V를 사용한 ASA 가상 구축 정보

독립형 Hyper-V 서버에 또는 Hyper-V Manager를 통해 Hyper-V를 구축할 수 있습니다. Powershell CLI 명령을 사용하여 설치하는 방법은 명령줄을 사용하여 Hyper-V에 ASA 가상를 설치(46페이지)를 참고하십시오. Hyper-V Manager를 사용하여 설치하는 방법은 Hyper-V Manager를 사용하여 Hyper-V에 ASA 가상 설치(46페이지)를 참고하십시오. Hyper-V는 시리얼 콘솔 옵션을 제공하지 않습니다. 관리 인터페이스에서 SSH 또는 ASDM을 통해 Hyper-V를 관리할 수 있습니다. SSH 설정 관련 정보는 SSH 구성(54페이지)을 참조하십시오.

다음 그림에서는 라우팅 방화벽 모드의 ASA 가상에 대한 권장 토폴로지를 확인할 수 있습니다. Hyper-V에 ASA 가상을 위한 3가지 서브넷(관리, 내부, 외부)이 설정되어 있습니다.

그림 37: 라우팅 방화벽 모드 ASA 가상의 권장 토폴로지



ASA 가상 및 Hyper-V에 대한 지침과 제한 사항

- 플랫폼 지원
  - Cisco UCS B-Series 서버
  - Cisco UCS C-Series 서버
  - Hewlett Packard Proliant DL160 Gen8
- OS 지원
  - Windows Server 2012
  - 기본 Hyper-V

참고 ASA 가상은 현재 가상화에 사용되는 최신 64비트 고성능 플랫폼에서 실행해야 합니다.

- 파일 형식

  Hyper-V에서의 ASA 가상 ASAv 초기 구축에 VHDX 형식을 지원합니다.

- Day 0 컨피그레이션

  필요한 ASA CLI 컨피그레이션 명령을 포함한 텍스트 파일을 생성합니다. 절차는 Day 0 컨피그레이션 파일 준비를 참조하십시오.

- Day 0 컨피그레이션의 방화벽 투명 모드

  컨피그레이션 줄 'firewall transparent'가 Day 0 컨피그레이션 파일의 맨 위에 있어야 합니다. 파일에서 다른 곳에 위치할 경우 잘못된 동작이 나올 수 있습니다. 절차는 Day 0 컨피그레이션 파일 준비를 참조하십시오.

- 장애 조치

  Hyper-V 기반 ASA 가상은 액티브/스탠바이 페일오버를 지원합니다. 라우팅 모드와 투명 모드에서 액티브/스탠바이 장애 조치를 구현하려면 모든 가상 네트워크 어댑터에서 MAC 주소 스푸핑을 활성화해야 합니다. 53페이지의 MAC 주소 스푸핑 구성을 참조하십시오. 독립형 ASA 가상의 투명 모드에서는 관리 인터페이스에서 MAC 주소 스푸핑을 활성화지 않아야 합니다. 액티브/액티브 장애 조치는 지원되지 않습니다.

- Hyper-V는 최대 8개의 인터페이스를 지원합니다. Management 0/0 및 GigabitEthernet 0/0 ~ 0/6입니다. GigabitEthernet을 장애 조치 링크로 사용할 수 있습니다.

- VLAN

  Set-VMNetworkAdapterVLan Hyper-V Powershell 명령을 사용하여 트렁크 모드의 인터페이스에 VLAN을 설정합니다. 관리 인터페이스에 대한 기본 VLAN ID를 특정 VLAN으로 또는 VLAN 없음을 의미하는 '0'으로 설정할 수 있습니다. 트렁크 모드는 Hyper-V 호스트 재부팅 시 유지되지 않습니다. 재부팅한 후 매번 트렁크 모드를 재구성해야 합니다.

- 레거시 네트워크 어댑터는 지원되지 않습니다.

- 2세대 가상 시스템은 지원되지 않습니다.

- Microsoft Azure는 지원되지 않습니다.

ASA 가상 및 Hyper-V 사전 요건

- MS Windows 2012에 Hyper-V를 설치합니다.

- Day 0 컨피그레이션 텍스트 파일을 생성합니다(사용 중인 경우).

  ASA 가상이 처음으로 구축되기 전에 Day 0 컨피그레이션을 추가해야 합니다. 그렇지 않으면 ASA 가상에서 write erase를 수행해야 Day 0 컨피그레이션을 사용할 수 있습니다. 절차는 Day 0 컨피그레이션 파일 준비를 참조하십시오.

- Cisco.com에서 ASA 가상 VHDX 파일을 다운로드합니다.

  http://www.cisco.com/go/asa-software

참고 Cisco.com 로그인 및 Cisco 서비스 계약이 필요합니다.

- 3개 이상의 서브넷/VLAN에 구성된 Hyper-V 스위치
- Hyper-V 시스템 요구 사항은 Cisco Secure Firewall ASA 호환성을 참조하십시오.

Day 0 컨피그레이션 파일 준비

ASA 가상를 실행하기 전에 Day 0 구성 파일을 준비할 수 있습니다. 이 파일은 ASA 가상을 시작할 때 적용하는 ASA 가상 컨피그레이션이 포함된 텍스트 파일입니다. 이 초기 컨피그레이션은 사용자가 선택하는 작업 디렉토리의 "day0-config"라는 이름의 텍스트 파일에 위치하며, 이 파일은 최초 부팅 시 마운트되고 읽히는 day0.iso 파일로 조작됩니다. Day 0 컨피그레이션 파일에는 최소한 관리 인터페이스를 활성화하고 공용 키 인증용 SSH 서버를 설정하는 명령이 포함되어야 할 뿐만 아니라, 완전한 ASA 컨피그레이션도 포함되어야 합니다. 최초 부팅 동안 day0.iso 파일(사용자 정의 day0.iso 또는 기본 day0.iso)을 사용할 수 있어야 합니다.

시작하기 전에

이 예에서는 Linux를 사용하지만 Windows에도 유사한 유틸리티가 있습니다.

- 초기 구축 동안 ASA 가상 라이센스를 자동으로 적용하려면, Cisco Smart Software Manager에서 다운로드한 Smart Licensing ID(Identity) Token을 Day 0 컨피그레이션 파일과 같은 디렉토리에 있는 'idtoken'이라는 이름의 텍스트 파일로 가져옵니다.
- 투명 모드에서 ASA 가상을 구축하려는 경우, 투명 모드에서 실행 중인 알려진 ASA 컨피그레이션 파일을 Day 0 컨피그레이션 파일로 사용해야 합니다. 이 사항은 라우팅 방화벽용 Day 0 컨피그레이션 파일에는 적용되지 않습니다.
- 처음으로 ASA 가상을 부팅하기 전에 Day 0 컨피그레이션 파일을 추가해야 합니다. 처음으로 ASA 가상을 부팅한 후에 Day 0 컨피그레이션을 사용하기로 결정할 경우 **write erase** 명령을 실행하고 Day 0 컨피그레이션 파일을 적용한 다음 ASA 가상을 부팅해야 합니다.

단계 **1** "day0 config"라는 텍스트 파일에 ASA 가상에 대한 CLI 컨피그레이션을 입력합니다. 3개의 인터페이스에 대한 인터페이스 컨피그레이션 및 원하는 기타 모든 컨피그레이션을 추가합니다.

첫 줄은 ASAv 버전으로 시작해야 합니다. day0-config는 유효한 ASA 컨피그레이션이어야 합니다. day0-config를 생성하는 가장 좋은 방법은 기존 ASA 또는 ASA 가상에서 실행 중인 컨피그레이션 중 원하는 부분을 복사하는 것입니다. day0-config에서 줄의 순서가 중요하며 기존 show run 명령 출력의 순서와 일치해야 합니다.

예제:

```
ASA Version 9.5.1
!
interface management0/0
nameif management
 security-level 100
```

```
 ip address 192.168.1.2 255.255.255.0
 no shutdown
interface gigabitethernet0/0
 nameif inside
 security-level 100
 ip address 10.1.1.2 255.255.255.0
 no shutdown
interface gigabitethernet0/1
 nameif outside
 security-level 0
 ip address 198.51.100.2 255.255.255.0
 no shutdown
http server enable
http 192.168.1.0 255.255.255.0 management
crypto key generate rsa modulus 1024
username AdminUser password paSSw0rd
ssh 192.168.1.0 255.255.255.0 management
aaa authentication ssh console LOCAL
```

단계 **2** (선택 사항) Cisco Smart Software Manager에서 발급한 스마트 라이센스 ID 토큰 파일을 컴퓨터에 다운로드합니다.

단계 **3** (선택 사항) 다운로드 파일에서 ID 토큰을 복사하고 ID 토큰만 포함된 텍스트 파일에 붙여넣습니다.

단계 **4** (선택 사항) 초기 ASA 가상 구축 동안 라이센싱이 자동으로 이루어진 경우, day0-config 파일에 다음 정보가 포함되어 있는지 확인합니다.

- 관리 인터페이스 IP 주소
- (선택 사항) Smart Licensing에 사용할 HTTP 프록시
- HTTP 프록시(지정된 경우) 또는 tools.cisco.com에 대한 연결을 지원하는 route 명령
- tools.cisco.com을 IP 주소에 확인하는 DNS 서버
- 사용자가 요청하는 ASA 가상 라이센스를 지정하는 Smart Licensing 컨피그레이션
- (선택 사항) ASA 가상이 CSSM에서 검색을 더욱 쉽게 수행할 수 있도록 하는 고유한 호스트 이름

단계 **5** 텍스트 파일을 ISO 파일로 전환하여 가상 CD-ROM을 생성합니다.

```
stack@user-ubuntu:-/KvmAsa$ sudo genisoimage -r -o day0.iso day0-config idtoken
I: input-charset not specified, using utf-8 (detected in locale settings)
Total translation table size: 0
Total rockridge attributes bytes: 252
Total directory bytes: 0
Path table size (byptes): 10
Max brk space used 0
176 extents written (0 MB)
stack@user-ubuntu:-/KvmAsa$
```

ID 토큰은 ASA 가상을 Smart Licensing 서버에 자동으로 등록합니다.

단계 **6** 1단계~5단계를 반복하여 구축하려는 각 ASA 가상에 대해 적절한 IP 주소가 포함된 별도의 기본 컨피그레이션 파일을 만듭니다.

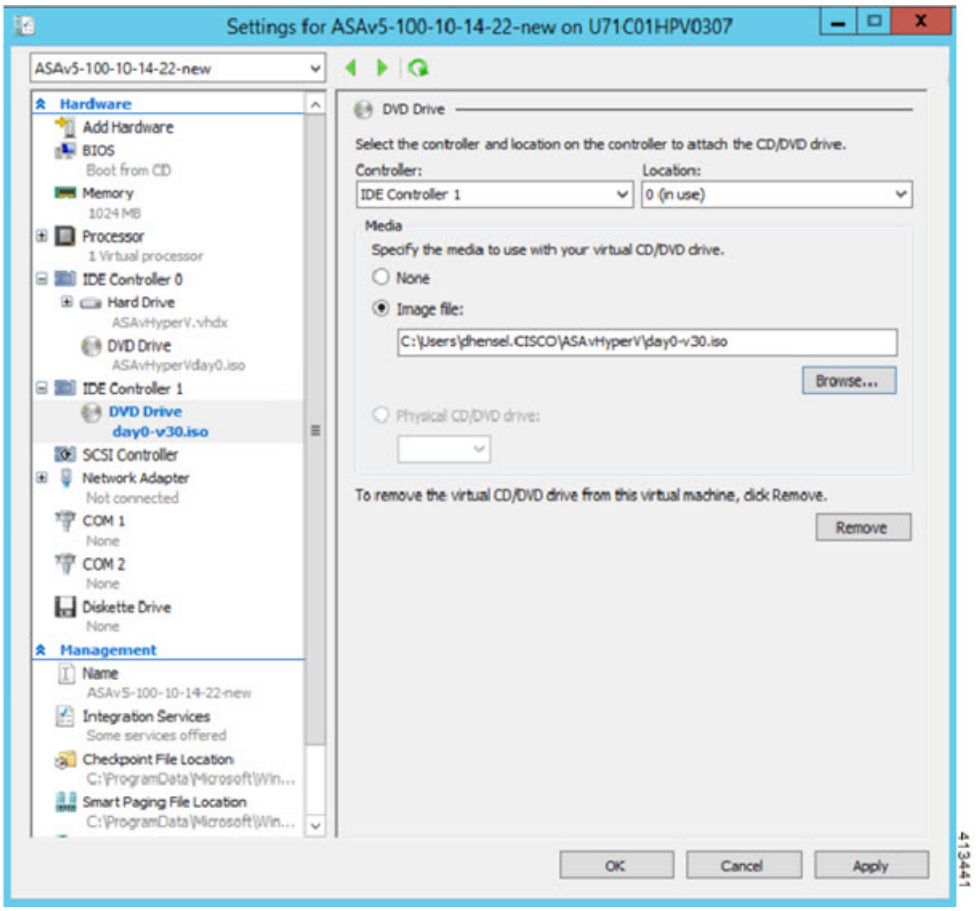
Hyper-V Manager를 사용하여 Day 0 컨피그레이션 파일로 ASA 가상 구축

Day 0 컨피그레이션 파일(Day 0 컨피그레이션 파일 준비)을 설정했다면 Hyper-V Manager를 사용하여 구축할 수 있습니다.

단계 **1** **Server Manager(**서버 관리자**)** > **Tools(**툴**)** > **Hyper-V Manager**로 이동합니다.

단계 **2** Hyper-V Manager의 오른쪽에 있는 **Settings(**설정**)**를 클릭합니다. Settings(설정) 대화 상자가 열립니다. 왼쪽의 **Hardware(**하드웨어**)**에서 **IDE Controller 1(IDE** 컨트롤러 **1)**을 클릭합니다.

그림 ***38: Hyper-V Manager***



단계 **3** 오른쪽 창의 **Media(**미디어**)**에서 **Image file(**이미지 파일**)** 라디오 버튼을 선택한 다음 Day 0 ISO 컨피그레이션 파일을 저장한 디렉터리로 이동하고 **Apply(**적용**)**를 클릭합니다. ASA 가상을 처음으로 부팅하는 경우 Day 0 컨피그레이션 파일의 내용에 따라 구성됩니다.

명령줄을 사용하여 Hyper-V에 ASA 가상 설치

Windows Powershell 명령줄을 통해 Hyper-V에 ASA 가상을 설치할 수 있습니다. 독립형 Hyper-V 서버에 있다면 명령줄을 사용하여 Hyper-V를 설치해야 합니다.

단계 **1** Windows Powershell을 엽니다.

단계 **2** ASA 가상을 구축합니다.

예제:

```
new-vm -name $fullVMName -MemoryStartupBytes $memorysize -Generation 1 -vhdpath
C:\Users\jsmith.CISCO\ASAvHyperV\$ImageName.vhdx -Verbose
```

단계 **3** ASA 가상 모델에 따라 CPU 카운트를 기본값인 1에서 변경합니다.

예제:

```
set-vm -Name $fullVMName -ProcessorCount 4
```

단계 **4** (선택 사항) 인터페이스 이름을 의미 있는 이름으로 바꿉니다.

예제:

```
Get-VMNetworkAdapter -VMName $fullVMName -Name "Network Adapter" | Rename-vmNetworkAdapter -NewName
mgmt
```

단계 **5** (선택 사항) 네트워크에서 필요하다면 VLAN ID를 변경합니다.

예제:

```
Set-VMNetworkAdapterVlan -VMName $fullVMName -VlanId 1151 -Access -VMNetworkAdapterName "mgmt"
```

단계 **6** Hyper-V에서 변경 사항을 적용하도록 인터페이스를 새로고침합니다.

예제:

```
Connect-VMNetworkAdapter -VMName $fullVMName -Name "mgmt" -SwitchName 1151mgmtswitch
```

단계 **7** 내부 인터페이스를 추가합니다.

예제:

```
Add-VMNetworkAdapter -VMName $fullVMName -name "inside" -SwitchName 1151mgmtswitch
Set-VMNetworkAdapterVlan -VMName $fullVMName -VlanId 1552 -Access -VMNetworkAdapterName "inside"
```

단계 **8** 외부 인터페이스를 추가합니다.

예제:

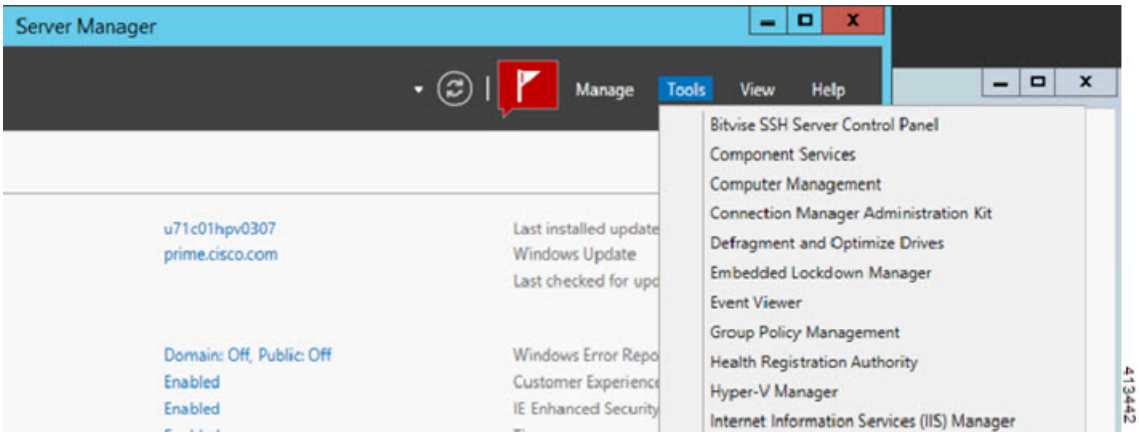
```
Add-VMNetworkAdapter -VMName $fullVMName -name "outside" -SwitchName 1151mgmtswitch
Set-VMNetworkAdapterVlan -VMName $fullVMName -VlanId 1553 -Access -VMNetworkAdapterName “outside"
```

Hyper-V Manager를 사용하여 Hyper-V에 ASA 가상 설치

Hyper-V Manager를 사용하여 Hyper-V에 ASA 가상을 설치할 수 있습니다.

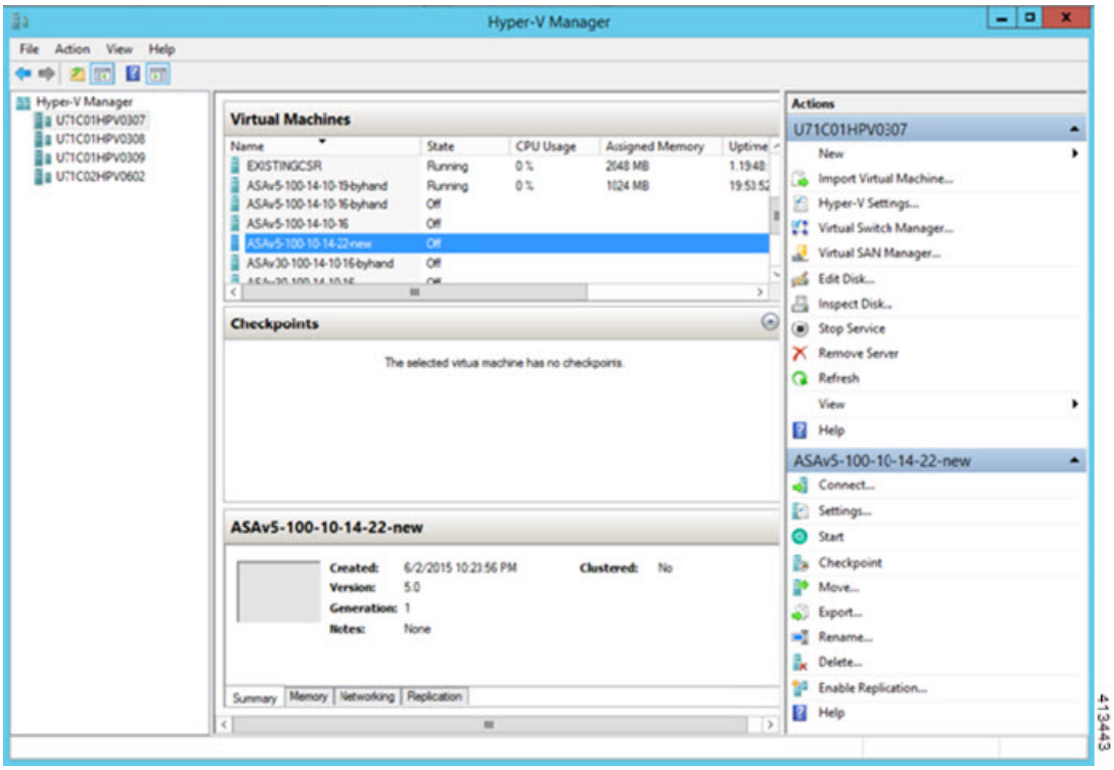
단계 **1** **Server Manager(**서버 관리자**)** > **Tools(**툴**)** > **Hyper-V Manager**로 이동합니다.

그림 ***39:*** 서버 관리자



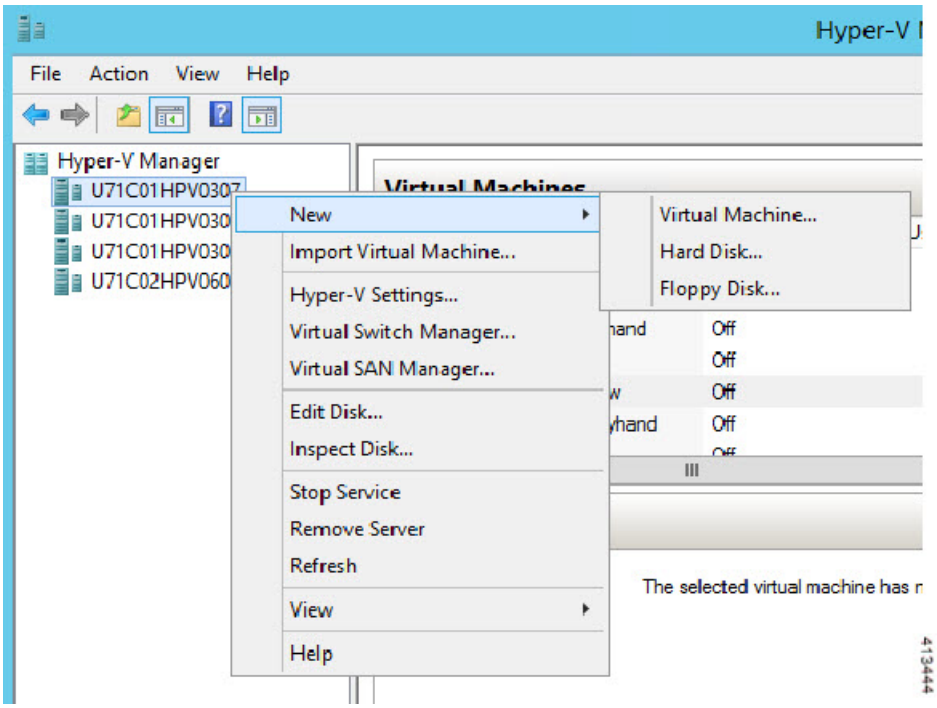
단계 **2** Hyper-V Manager가 나타납니다.

그림 *40: Hyper-V Manager*



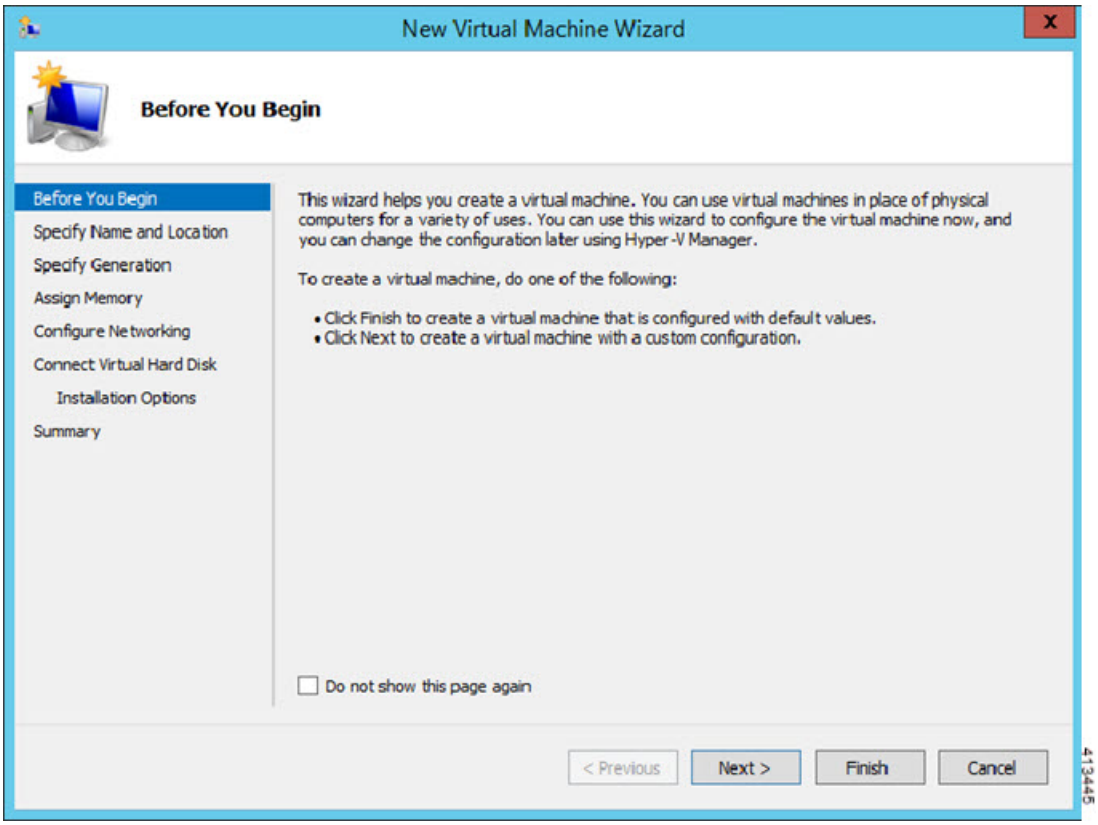
단계 **3** 오른쪽의 하이퍼바이저 목록에서 원하는 하이퍼바이저를 마우스 오른쪽 버튼으로 클릭하고 **New(**새로 만들기**)** > **Virtual Machine(**가상 머신**)**을 선택합니다.

그림 *41:* 새 가상 머신 시작



단계 **4** New Virtual Machine Wizard(새 가상 시스템 마법사)가 나타납니다.

그림 ***42: New Virtual Machine Wizard***(새 가상 머신 마법사)



단계 **5** 마법사에서 다음 정보를 지정합니다.

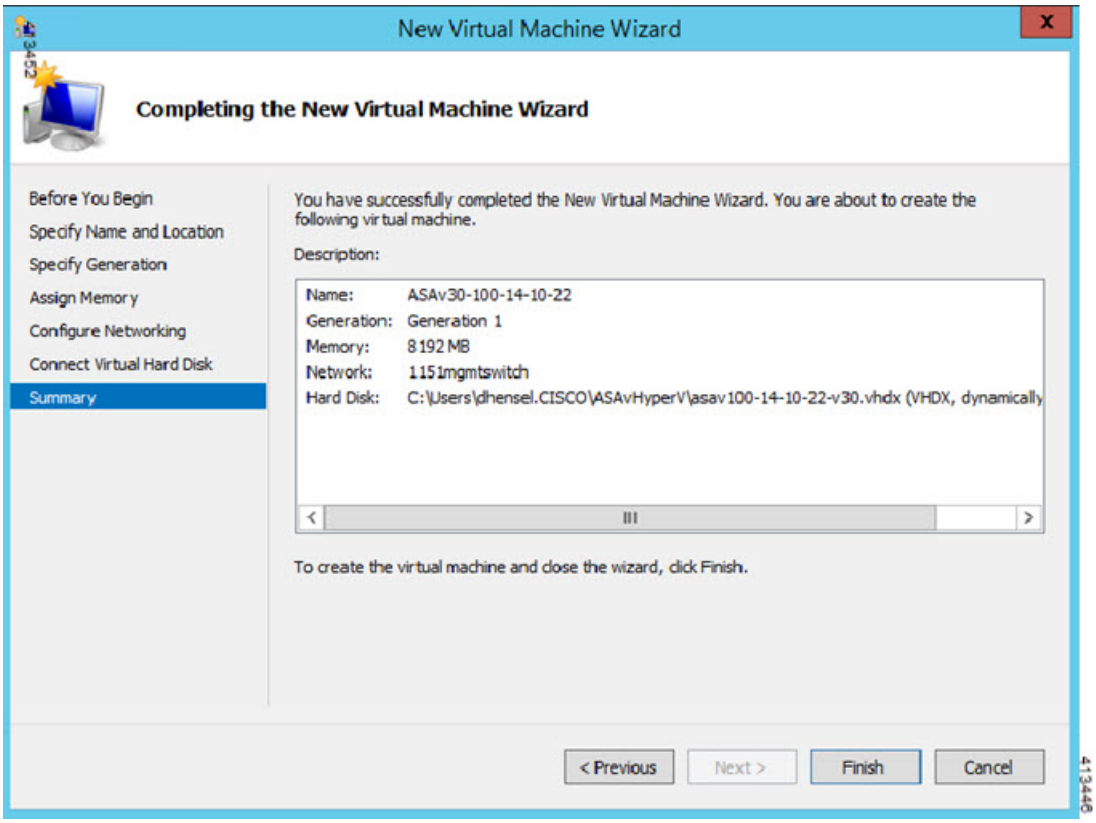
- ASA 가상의 이름 및 위치
- ASA 가상 생성

  ASA 가상에서 유일하게 지원되는 세대는 **Generation 1**입니다.
- ASA 가상의 메모리 용량 (100Mbps는 1024MB, 1Gbps는 2048MB, 2Gbps는 8192MB)
- 네트워크 어댑터(이미 설정한 가상 스위치에 연결)
- 가상 하드 디스크 및 위치

  Use an existing virtual hard disk(기존 가상 하드 디스크 사용**)**를 선택하고 VHDX 파일의 위치로 이동합니다.

단계 **6** Finish(마침)를 클릭하면 ASA 가상 컨피그레이션을 보여주는 대화 상자가 나타납니다.

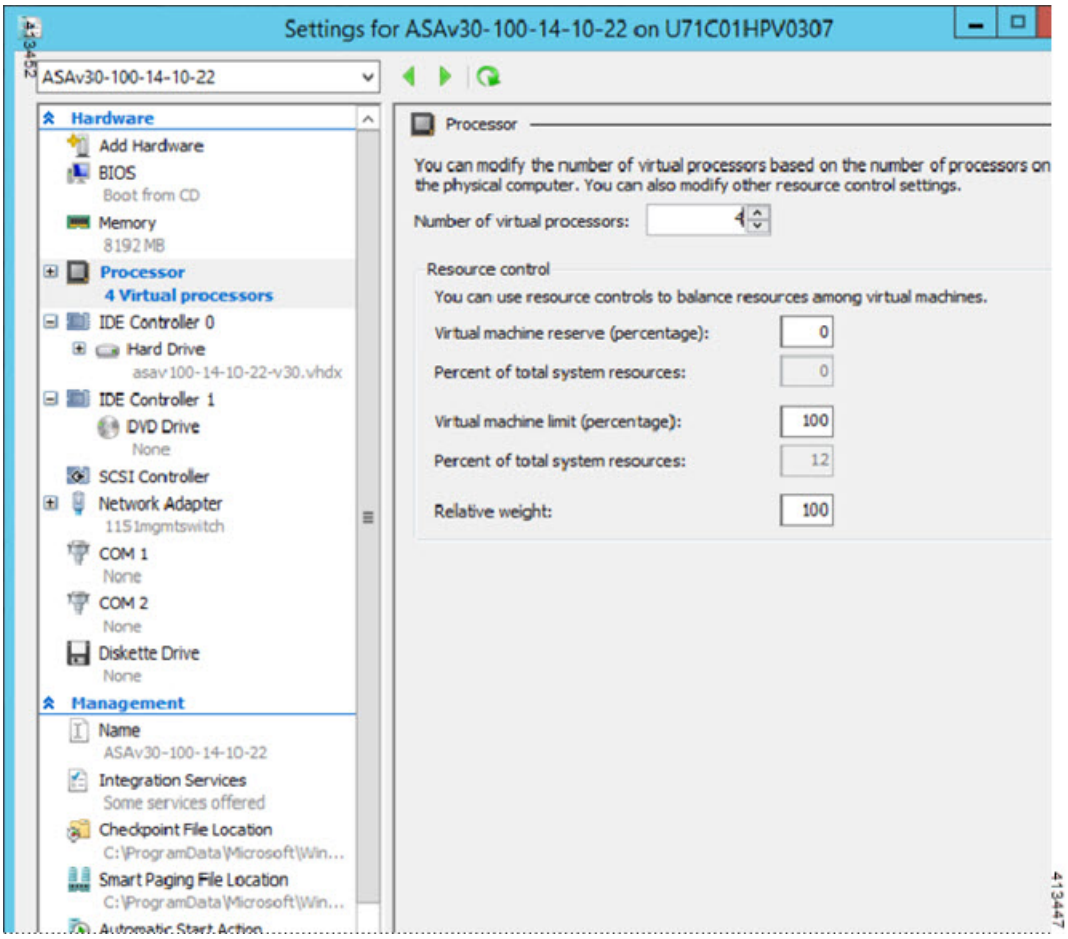
그림 *43:* 새 가상 머신 요약



단계 **7** ASA 가상에 vCPU 4개가 있다면, ASA 가상을 시작하기 전에 vCPU 값을 수정해야 합니다. Hyper-V Manager의 오른쪽에 있는 **Settings(**설정**)**를 클릭합니다. Settings(설정) 대화 상자가 열립니다. 왼쪽의 Hardware(하드웨어) 메뉴에서 **Processor(**프로세서**)**를 클릭하여 Processor(프로세서) 창으로 이동합니다. **Number of virtual processors(**가상 프로세서 수**)**를 4로 변경합니다.

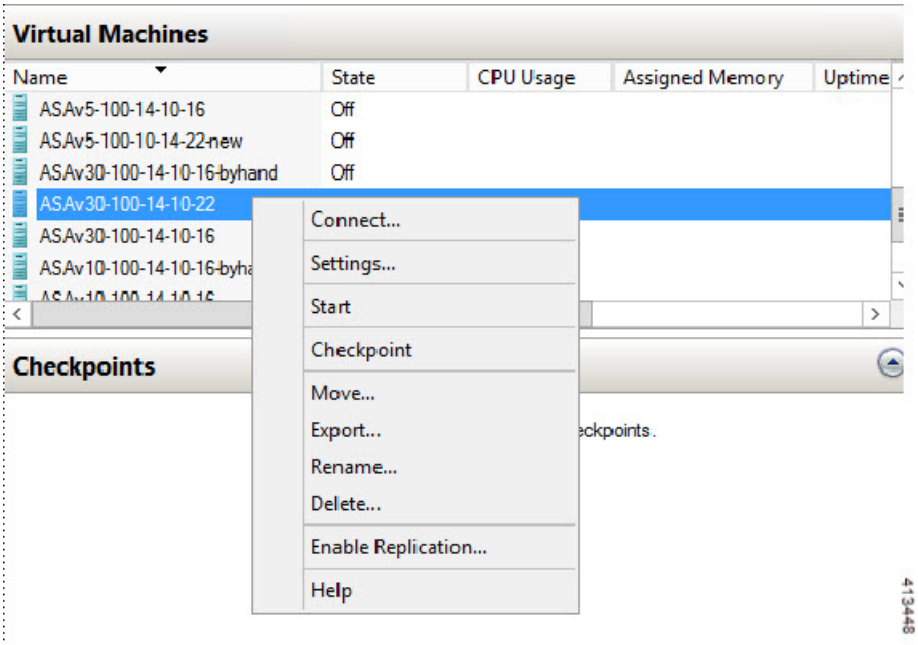
100Mbps 및 1Gbps 엔타이틀먼트에는 vCPU 1개가 있고 2Gbps 엔타이틀먼트에는 vCPU 4개가 있습니다. 기본값은 1입니다.

그림 *44:* 가상 머신 프로세서 설정



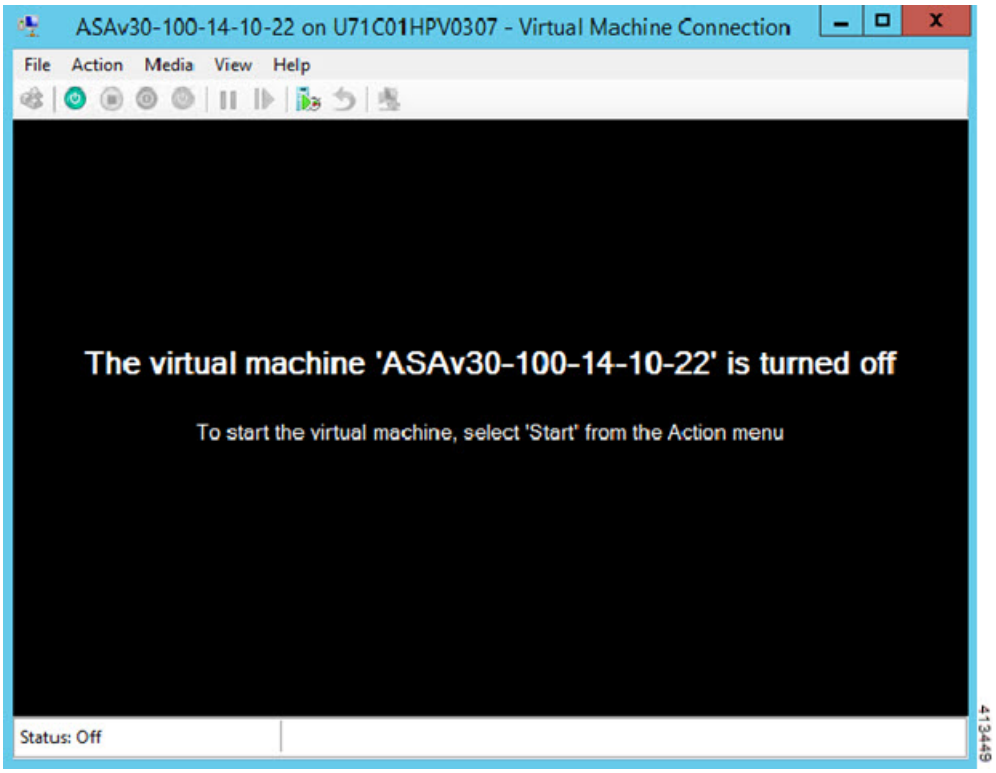
단계 **8** Virtual Machines(가상 시스템) 메뉴에서 목록의 ASA 가상 이름을 마우스 오른쪽 버튼으로 클릭하고 **Connect(**연결**)**를 클릭하여 ASA 가상에 연결합니다. 중지된 ASA 가상에 대한 콘솔이 열립니다.

그림 *45:* 가상 머신에 연결



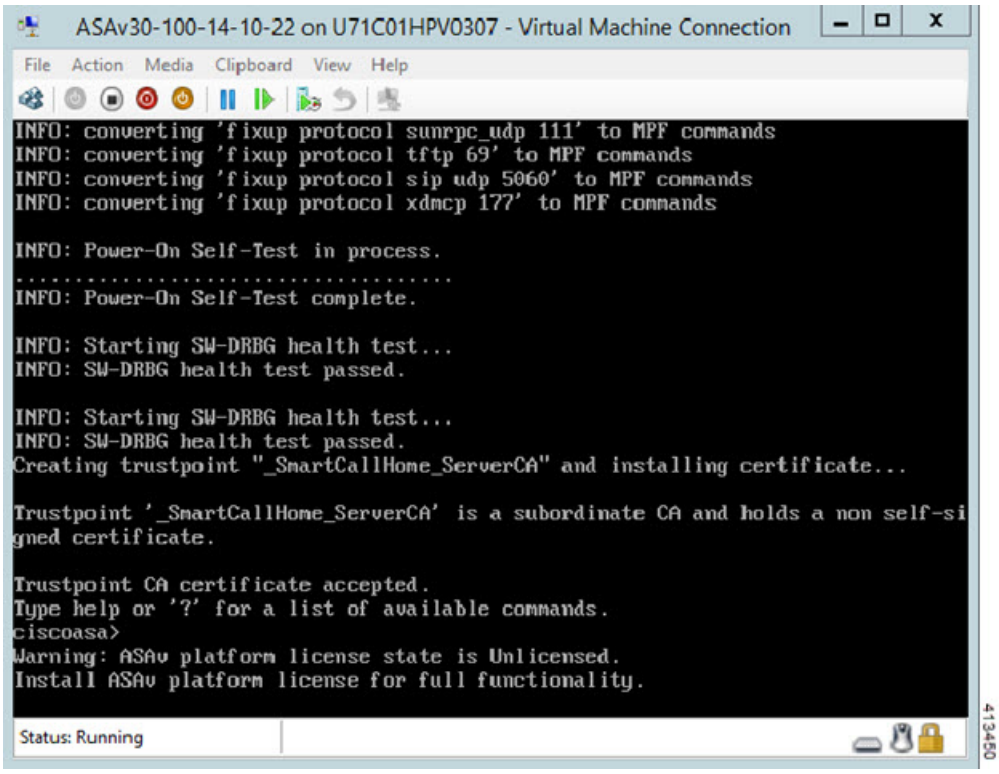
단계 **9** Virtual Machine Connection(가상 시스템 연결) 콘솔 창에서 청록색 Start(시작) 버튼을 클릭하여 ASA 가상을 시작합니다.

그림 *46:* 가상 머신 시작



단계 **10** ASA 가상의 부팅 진행 상황이 콘솔에 표시됩니다.

그림 **47:** 가상 머신 부팅 진행 중



Hyper-V Manager에서 네트워크 어댑터 추가

새로 구축된 ASA 가상에는 네트워크 어댑터가 1개뿐입니다. 네트워크 어댑터를 2개 이상 추가해야 합니다. 여기서는 내부 네트워크 어댑터를 추가하고 있습니다.

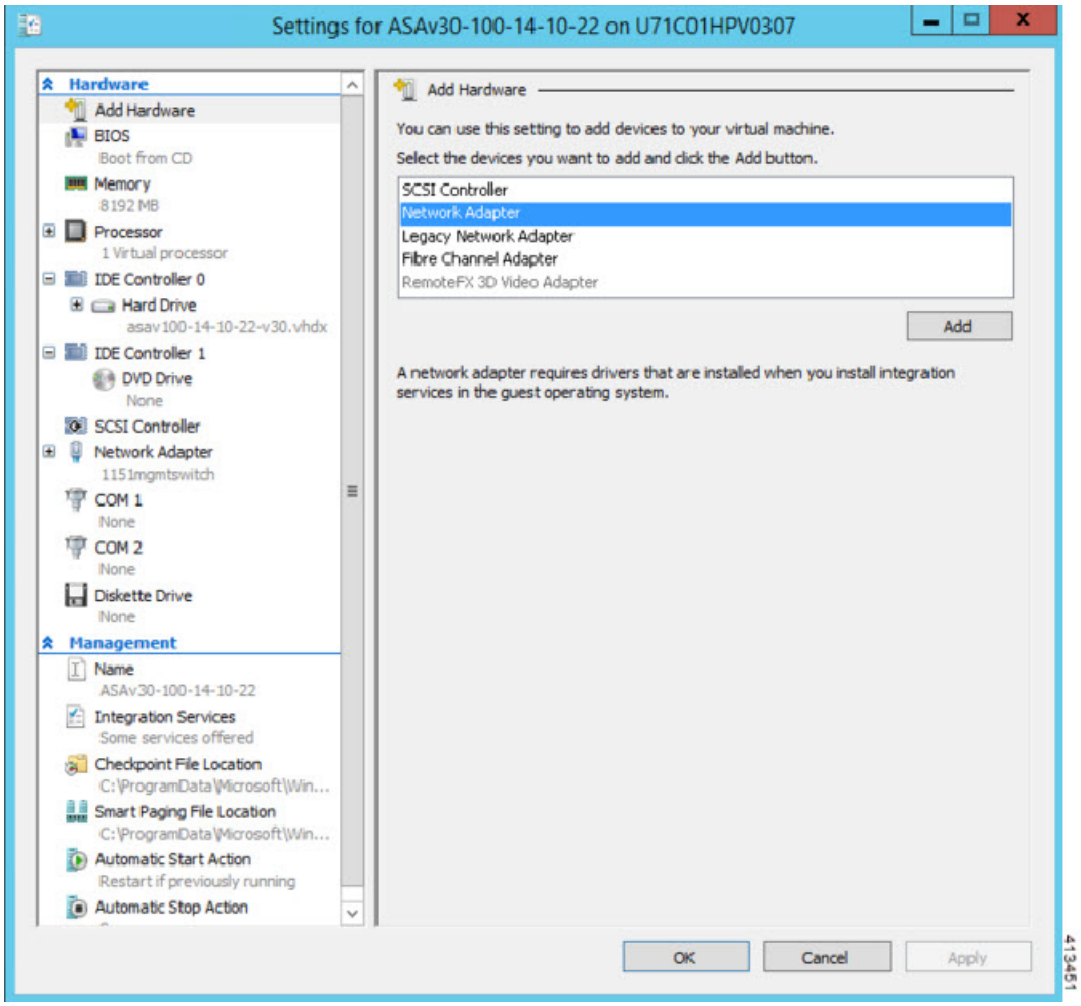
시작하기 전에

- ASA 가상이 꺼진 상태여야 합니다.

단계 **1** Hyper-V Manager의 오른쪽에 있는 **Settings(**설정**)**를 클릭합니다. Settings(설정) 대화 상자가 열립니다. 왼쪽의 Hardware(하드웨어) 메뉴에서 **Add Hardware(**하드웨어 추가**)**를 클릭하고 **Network Adapter(**네트워크 어댑터**)**를 클릭합니다.

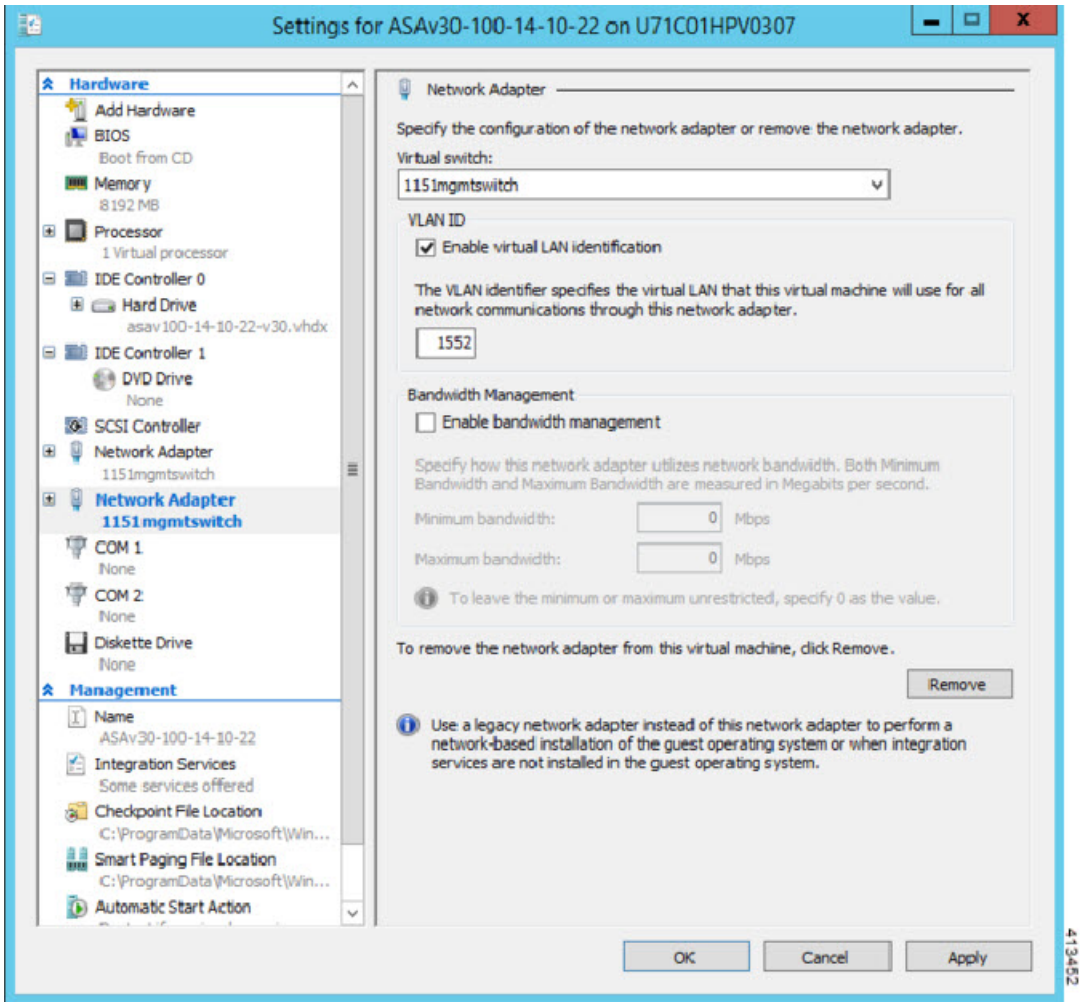
참고 레거시 네트워크 어댑터는 사용하지 마십시오.

그림 48: 네트워크 어댑터 추가



단계 2 네트워크 어댑터가 추가된 다음 가상 스위치 및 기타 기능을 수정할 수 있습니다. 필요하다면 여기서 VLAN ID도 설정할 수 있습니다.

그림 *49:* 네트워크 어댑터 설정 수정



네트워크 어댑터 이름 수정

Hyper-V에서는 일반 네트워크 인터페이스 이름인 'Network Adapter'가 사용됩니다. 네트워크 인터페이스가 모두 동일한 이름일 경우 혼동될 수 있습니다. Hyper-V Manager를 사용하여 이름을 수정할 수 없습니다. Windows Powershell 명령을 사용하여 수정해야 합니다.

단계 **1** Windows Powershell을 엽니다.

단계 **2** 필요에 따라 네트워크 어댑터를 수정합니다.

예제:

```
$NICRENAME= Get-VMNetworkAdapter -VMName 'ASAvVM' -Name "Network Adapter"
rename-VMNetworkAdapter -VMNetworkAdapter $NICRENAME[0] -newname inside
rename-VMNetworkAdapter -VMNetworkAdapter $NICRENAME[1] -newname outside
```

MAC 주소 스푸핑

ASA 가상이 투명 모드에서 패킷을 전달하려면 그리고 HA 액티브/스탠바이 페일오버를 위해서는 모든 인터페이스에 대해 MAC 주소 스푸핑을 활성화해야 합니다. 이는 Hyper-V Manager에서 또는 Powershell 명령을 사용하여 수행할 수 있습니다.

Hyper-V 관리자를 사용하여 MAC 주소 스푸핑 구성

Hyper-V Manager를 사용하여 Hyper-V에 MAC 스푸핑을 구성할 수 있습니다.

단계 **1** **Server Manager(**서버 관리자**)** > **Tools(**툴**)** > **Hyper-V Manager**로 이동합니다.

Hyper-V Manager가 나타납니다.

단계 **2** Hyper-V Manager의 오른쪽에 있는 **Settings(**설정**)**를 클릭하여 설정 대화상자를 엽니다.

단계 **3** 왼쪽의 **Hardware(**하드웨어**)** 메뉴에서 다음을 수행합니다.

1. **Inside(**내부**)**를 클릭하고 메뉴를 확장합니다.
2. **Advanced Features(**고급 기능**)**를 클릭하여 MAC 주소 옵션으로 이동합니다.
3. **Enable MAC address spoofing(MAC** 주소 스푸핑 활성화**)** 라디오 버튼을 클릭합니다.

단계 **4** Outside(외부) 인터페이스를 대상으로 같은 작업을 반복합니다.

명령줄을 사용하여 MAC 주소 스푸핑 구성

Windows Powershell 명령줄을 사용하여 Hyper-V에서 MAC 스푸핑을 구성할 수 있습니다.

단계 **1** Windows Powershell을 엽니다.

단계 **2** MAC 주소 스푸핑을 구성합니다.

예제**:**

```
Set-VMNetworkAdapter -VMName $vm_name\
-ComputerName $computer_name -MacAddressSpoofing On\
-VMNetworkAdapterName $network_adapter\r"
```

SSH 구성

Hyper-V Manager의 Virtual Machine Connection(가상 시스템 연결)에서 관리 인터페이스를 통한 SSH 액세스를 위해 ASA 가상을 구성할 수 있습니다. Day 0 컨피그레이션 파일을 사용하는 경우 여기에 SSH 액세스를 추가할 수 있습니다. 자세한 내용은 Day 0 컨피그레이션 파일 준비를 참조하십시오.

단계 **1** RSA 키 쌍이 있음을 확인합니다.

예제:

```
asav# show crypto key mypubkey rsa
```

단계 **2** RSA 키 쌍이 없을 경우 RSA 키 쌍을 생성합니다.

예제:

```
asav(conf t)# crypto key generate rsa modulus 2048

username test password test123 privilege 15
aaa authentication ssh console LOCAL
ssh 10.7.24.0 255.255.255.0 management
ssh version 2
```

단계 **3** 다른 PC에서 SSH를 사용하여 ASA 가상에 액세스할 수 있는지 확인합니다.

CPU 사용량 및 보고

CPU Utilization(CPU 사용률) 보고서에는 지정된 시간 내에 사용된 CPU의 백분율이 요약되어 있습니다. 일반적으로 코어는 사용량이 적은 시간에는 총 CPU 용량의 약 30~40%, 사용량이 많은 시간에는 약 60~70%로 작동합니다.

ASA Virtual의 vCPU 사용량

ASA virtual vCPU 사용량에서는 데이터 경로, 제어 지점, 외부 프로세스에 사용된 vCPU 양을 확인할 수 있습니다.

Hyper-V에서 보고하는 vCPU 사용량에는 앞서 설명한 ASA virtual 사용량과 함께 다음 항목도 포함되어 있습니다.

- ASA Virtual 유휴 시간
- ASA 가상 머신에 사용된 %SYS 오버헤드

CPU 사용량의 예

show cpu usage 명령을 사용하여 CPU 사용률 통계를 표시할 수 있습니다.

예

```
Ciscoasa#show cpu usage

CPU utilization for 5 seconds = 1%; 1 minute: 2%; 5 minutes: 1%
```

다음은 보고된 vCPU 사용량이 상당한 차이를 보이는 예입니다.

- ASA Virtual 보고서: 40%
- DP: 35%
- 외부 프로세스: 5%
- ASA(ASA Virtual 보고서): 40%
- ASA 유휴 폴링: 10%
- 오버헤드: 45%

10 장

Oracle Cloud Infrastructure에 ASA 가상 구축

OCI(Oracle Cloud Infrastructure)에 ASA 가상을 구축할 수 있습니다.

OCI에 ASA 가상 구축 정보

OCI는 Oracle에서 제공하는 고가용성 호스팅 환경에서 애플리케이션을 실행할 수 있는 퍼블릭 클라우드 컴퓨팅 서비스입니다.

ASA 가상은 물리적 ASA 가상과 동일한 소프트웨어를 실행하여 가상 폼 팩터에서 검증된 보안 기능을 제공합니다. ASA 가상은 퍼블릭 OCI에서 구축될 수 있습니다. 그러면 시간이 경과함에 따라 해당 위치를 확장, 축소 또는 이동하는 가상 및 물리적 데이터 센터 워크로드를 보호하기 위한 구성이 가능하게 됩니다.

OCI 컴퓨팅 셰이프

셰이프는 인스턴스 수에 할당되는 CPU 수, 메모리 양 및 기타 리소스를 결정하는 템플릿입니다. ASA 가상은 다음 표준 – 범용 OCI 셰이프 유형을 지원합니다.

표 *21:* 지원되는 컴퓨팅 셰이프 *ASA* 가상

가상 머신 셰이프	특성		인터페이스
	oCPU	메모리**(GB)**	
VM.Standard2.4	4	60	최소 3, 최대 4
VM.Standard2.8	8	120	최소 3개, 최대 8개

- ASA 가상에는 최소 3 개의 인터페이스가 필요합니다.
- OCI에서 1 oCPU는 vCPU 2개와 같습니다.
- 지원되는 최대 vCPU는 16개(8 oCPU)입니다.

OCI에서 계정을 생성하고, Oracle Cloud Marketplace에서 Cisco ASA 가상 방화벽(ASA 가상) 제품을 사용하여 컴퓨팅 인스턴스를 실행한 다음 OCI 셰이프를 선택합니다.

ASA 가상 및 OCI의 사전 요건

- https://www.oracle.com/cloud/sign-in.html에서 계정을 만듭니다.
- ASA 가상에 라이선스를 부여합니다. ASA 가상 라이선스를 등록하기 전에는 저성능 모드에서 실행됩니다. 이 모드에서는 100개의 연결 및 100Kbps의 처리량만 허용됩니다. 라이선스: 스마트 소프트웨어 라이선싱을 참조하십시오.
- 인터페이스 요건:
  - 관리 인터페이스
  - 내부 및 외부 인터페이스
  - (선택 사항) 추가 서브넷(DMZ)
- 통신 경로:
  - 관리 인터페이스 - ASA 가상을 ASDM에 연결할 때 사용합니다. 통과 트래픽에는 사용할 수 없습니다.
  - 내부 인터페이스(필수)—ASA 가상를 내부 호스트에 연결하는 데 사용합니다.
  - 외부 인터페이스(필수)—ASA 가상를 공용 네트워크에 연결하는 데 사용합니다.
  - DMZ 인터페이스(선택 사항)—ASA 가상을 DMZ 네트워크에 연결하는 데 사용합니다.
- ASA 가상 시스템 요구 사항은 Cisco Secure Firewall ASA호환성을 참조하십시오.

ASA 가상 및 OCI에 대한 지침 및 제한 사항

지원 기능

OCI의 ASA 가상은 다음 기능을 지원합니다.

- OCI VCN(Virtual Cloud Network)에 구축
- 인스턴스당 최대 16개의 vCPUs(8 oCPU)
- 라우팅 모드(기본값)

- 라이선싱 – BYOL만 지원됩니다.
- SR-IOV)Single Root I/O Virtualization)이 지원됩니다.

지원되지 않는 기능

OCI ASA 가상은 다음을 지원하지 않습니다.

- ASA 가상 기본 HA
- 투명 / 인라인 / 패시브 모드
- 멀티컨텍스트 모드

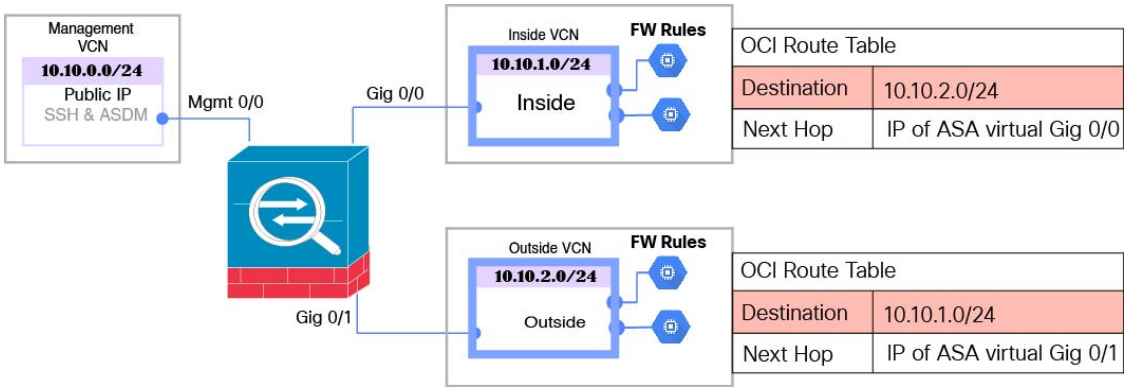
제한 사항

- OCI에서의 ASA 가상 구축은 SR-IOV 모드에서 Merlanox 5를 vNIC로 지원하지 않습니다.
- 고정 및 DHCP 구성 모두에 대해 ASAv에 별도의 라우팅 규칙이 필요합니다.

OCI 기반 ASA 가상의 샘플 네트워크 토폴로지

다음 그림은 Routed Firewall Mode의 ASA 가상에 대한 권장 네트워크 토폴로지와 ASA 가상에 대해 OCI에 구성된 3개의 서브넷(관리, 내부 및 외부)을 보여줍니다.

그림 *50: OCI* 구축에 대한 *ASA* 가상 샘플



OCI에 ASA 가상 구축

다음 절차에서는 OCI 환경을 준비하고 ASA 가상 인스턴스를 시작하는 방법을 설명합니다. OCI 포털에 로그인하고 OCI Marketplace에서 Cisco ASA 가상 방화벽(ASA 가상) 제품을 검색한 다음 컴퓨팅 인스턴스를 시작합니다. ASA 가상을 시작한 후에는 트래픽의 소스 및 대상에 따라 트래픽을 방화벽으로 전달하도록 경로 테이블을 구성해야 합니다.

VCN(Virtual Cloud Network) 생성

ASA 가상 구축을 위해 VCN(Virtual Cloud Network)을 구성합니다. 각 ASA 가상 인터페이스에 하나씩 최소 3개의 VCN이 필요합니다.

다음 절차를 계속 진행하여 관리 VCN을 완료할 수 있습니다. 그런 다음 **Networking(**네트워킹**)**으로 돌아가 내부 및 외부 인터페이스에 대한 VCN을 생성합니다.

시작하기 전에

참고 탐색 메뉴에서 서비스를 선택하면 왼쪽의 메뉴에 컴파트먼트 목록이 포함됩니다. 컴파트먼트는 리소스에 대한 액세스를 보다 쉽게 제어할 수 있도록 구성하는 데 도움이 됩니다. 테넌시가 프로비저닝되면 루트 컴파트먼트가 Oracle에 의해 생성됩니다. 관리자는 루트 컴파트먼트에서 더 많은 컴파트먼트를 생성한 다음 액세스 규칙을 추가하여 어떤 사용자가 보고 액세스할 수 있는지 제어할 수 있습니다. 자세한 내용은 Oracle 문서 "Managing Compartments"을 참조하십시오.

단계 **1** OCI에 로그인하고 지역을 선택합니다.

OCI는 여러 지역으로 나뉘며, 이 지역은 상호 격리되어 있습니다. 화면의 우측 상단에 지역이 표시됩니다. 한 지역의 리소스가 다른 지역에는 나타나지 않습니다. 원하는 지역에 있는지 정기적으로 확인합니다.

단계 **2** **Networking(**네트워킹**)** > **Virtual Cloud Networks(**가상 클라우드 네트워크**)**를 선택하고 Create Virtual Cloud Networks(가상 클라우드 네트워크 생성)를 클릭합니다.

단계 **3** VCN을 설명하는 **Name(**이름**)**(예: *ASAvManagement*)을 입력합니다.

단계 **4** VCN의 **CIDR** 블록을 입력합니다.

단계 **5** **Create VCN(VCN** 생성**)**을 클릭합니다.

네트워크 보안 그룹 생성

네트워크 보안 그룹은 vNIC에 적용되는 vNIC 집합과 보안 규칙 집합으로 구성됩니다.

단계 **1** **Networking(**네트워킹**)** > **Virtual Cloud Networks(**가상 클라우드 네트워크**)** > **Virtual Cloud Network Details(**가상 클라우드 네트워크 세부 사항**)** > **Network Security Groups(**네트워크 보안 그룹**)**를 선택하고 **Create Network Security Group(**네트워크 보안 그룹 생성**)**을 클릭합니다.

단계 **2** 네트워크 보안 그룹을 설명하는 **Name(**이름**)**을 입력합니다(예: *ASAv-Mgmt-Allow-22-443*).

단계 **3** **Next(**다음**)**를 클릭합니다.

단계 **4** 보안 규칙을 추가합니다.

a) SSH 액세스를 위해 TCP 포트 22를 허용하는 규칙을 ASA 가상 콘솔에 추가합니다.

b) HTTPS 액세스를 위해 TCP 포트 443을 허용하는 규칙을 ASDM에 추가합니다.

ASA 가상은 ASDM을 통해 관리할 수 있으며, 이 경우 HTTPS 연결을 위해 포트 443를 열어야 합니다.

단계 **5** **Create(**생성**)**를 클릭합니다.

인터넷 게이트웨이 생성

관리 서브넷에 액세스를 개방하려면 인터넷 게이트웨이가 필요합니다.

단계 **1** **Networking(**네트워킹**)** > **Virtual Cloud Networks(**가상 클라우드 네트워크**)** > **Virtual Cloud Network Details(**가상 클라우드 네트워크 세부 사항**)** > **Internet Gateways(**인터넷 게이트웨이**)**를 선택하고 **Create Internet Gateway(**인터넷 게이트웨이 생성**)**를 클릭합니다.

단계 **2** 인터넷 게이트웨이를 설명하는 **Name(**이름**)**을 입력합니다(예: *ASAv-IG*).

단계 **3** **Create Internet Gateway(**인터넷 게이트웨이 생성**)**를 클릭합니다.

단계 **4** 인터넷 게이트웨에 라우트 추가

a) **Networking(**네트워킹**)** > **Virtual Cloud Networks(**가상 클라우드 네트워크**)** > **Virtual Cloud Network Details(**가상 클라우드 네트워크 세부 사항**)** > **Route Tables(**라우트 테이블**)**를 선택합니다.

b) 경로 규칙을 추가하려면 기본 경로 테이블에 대한 링크를 클릭합니다.

c) 경로 규칙 추가를 클릭합니다.

d) **Target Type(**대상 유형**)** 드롭 다운에서 **Internet Gateway(**인터넷 게이트웨이**)**를 선택합니다.

e) 대상 IPv4 CIDR 블록을 입력합니다(예: 0.0.0.0/0).

f) **Target Internet Gateway(**대상 인터넷 게이트웨이**)** 드롭 다운에서 생성한 게이트웨이를 선택합니다.

g) 경로 규칙 추가를 클릭합니다.

서브넷 생성

각 VCN에는 최소한 하나의 서브넷이 있습니다. 관리 VCN에 대한 관리 서브넷을 생성합니다. 또한 내부 VCN에는 내부 서브넷이 필요하고 외부 VCN에는 외부 서브넷이 필요합니다.

단계 **1** **Networking(**네트워킹**)** > **Virtual Cloud Networks(**가상 클라우드 네트워크**)** > **Virtual Cloud Network Details(**가상 클라우드 네트워크 세부 사항**)** > **Subnets(**서브넷**)**를 선택하고 **Create Subnet(**서브넷 생성**)**를 클릭합니다.

단계 **2** 서브넷을 설명하는 이름(예: *Management(*관리*)*)을 입력합니다.

단계 **3** 서브넷 유형을 선택합니다(권장 기본값 **Regional(**지역별**)**은 유지).

단계 **4** **CIDR** 블록을 입력합니다(예: 10.10.0.0/24). 서브넷의 내부(비 공용) IP 주소는 이 CIDR 블록에서 가져옵니다.

단계 **5** **Route Table(**경로 테이블**)** 드롭 다운에서 이전에 생성한 경로 테이블 중 하나를 선택합니다.

단계 **6** 서브넷의 서브넷 액세스를 선택합니다.

관리 서브넷의 경우 공용 서브넷이어야 합니다.

단계 **7** **DHCP** 옵션을 선택합니다.

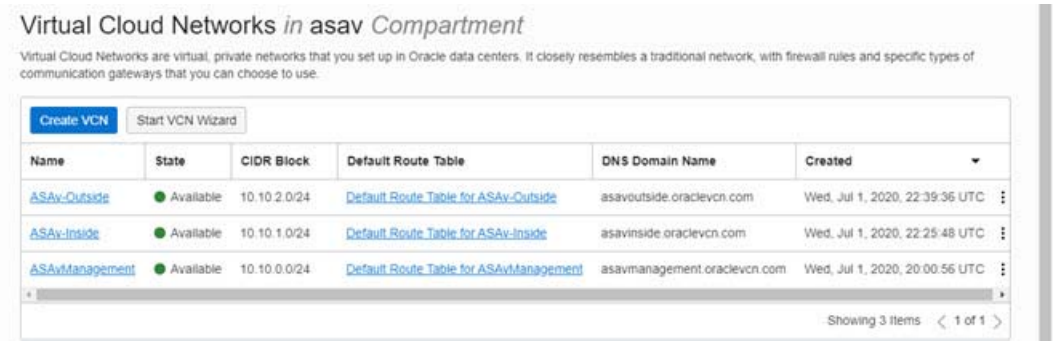
단계 **8** 이전에 생성한 보안 목록을 선택합니다.

단계 **9** **Create Subnet(**서브넷 생성**)**을 클릭합니다.

다음에 수행할 작업

VCN(관리, 내부, 외부)을 구성하고 나면 ASA 가상을 시작할 수 있습니다. ASA 가상 VCN 컨피그레이션의 예는 다음 그림을 참조하십시오.

그림 ***51: ASA*** 가상 클라우드 네트워크



OCI에서 ASA 가상 인스턴스 생성

Oracle Cloud Marketplace에서 제공하는 Cisco ASA 가상 방화벽(ASA 가상) 제품을 사용하여 컴퓨팅 인스턴스를 통해 OCI에 ASA 가상을 구축합니다. CPU 수, 메모리 양, 네트워크 리소스 등의 특성에 따라 가장 적합한 시스템 형태를 선택합니다.

단계 **1** OCI 포털에 로그인합니다.

화면의 우측 상단에 지역이 표시됩니다. 원하는 지역에 있는지 정기적으로 확인합니다.

단계 **2** **Marketplace(**마켓플레이스**)** > **Applications(**애플리케이션**)**을 선택합니다.

단계 **3** Marketplace에서 "Cisco ASA 가상 방화벽(ASAv)"을 검색하고 제품을 선택합니다.

단계 **4** 이용 약관을 검토하고, 제가 검토한 후 **Oracle** 이용 약관 및 파트너 이용 약관에 동의함 확인란을 선택합니다.

단계 **5** **Launch Instance(**인스턴스 실행**)**를 클릭합니다.

단계 **6** 인스턴스를 설명하는 **Name(**이름**)**(예: *ASAv-9-15*)을 입력합니다.

단계 **7** **Change Shape(**셰이프 변경**)**를 클릭하고 oCPU 수, RAM 크기, ASA 가상에 필요한 인터페이스 수를 포함하는 셰이프를 선택합니다(예: VM.Standard2.4(표 21: 지원되는 컴퓨팅 셰이프 ASA 가상, 193 페이지 참조)).

단계 **8** **Virtual Cloud Network(**가상 클라우드 네트워크**)** 드롭 다운에서 Management VCN (관리 VCN)을 선택합니다.

단계 **9** **Subnet(**서브넷**)** 드롭 다운에서 관리 서브넷이 자동으로 채워지지 않은 경우 선택합니다.

단계 **10** **Use Network Security Groups to Network Traffic(**트래픽을 제어하기 위해 네트워크 보안 그룹 사용**)**을 선택하고 관리 VCN에 대해 구성한 보안 그룹을 선택합니다.

단계 **11** **Assign a Public Ip Address(**공용 **IP** 주소 할당**)** 라디오 버튼을 클릭합니다.

단계 **12** **Add SSH keys(SSH** 키 추가**)**에서 **Paste Public Keys(**공개 키 붙여 넣기**)** 라디오 버튼을 클릭하고 SSH 키를 붙여 넣습니다.

Linux 기반 인스턴스는 비밀번호 대신 SSH 키 쌍을 사용하여 원격 사용자를 인증합니다. 키 쌍은 개인 키와 공개 키로 구성됩니다. 인스턴스를 생성할 때 개인 키를 컴퓨터에 보관하고 공개 키를 제공해야 합니다. 지침은 Linux 인스턴스에서 키 쌍 관리를 참조하십시오.

단계 **13** **Show Advanced Options(**고급 옵션 표시**)** 링크를 클릭하여 옵션을 확장합니다.

단계 **14** (선택사항) **Initialization Script(**초기화 스크립트**)**에서 **Paste Cloud-Init Script(**클라우드 초기화 스크립트**)** 라디오 버튼을 클릭하여 ASA 가상을 위한 day0 컨피그레이션을 제공합니다. day0 컨피그레이션은 ASA 가상이 실행될 때 적용됩니다.

다음 예는 **Cloud-Init Script(Cloud-Init** 스크립트**)** 필드에서 복사하여 붙여넣을 수 있는 샘플 day0 컨피그레이션을 보여줍니다.

ASA 명령에 대한 자세한 내용은 ASA 구성 가이드 및 ASA 명령 참조를 참조하십시오.https://www.cisco.com/c/en/us/support/security/asa-5500-series-next-generation-firewalls/products-installation-and-configuration-guides-list.htmlhttps://www.cisco.com/c/en/us/support/security/asa-5500-series-next-generation-firewalls/products-command-reference-list.html

중요 이 예의 텍스트를 복사할 때는 서드파티 텍스트 편집기 또는 검증 엔진에서 스크립트를 검증하여 형식 오류를 방지하고 유효하지 않은 유니코드 문자를 제거해야 합니다.

```
!ASA Version 9.18.1
interface management0/0
management-only
nameif management
security-level 100
ip address dhcp setroute

no shut
!
same-security-traffic permit inter-interface
same-security-traffic permit intra-interface
!
crypto key generate rsa modulus 2048
ssh 0 0 managementssh timeout 60
ssh version 2
username admin nopassword privilege 15
username admin attributes
service-type admin
http server enable
http 0 0 management
aaa authentication ssh console LOCAL
```

단계 **15** **Create(**생성**)**를 클릭합니다.

다음에 수행할 작업

ASA 가상 인스턴스를 모니터링합니다. **Create(**생성**)** 버튼을 클릭하면 상태가 Provisionin (프로비저닝)으로 표시됩니다.

중요 상태를 모니터링하는 것이 중요합니다. ASA 가상 인스턴스가 Provisioning(프로비저닝)에서 Running(실행 중) 상태로 전환되면, 필요한 경우 ASA 가상 부팅이 완료되기 전에 VNIC를 연결해야 합니다.

인터페이스 연결

ASA 가상은 VNIC 하나가 연결된 상태로 Running(실행 중) 상태가 됩니다(**Compute(**컴퓨팅**)** > **Instances(**인스턴스**)** > **Instance Details(**인스턴스 세부 정보**)** > **Attached VNICs(**연결된 **VNIC)** 참조). 이를 기본 VNIC라고 하며, 관리 VCN에 매핑됩니다. ASA 가상이 첫 번째 부팅을 완료하기 전에, 이전에 생성한 다른 VCN 서브넷(내부, 외부)에 VNIC를 연결하여 VNIC가 ASA 가상에서 올바르게 탐지되게 해야 합니다.

단계 **1** 새로 시작한 ASA 가상 인스턴스를 선택합니다.

단계 **2** **Attached VNICs(**연결된 **VNIC)** > **Create VNIC(VNIC** 생성**)**를 선택합니다.

단계 **3** VNIC를 설명하는 **Name(**이름**)**을 입력합니다(예: 내부).

단계 **4** **Virtual Cloud Network(**가상 클라우드 네트워크**)** 드롭다운에서 VCN을 선택합니다.

단계 **5** **Subnet(**서브넷**)** 드롭다운에서 서브넷을 선택합니다.

단계 **6** **Use Network Security Groups to Network Traffic(**트래픽을 제어하기 위해 네트워크 보안 그룹 사용**)**을 선택하고, 선택된 VCN에 대해 구성한 보안 그룹을 선택합니다.

단계 **7** **Skip Source Destination Check(**소스 대상 검사 건너뛰기**)** Network Security Groups to Control Traffic(트래픽 제어를 위한 네트워크 보안 그룹)을 확인합니다.

단계 **8** (선택 사항) **Private IP Address(**개인 **IP** 주소**)**를 지정합니다. VNIC를 위한 특정 IP를 선택하려는 경우에만 필요합니다.

IP를 지정하지 않으면 OCI는 사용자가 서브넷에 할당한 CIDR 블록에서 IP 주소를 할당합니다.

단계 **9** **Save Changes(**변경 사항 저장**)**을 클릭하여 VNIC를 생성합니다.

단계 **10** 구축에 필요한 각 VNIC에 대해 이 절차를 반복합니다.

연결된 **VNIC**에 대한 경로 규칙 추가

내부 및 외부 경로 테이블에 경로 테이블 규칙을 추가합니다.

단계 **1** **Networking(**네트워킹**)** > **Virtual Cloud Networks(**가상 클라우드 네트워크**)**를 선택하고 VCN(내부 또는 외부)에 연결된 기본 경로 테이블을 클릭합니다.

단계 **2** 경로 규칙 추가를 클릭합니다.

단계 **3** **Target Type(**대상 유형**)** 드롭 다운에서 **Private IP(**프라이빗 **IP)**를 선택합니다.

단계 **4** **Destination Type(**대상 유형**)** 드롭다운에서 **CIDR Block(CIDR** 블록**)**을 선택합니다.

단계 **5** 대상 **IPv4 CIDR** 블록을 입력합니다(예: 0.0.0.0/0).

단계 **6** **Target Selection(**대상 선택**)** 필드에 VNIC의 프라이빗 IP 주소를 입력합니다.

VNIC에 IP 주소를 명시적으로 할당하지 않은 경우 VNIC 세부 정보에서 자동 할당된 IP 주소를 찾을 수 있습니다 (**Compute(**컴퓨팅**)** > **Instances(**인스턴스**)** > **Instance Details(**인스턴스 세부 정보**)** > **Attached VNICs(**연결된 **VNIC)**).

단계 **7** 경로 규칙 추가를 클릭합니다.

단계 **8** 구축에 필요한 각 VNIC에 대해 이 절차를 반복합니다.

참고 ASA Virtual (정적 및 DHCP) 컨피그레이션에는 별도의 라우팅 규칙이 필요합니다.

OCI에서 ASA 가상 인스턴스에 액세스

SSH(Secure Shell) 연결을 사용하여 실행중인 인스턴스에 연결할 수 있습니다.

- 대부분의 UNIX 스타일 시스템에는 기본적으로 SSH 클라이언트가 포함되어 있습니다.
- Windows 10 및 Windows Server 2017 시스템에는 Oracle Cloud Infrastructure에서 생성된 SSH 키를 사용하여 인스턴스를 생성한 경우 필요한 OpenSSH 클라이언트가 포함되어야 합니다.
- 다른 Windows 버전의 경우 http://www.putty.org에서 무료 SSH 클라이언트인 PuTTY를 다운로드할 수 있습니다.

사전 요건

인스턴스에 연결하려면 다음 정보가 필요합니다.

- 해당 인스턴스의 퍼블릭 IP 주소 콘솔의 Instance Details(인스턴스 세부 사항) 페이지에서 해당 주소를 가져올 수 있습니다. Navigation(탐색) 메뉴를 엽니다. **Core Infrastructure(**코어 인프라**)**에서 **Compute(**계산**)**로 이동하여 **Instances(**인스턴스**)**를 클릭합니다. 그런 다음 인스턴스를 선택합니다. 또는 Core Services API ListVnicAttachments 및 GetVnic 작업을 사용할 수 있습니다.
- 인스턴스의 사용자 이름 및 비밀번호입니다.
- 인스턴스를 시작할 때 사용한 SSH 키 쌍의 개인 키 부분에 대한 전체 경로입니다. 키 쌍에 대한 자세한 내용은 Linux 인스턴스에서 키 쌍 관리를 참조하십시오.

참고 day0 컨피그레이션에 지정된 자격 증명을 사용하거나 인스턴스 시작 과정에서 생성된 SSH 키 쌍을 사용하여 ASA 가상 인스턴스에 로그인할 수 있습니다.

SSH를 사용하여 ASA 가상 인스턴스에 연결

Unix식 시스템에서 ASA 가상 인스턴스에 연결하려면 SSH를 사용해서 인스턴스에 연결합니다.

단계 **1** 다음 명령을 사용해서 파일 권한을 설정해서 본인만 파일을 읽을 수 있도록 합니다.

```
$ chmod 400 <private_key>
```

여기서 각 항목은 다음을 나타냅니다.

<private_key>는 액세스하고자 하는 인스턴스에 연결된 개인 키를 포함하고 있는 파일의 전체 경로와 이름입니다.

단계 **2** 다음 SSH 명령을 사용해서 인스턴스에 액세스합니다.

```
$ ssh –i <private_key> <username>@<public-ip-address>
```

여기서 각 항목은 다음을 나타냅니다.

<private_key>는 액세스하고자 하는 인스턴스에 연결된 개인 키를 포함하고 있는 파일의 전체 경로와 이름입니다.

<username>은 ASA 가상 인스턴스를 위한 사용자 이름입니다.

<public-ip-address>는 콘솔에서 가져온 인스턴스 IP 주소입니다.

OpenSSH를 사용하여 ASA 가상 인스턴스 연결

Windows 시스템에서 ASA 가상 인스턴스에 연결하려면 OpenSSH를 사용하여 인스턴스에 로그인합니다.

단계 **1** 이 키 쌍을 처음 사용하는 경우에는 파일 읽기만 가능하도록 파일 권한을 설정해야 합니다.

다음을 수행합니다.

a) Windows Explorer에서 개인 키 파일로 이동하여 파일을 마우스 오른쪽 버튼으로 클릭한 다음 **Properties(**속성**)**를 클릭합니다.
b) **Security(**보안**)** 탭에서 **Advanced(**고급**)**를 클릭합니다.
c) 소유자가 사용자 계정인지 확인하십시오.
d) **Disable Inheritance(**상속 비활성화**)**를 클릭한 다음 이 개체에 대해 상속된 권한을 명시적 권한으로 변환을 선택합니다.
e) 사용자 계정이 아닌 각 권한 항목을 선택하고 **Remove(**제거**)**를 클릭합니다.
f) 사용자 계정에 대한 액세스 권한이 모든 권한인지 확인합니다.
g) 변경 내용을 저장합니다.

단계 **2** 인스턴스에 연결하려면 Windows PowerShell을 열고 다음 명령을 실행합니다.

```
$ ssh –i <private_key> <username>@<public-ip-address>
```

여기서 각 항목은 다음을 나타냅니다.

<private_key>는 액세스하고자 하는 인스턴스에 연결된 개인 키를 포함하고 있는 파일의 전체 경로와 이름입니다.

<username>은 ASA 가상 인스턴스를 위한 사용자 이름입니다.

<public-ip-address>는 콘솔에서 가져온 인스턴스 IP 주소입니다.

PuTTY를 사용하여 ASA 가상 인스턴스 연결

PuTTY를 사용하여 Windows 시스템에서 ASA 가상 인스턴스에 연결하려면:

단계 **1** PuTTY를 엽니다.

단계 **2** **Category(**범주**)** 창에서 **Session(**세션**)**을 선택하고 다음을 입력합니다.

- 호스트 이름 또는 **IP** 주소**:**

  <username>@<public-ip-address>

  여기서 각 항목은 다음을 나타냅니다.

  <username>은 ASA 가상 인스턴스의 사용자 이름입니다.

  <public-ip-address>은 콘솔에서 검색한 인스턴스 공용 IP 주소입니다.

- 포트**:** 22
- 연결 유형**:** SSH

단계 **3** **Category(**카테고리**)** 창에서 **Window(**창**)**를 확장한 다음 **Translation(**변환**)**을 선택합니다.

단계 **4** **Remote character set(**원격 문자 집합**)** 드롭 다운 목록에서 **UTF-8**을 선택합니다.

Linux 기반 인스턴스의 기본 로캘 설정은 UTF-8이며, 이 설정이 동일한 로캘을 사용하도록 PuTTY를 구성합니다.

단계 **5** **Category(**카테고리**)** 창에서 **Connection(**연결**)**, **SSH**를 차례로 확장한 다음 **Auth(**인증**)**를 클릭합니다.

단계 **6** **Browse(**찾아보기**)**를 클릭한 다음 개인 키를 선택합니다.

단계 **7** **Open(**열기**)**을 클릭하여 세션을 시작합니다.

인스턴스에 처음 연결하는 경우에는 서버의 호스트 키가 레지스트리에 캐시되지 않는다는 메시지가 표시될 수 있습니다. **Yes(**예**)**를 클릭하여 연결을 계속합니다.

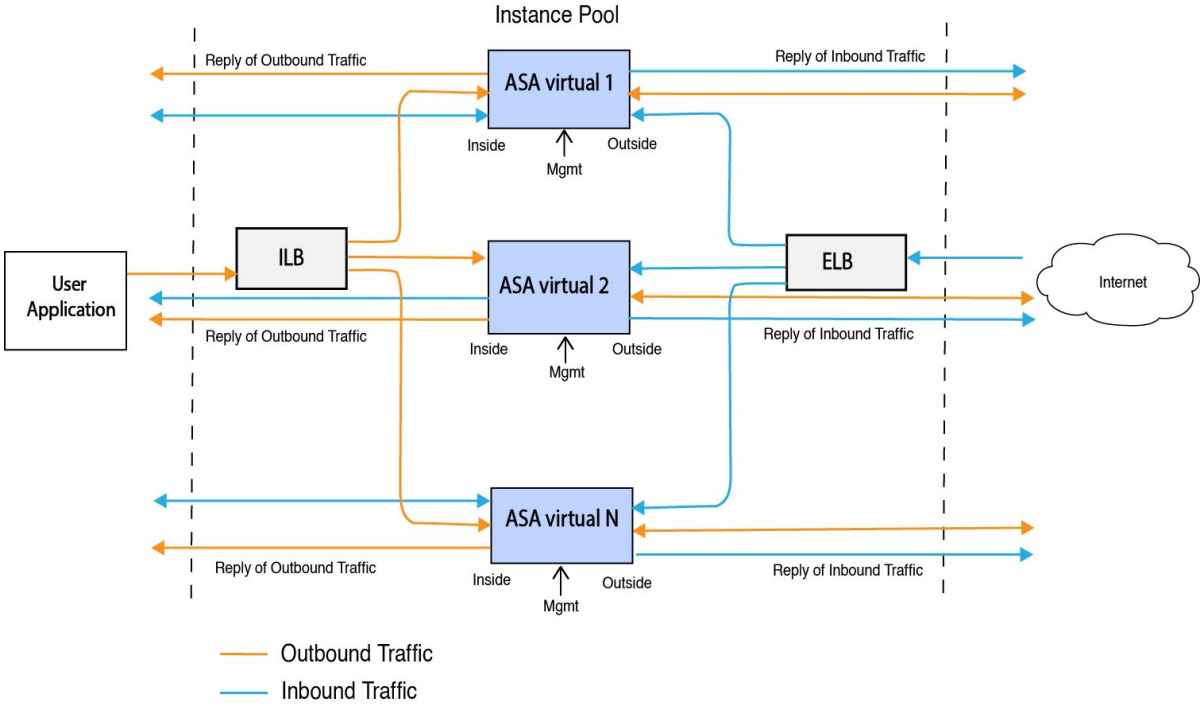
11 장

OCI에 ASA 가상 Auto Scale 솔루션 구축

Autoscale 사용 사례

ASA 가상의 사용 사례 – OCI Autoscale 솔루션은 사용 사례 다이어그램에 나와 있습니다. 인터넷 연결 로드 밸런서에는 리스너 및 대상 그룹 조합을 사용하여 활성화된 포트가 있는 공용 IP 주소가 있습니다.

그림 *52:* 사용 사례 다이어그램



포트 기반 분기는 네트워크 트래픽을 대상으로 구현할 수 있습니다. 이 작업은 NAT 규칙을 통해 수행할 수 있습니다. 이 컨피그레이션 예는 다음 섹션에서 설명합니다.

사전 요구 사항

권한 및 정책

다음은 솔루션을 구현하는 데 필요한 OCI 권한 및 정책입니다.

1. 사용자 및 그룹

참고 사용자 및 그룹을 생성하려면 OCI 사용자 또는 테넌시 관리자여야 합니다.

Oracle Cloud Infrastructure 사용자 계정 및 사용자 계정이 속한 그룹을 생성합니다. 사용자 계정이 있는 관련 그룹이 존재한다면 생성하지 않아도 됩니다. 사용자 및 그룹 생성 지침은 그룹 및 사용자 생성을 참고하십시오.

2. 그룹 정책

정책을 생성한 다음 그룹에 매핑해야 합니다. 정책을 생성하려면 **OCI** > **Identity & Security(ID** 및 보안**)** > **Policies(**정책**)** > **Create Policy(**정책 생성**)**로 이동합니다. 다음 정책을 생성하고 원하는 그룹에 추가합니다.

- <*Group_Name*> 그룹을 허용하여 컴파트먼트 <*Compartment_Name*>에서 메트릭 사용
- <*Group_Name*>그룹을 허용하여 컴파트먼트 <*Compartment_Name*>에서 알람 관리
- <*Group_Name*> 그룹을 허용하여 컴파트먼트 <*Compartment_Name*>에서 ons-topics 관리
- <*Group_Name*> 그룹을 허용하여 컴파트먼트 <*Compartment_Name*>에서 메트릭 검사
- <*Group_Name*> 그룹을 허용하여 컴파트먼트 <*Compartment_Name*>에서 메트릭 읽기
- <*Group_Name*> 그룹을 허용하여 컴파트먼트 <*Compartment_Name*>에서 태그 이름 공간 (tag-namespace) 사용
- <*Group_Name*> 그룹을 허용하여 컴파트먼트 <*Compartment_Name*>에서 로그 그룹 읽기
- <*Group_Name*>그룹을 허용하여 컴파트먼트 <*Compartment_Name*>에서 인스턴스 풀 (instance-pool) 사용
- <*Group_Name*> 그룹을 허용하여 테넌시에서 클라우드 쉘(cloud-shell) 사용
- <*Group_Name*> 그룹을 허용하여 테넌시에서 objectstorage-namespace 읽기
- <*Group_Name*> 그룹을 허용하여 테넌시에서 리포지토리 관리

참고 테넌시 레벨에서 정책을 생성할 수도 있습니다. 모든 권한을 제공하는 방법은 사용자가 결정합니다.

3. **Oracle Functions**에 대한 권한

   Oracle-Function이 다른 Oracle Cloud Infrastructure 리소스에 액세스할 수 있게 하려면, 동적 그룹에 기능을 포함한 다음 동적 그룹에 리소스에 대한 액세스 권한을 부여하는 정책을 만듭니다.

4. 동적 그룹 생성

   동적 그룹을 만들려면 **OCI** > **Identity & Security(ID** 및 보안**)** > **Dynamic Group(**동적 그룹**)** > **Create Dynamic Group(**동적 그룹 생성**)**으로 이동합니다.

   동적 그룹을 생성하는 동안 다음 규칙을 지정합니다.

   ```
   모든 {resource.type = 'fnfunc', resource.compartment.id = '<Your_Compartment_OCID>'}
   ```

   동적 그룹에 대한 자세한 내용은 다음을 참조하십시오.

   - https://docs.oracle.com/en-us/iaas/Content/Functions/Tasks/functionsaccessingociresources.htm
   - https://docs.oracle.com/en-us/iaas/Content/Identity/Tasks/managingdynamicgroups.htm

5. 동적 그룹에 대한 정책 생성

   정책을 추가하려면 **OCI** > **Identity & Security(ID** 및 보안**)** > **Policies(**정책**)** > **Create Policy(**정책 생성**)**로 이동합니다. 그룹에 다음 정책을 추가합니다.

   ```
   <Dynamic_Group_Name> 다이내믹 그룹을 허용하여 컴파트먼트<Compartment_OCID>에서 모든 리소스 관리
   ```

GitHub에서 파일 다운로드

ASA 가상 – OCI Autoscale 솔루션은 GitHub 리포지토리로서 전달됩니다. 리포지토리에서 파일을 가져오거나 다운로드할 수 있습니다.

Python3 환경

make.py 파일은 복제된 리포지토리에 있습니다. 이 프로그램은 Oracle 함수와 템플릿 파일을 Zip 파일로 압축합니다. 이 파일을 대상 폴더에 복사합니다. 이러한 작업을 수행하려면 Python 3 환경이 구성되어야 합니다.

참고 이 python 스크립트는 Linux 환경에서만 사용할 수 있습니다.

인프라 구성

다음을 구성해야 합니다.

1. **VCN**

   ASA 가상 애플리케이션의 필요에 맞게 VCN을 만듭니다. 인터넷에 대한 경로가 연결된 서브넷이 하나 이상 있는 인터넷 게이트웨이를 이용해 VCN을 만듭니다.

   VCN 생성에 대한 자세한 내용은 https://docs.oracle.com/en-us/iaas/Content/GSG/Tasks/creatingnetwork.htm을 참조하십시오.

2. 애플리케이션 서브넷

   ASA 가상 애플리케이션의 필요에 맞게 서브넷을 만듭니다. 이 사용 사례에 맞는 솔루션을 구현하려면 ASA 가상 인스턴스에 3개의 서브넷이 필요합니다.

   서브넷 생성에 대한 자세한 내용은 https://docs.oracle.com/en-us/iaas/Content/Network/Tasks/managingVCNs_topic-Overview_of_VCNs_and_Subnets.htm#을 참조하십시오.

3. 외부 서브넷

   서브넷은 인터넷 게이트웨이에 대한 기본 경로가 '0.0.0.0/0'이어야 합니다. 이 서브넷에는 Cisco ASA 가상 및 인터넷 연결 로드 밸런서의 외부 인터페이스가 포함되어 있습니다. 아웃바운드 트래픽에 NAT 게이트웨이가 추가되었는지 확인합니다.

   자세한 내용은 다음 문서를 참조하십시오.

   - https://docs.oracle.com/en-us/iaas/Content/Network/Tasks/managingIGs.htm
   - https://docs.oracle.com/en-us/iaas/Content/Network/Tasks/NATgateway.htm#To_create_a_NAT_gateway

4. 내부 서브넷

   NAT/인터넷 게이트웨이가 있거나 없는 애플리케이션 서브넷과 유사할 수 있습니다.

참고 ASA 가상 상태 프로브의 경우 포트 80을 통해 메타데이터 서버(169.254.169.254)에 연결할 수 있습니다.

5. 관리 서브넷

   관리 서브넷은 ASA 가상에 대한 SSH 액세스를 지원하기 위해 퍼블릭이어야 합니다.

6. 보안 그룹 - **ASA** 가상 인스턴스에 대한 네트워크 보안 그룹

   다음 요구 사항을 충족하는 ASA 가상 인스턴스에 대한 보안 그룹을 구성합니다.

   - (동일한 VCN에 있는) Oracle Functions는 ASA 가상의 관리 주소에 대한 SSH 연결을 수행합니다.
   - 관리 호스트는 ASA 가상 인스턴스에 대한 SSH 액세스가 필요할 수 있습니다.
   - ASA 가상은 라이선싱을 위해 CSSM/위성 서버와의 통신을 시작합니다.

7. 개체 스토리지 네임스페이스

   이 개체 스토리지 네임스페이스는 configuration.txt 파일이 있는 정적 웹사이트를 호스팅하는 데 사용합니다. configuration.txt 파일에 대해 사전 인증된 요청을 생성해야 합니다. 이 사전 인증된 URL은 템플릿 구축 중에 사용됩니다.

참고 업로드된 다음 컨피그레이션이 HTTP URL을 통해 ASA 가상 인스턴스에 액세스할 수 있는지 확인합니다.

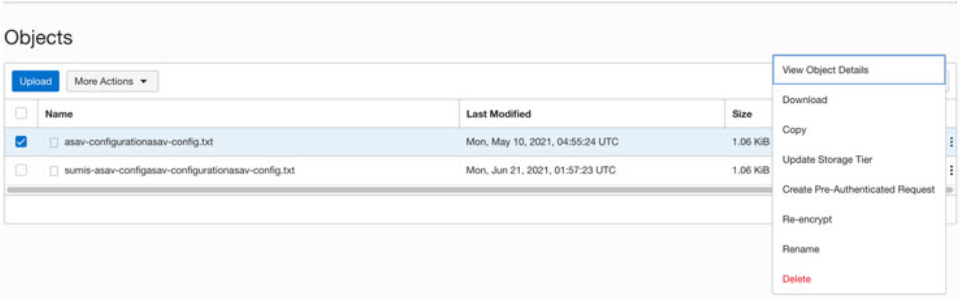
부팅된 ASA 가상은 다음 명령을 실행합니다. `$ copy /noconfirm <configuration.txt file's pre-authenticated request URL > disk0:Connfiguration.txt`

이 명령을 사용하면 configuration.txt 파일을 사용하여 ASA 가상 실행을 구성할 수 있습니다.

8. **configuration.txt** 파일 업로드

   ASA 가상 컨피그레이션 파일의 사전 인증된 요청 URL을 생성하려면 다음을 수행합니다.

   1. **Buckets(**버킷**)** > **Create Bucket(**버킷 생성**)**을 클릭합니다.
   2. **Upload(**업로드**)**를 클릭합니다.
   3. 구성 파일이 업로드되면, 아래 그림에서처럼 **Create Pre-Authenticated Request(**사전 인증된 요청 생성**)**를 선택합니다.



참고 이제 oracle-function에서 컨피그레이션 파일에 액세스할 수 있습니다.

네트워크 설정

1. **Inbound traffic(**인바운드 트래픽**)**

   configuration.txt의 <*Application VM IP*> 주소가 단계 2에 언급된 주소와 일치하는지 확인합니다.

2. **Outbound Traffic(**아웃 바운드 트래픽**)**

   - configuration.txt의 <*External Server IP*> 주소가 단계 2에 언급된 주소와 일치하는지 확인합니다.
   - 외부 VCN에 NAT 게이트웨이 하나가 있는지 확인합니다.
   - 아래 그림에서처럼 외부 VCN의 경로 테이블에 있는 동일한 <*External Server IP*> 주소가 NAT 게이트웨이를 대상으로 하는지 확인합니다.

Destination	Target Type	Target
0.0.0.0/0	Internet Gateway	outside-ig
8.8.8.8/32	NAT Gateway	nat-gw

비밀번호 암호화

참고 이 절차에 대한 자세한 내용은 Create and Secrets(자격 증명 모음 및 비밀번호 생성)을 참고하십시오.

ASA 가상의 비밀번호는 자동 확장 중에 사용하는 모든 ASA 가상 인스턴스를 구성하는 데 사용하며, ASA 가상 인스턴스의 CPU 사용량 데이터를 검색하는 데 사용합니다.

따라서 때때로 비밀번호를 저장하고 처리해야 합니다. 빈번한 변경 및 취약성 가능성 때문에, 비밀번호를 일반 텍스트 형식으로 수정하거나 저장할 수는 없습니다. 비밀번호는 반드시 암호화된 형식이어야 합니다.

암호화된 형식으로 비밀번호를 가져오려면 다음을 수행합니다.

단계 **1** Vault를 생성합니다.

OCI Vault는 마스터 암호화 키를 안전하게 생성하고 저장하는 서비스와, 이러한 서비스를 이용한 암호화 및 암호 해독 방법을 제공합니다. 따라서 (아직 생성하지 않았다면) Vault는 자동 확장 솔루션의 나머지 부분과 동일한 구획에 생성해야 합니다.

OCI > **Identity & Security(ID** 및 보안**)** > **Vault** > **Choose or Create New Vault(**새 **Vault** 선택 또는 생성**)**

.

단계 **2** 마스터 암호화 키를 생성합니다.

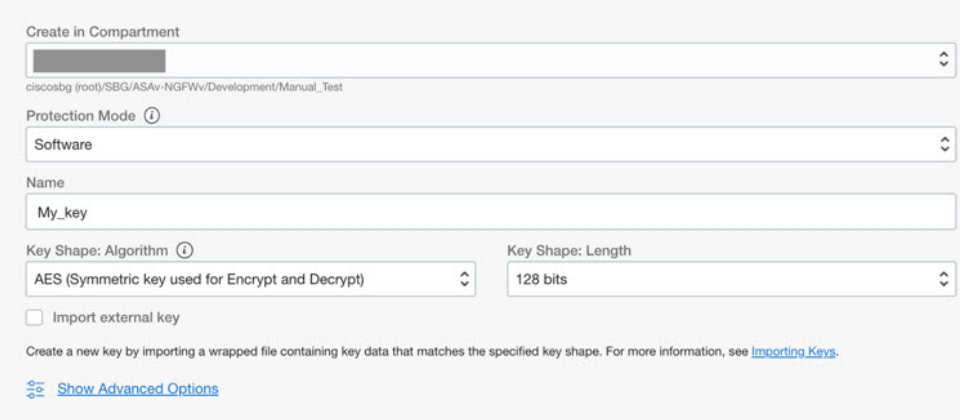
일반 텍스트 비밀번호를 암호화하려면 마스터 암호화 키 하나가 필요합니다.

OCI > **Identity & Security(ID** 및 보안**)** > **Vault** > **Choose or Create Key(**키 선택 또는 생성**)**으로 이동합니다.

임의 비트 길이의 알고리즘에서 키를 선택합니다.

1. AES – 128, 192, 256
2. RSA – 2048, 3072, 4096
3. ECDSA – 256, 384, 521

그림 ***53:*** 새 키 생성



단계 **3** 암호화된 비밀번호를 만듭니다.

1. **OCI** > **Open CloudShell(CloudShell** 열기**)(OCI Cloud Terminal)**로 이동합니다.
2. <*Password*>를 자신의 비밀번호로 교체하여 다음 명령을 실행합니다.

```
echo -n '<Password>' | base64
```

3. 선택한 Vault에서 암호화 엔드포인트 및 마스터 암호화 키 OCID를 복사합니다. 다음 값을 교체하고 암호화 명령을 실행합니다.
   - 사용자 키의 OCID가 포함된 KEY_OCID

- 사용자 Vault의 암호화 엔드포인트 URL이 포함된 Cryptographic_Endpoint_URL
- 비밀번호와 사용자의 비밀번호

암호화 명령

```
oci kms crypto encrypt --key-id Key_OCID --endpoint
Cryptographic_Endpoint_URL --plaintext <base64-value-of-password>
```

4. 위 명령의 출력에서 암호문을 복사하고 필요한 경우 사용합니다.

ASA 컨피그레이션 파일 준비

애플리케이션이 구축되었거나 구축 계획을 사용할 수 있는지 확인합니다.

단계 **1** 구축하기 전에 다음 입력 매개변수를 수집합니다.

매개변수	데이터 유형	설명
tenancy_ocid	문자열	계정이 속한 테넌시의 OCID입니다. 테넌시 OCID를 찾는 방법은 여기를 참조하십시오. 테넌시 OCID는 `ocid1.tenancy.oc1..<unique_ID>` 같은 형식을 취합니다.
compartment_id	문자열	리소스를 생성할 구획의 OCID입니다. 예: `ocid1.compartment.oc1..<unique_ID>`
compartment_name	문자열	구획의 이름
region	문자열	리소스를 생성할 지역의 고유 식별자입니다. 예: `us-phoenix-1, us-ashburn-1`

매개변수	데이터 유형	설명
lb_size	문자열	외부 및 내부 로드 밸런서의 사전 프로비저닝된 총 대역폭(인그레스 + 이그레스)을 결정하는 템플릿입니다. 지원되는 값: 100Mbps, 10Mbps, 10Mbps-Micro, 400Mbps, 8000Mbps 예: 100Mbps
availability_domain	쉼표로 구분된 값	예: Tpeb:PHX-AD-1 참고 Cloud Shell에서 **oci iam availability-domain list** 명령을 실행하여 가용성 도메인 이름을 가져옵니다.
min_and_max_instance_count	쉼표로 구분된 값	인스턴스 풀에서 유지할 최소 및 최대 인스턴스 수입니다. 예: 1,5
autoscale_group_prefix	문자열	템플릿을 사용하여 생성되는 모든 리소스의 이름을 지정하는 데 사용하는 접두사입니다. 예를 들어 리소스 접두사가 'autoscale'로 지정된 경우, 모든 리소스의 이름은 autoscale_resource1, autoscale_resource2 등으로 지정됩니다.
asav_config_file_url	URL	ASA 가상을 구성하는 데 사용할 개체 스토리지에 업로드된 구성 파일의 URL입니다. 참고 컨피그레이션 파일의 사전 인증된 요청 URL이 제공되어야 합니다. 예: https://objectstorage.<region-name>.oraclecloud.com/<object-storage-name>/oci-asav-configuration.txt
mgmt_subnet_ocid	문자열	사용할 관리 서브넷의 OCID입니다.
inside_subnet_ocid	문자열	사용할 내부 서브넷의 OCID입니다.
outside_subnet_ocid	문자열	사용할 외부 서브넷의 OCID입니다.

매개변수	데이터 유형	설명
mgmt_nsg_ocid	문자열	사용할 관리 서브넷 네트워크 보안 그룹의 OCID입니다.
inside_nsg_ocid	문자열	사용할 내부 서브넷 네트워크 보안 그룹의 OCID입니다.
outside_nsg_ocid	문자열	사용할 외부 서브넷 네트워크 보안 그룹의 OCID입니다.
elb_listener_port	쉼표로 구분된 값	외부 로드 밸런서 리스너의 통신 포트 목록입니다. 예: 80
ilb_listener_port	쉼표로 구분된 값	내부 로드 밸런서 리스너의 통신 포트 목록입니다. 예: 80
health-check-port	문자열	상태 확인을 실행할 로드 밸런서의 백엔드 서버 포트입니다. 예: 8080
instance_shape	문자열	생성할 인스턴스의 셰이프입니다. 셰이프는 인스턴스에 할당되는 CPU 수, 메모리 양 및 기타 리소스를 결정합니다. 지원되는 셰이프: "VM.Standard2.4" 및 "VM.Standard2.8"
lb_bs_policy	문자열	내부 및 외부 로드 밸런서의 백엔드 집합에 사용할 로드 밸런서 정책입니다. 로드 밸런서 정책의 작동 방식에 대한 자세한 내용은 여기를 참조하십시오. 지원되는 값: "ROUND_ROBIN", "LEAST_CONNECTIONS", "IP_HASH"

매개변수	데이터 유형	설명
image_name	문자열	인스턴스 구성을 생성하는 데 사용하는 마켓플레이스 이미지의 이름입니다. 기본값: "Cisco ASAv(ASA 가상 방화벽)" 참고 맞춤형 이미지를 구축하고 싶은 사용자는 custom_image_ocid 매개변수를 구성해야 합니다.
image_version	문자열	사용할 OCI Marketplace에서 사용할 수 있는 ASA 가상 이미지의 버전입니다. 현재 9.15.1.15 및 9.16.1 버전을 사용할 수 있습니다. 기본값: "Cisco ASAv(ASA 가상 방화벽)"
scaling_thresholds	쉼표로 구분된 값	축소 및 확장에 사용할 CPU 사용량 임계값입니다. 축소 및 확장 임계값을 쉼표로 구분된 입력으로 지정하십시오. 예: 15,50 여기서 15는 축소 임계값이고 50은 확장 임계값입니다.
custom_image_ocid	문자열	마켓플레이스 이미지를 사용하지 않을 경우 인스턴스 컨피그레이션을 생성하는 데 사용할 맞춤형 이미지의 OCID입니다. 참고 custom_image_ocid는 선택적 매개변수입니다.
asav_password	문자열	암호화된 형식의 ASA 가상용 암호로, 구성을 위한 ASA 가상로의 SSH에 사용합니다. 비밀번호를 암호화하는 방법에 대한 지침은 구성 가이드나 여기를 참조하십시오.
cryptographic_endpoint	문자열	암호화 엔드포인트는 비밀번호 해독에 사용하는 URL입니다. Vault에서 찾을 수 있습니다.

매개변수	데이터 유형	설명
master_encryption_key_id	문자열	비밀번호가 암호화된 키의 OCID입니다. Vault에서 찾을 수 있습니다.
Profile Name(프로필 이름)		OCI에서의 사용자 프로파일 이름입니다. 사용자의 프로파일 섹션에서 확인할 수 있습니다. 예: oracleidentitycloudservice/<user>@<mail>.com
개체 스토리지 네임스페이스		테넌시를 만들 때 생성되는 고유한 식별자입니다. 이 값은 **OCI** > **Administration(**관리**)** > **Tenancy Details(**테넌시 세부 정보**)**에서 확인할 수 있습니다.
권한 부여 토큰		Oracle-Functions를 OCI 컨테이너 레지스트리에 푸시할 권한을 부여하는 docker 로그인의 비밀번호로 사용됩니다. 토큰을 얻으려면 **OCI** > **Identity(ID)** > **Users(**사용자**)** > **User Details(**사용자 세부 정보**)** > **Auth Tokens(**인증 토큰**)** > **Generate Token(**토큰 생성**)**으로 이동합니다.

단계 **2** 로드 밸런서 상태 프로브 및 액세스 정책에 대한 개체, 라이선싱 및 NAT 규칙을 구성합니다.

```
! Default route via outside
route outside 0.0.0.0 0.0.0.0 <Outside Subnet gateway> 2

! Health Check Configuration
object network metadata-server
host 169.254.169.254
object service health-check-port
service tcp destination eq <health-check-port>
object service http-port
service tcp destination eq <traffic port>
route inside 169.254.169.254 255.255.255.255 <Inside Subnet GW> 1

! Health check NAT
nat (outside,inside) source static any interface destination static interface metadata-server service
health-check-port http-port
nat (inside,outside) source static any interface destination static interface metadata-server service
health-check-port http-port

! Outbound NAT
object network inside-subnet
subnet <Inside Subnet> <Inside Subnet Gateway>
object network external-server
host <External Server IP>
nat (inside,outside) source static inside-subnet interface destination static interface external-server


! Inbound NAT
object network outside-subnet
```

```
subnet <Outside Subnet> <Outside Subnet GW>
object network http-server-80
host <Application VM IP>
nat (outside,inside) source static outside-subnet interface destination static interface http-server-80


!
dns domain-lookup outside
DNS server-group DefaultDNS

! License Configuration
call-home
profile license
destination transport-method http
destination address http <URL>
debug menu license 25 production
license smart
feature tier standard
throughput level <Entitlement>
licence smart register idtoken <License token> force
!
```

이러한 상태 프로브 연결 및 데이터 플레인 구성은 액세스 정책에서 허용되어야 합니다.

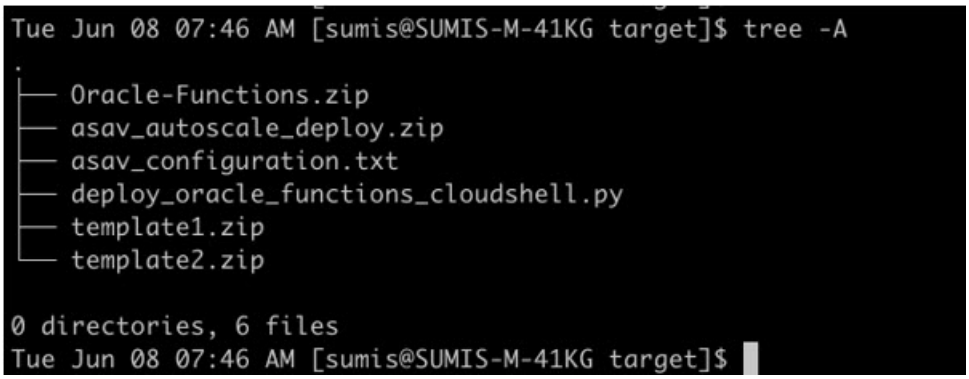
단계 **3** 컨피그레이션 세부 정보를 이용해 *configuration.txt* 파일을 업데이트합니다.

단계 **4** *configuration.txt* 파일을 사용자가 생성한 개체 스토리지 공간에 업로드하고, 업로드된 파일에 대한 사전 인증된 요청을 생성합니다.

참고 configuration.txt의 사전 인증된 요청 URL을 스택 구축에 사용하는지 확인합니다.

단계 **5** Zip 파일을 생성합니다.

make.py 파일은 복제된 리포지토리에 있습니다. `python3 make.py build` 명령을 실행하여 zip 파일을 생성합니다. 대상 폴더에는 다음 파일이 있습니다.



참고 클라우드 셸을 사용하여 자동 확장 솔루션을 구축하는 경우 `python3 make.py build`를 실행하기 전에 *easy_deploy/deployment_parameters.json* 파일을 업데이트합니다. 업데이트에 대해서는 단계 1 및 Oracle Functions 구축을 참조하십시오.

OCI에 자동 확장 구축

구축을 위한 사전 요구 사항 단계가 완료되면 OCI 스택 생성을 시작합니다. 수동 구축을 수행하거나 Cloud Shell을 사용하여 자동 확장 구축을 수행할 수 있습니다. 사용자 버전에 맞는 구축 스크립트 및 템플릿은 GitHub 리포지토리에서 제공됩니다.

수동 구축

엔드 투 엔드 Autoscale 솔루션 구축은 Terraform 템플릿 1 스택 구축 , Oracle Functions 구축, Terraform 템플릿-2 구축이라는 3가지 단계로 구성됩니다.

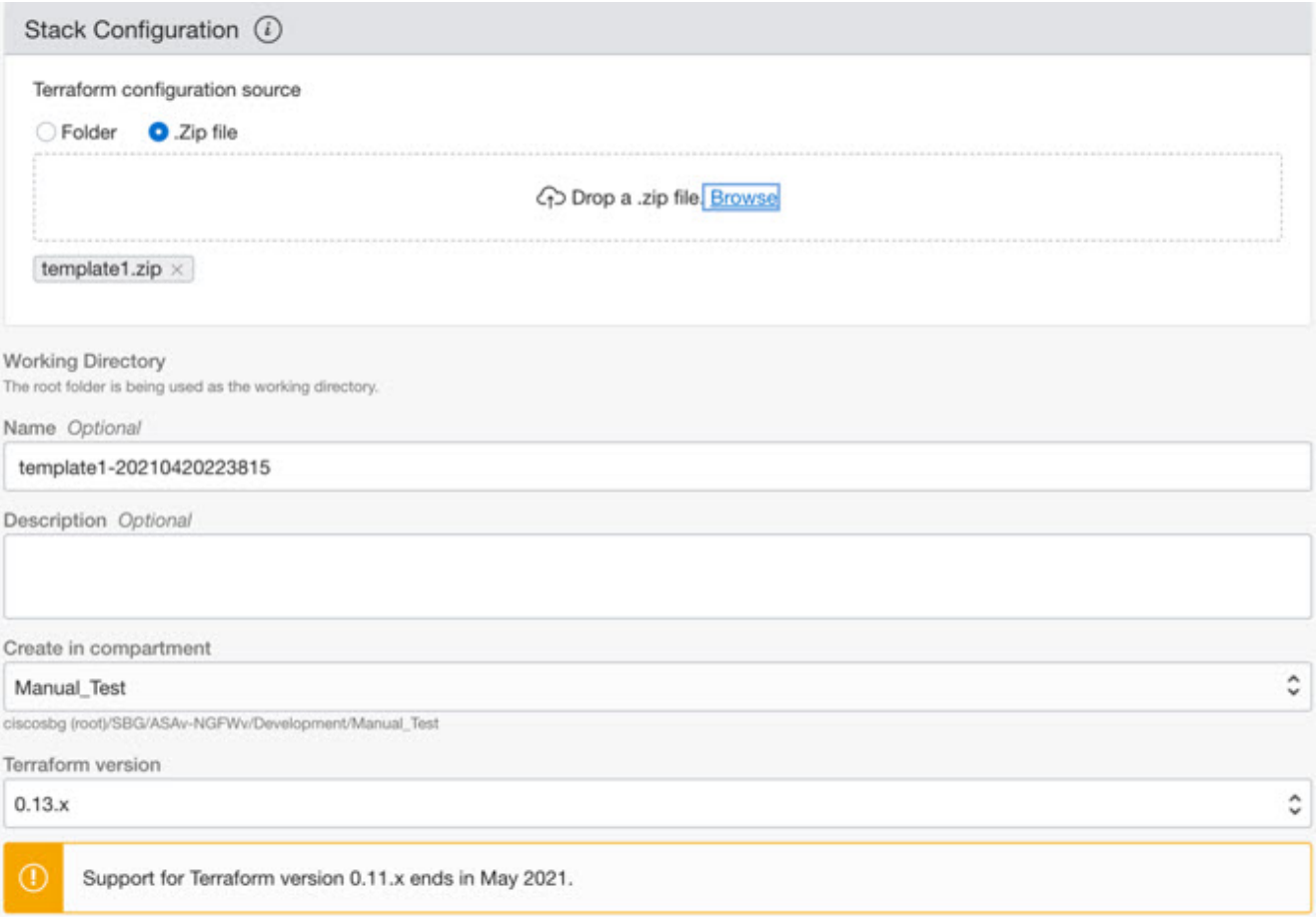
Terraform 템플릿 1 스택 구축

단계 **1** OCI 포털에 로그인합니다.

화면의 우측 상단에 지역이 표시됩니다. 원하는 지역에 있는지 정기적으로 확인합니다.

단계 **2** **Developer Service**(개발자 서비스) > **Resource Manager**(리소스 관리자) > **Stack**(스택) > **Create Stack**(스택 생성)을 선택합니다.

My Configuration(내 구성)을 선택하고, 아래 그림에서처럼 대상 폴더에 있는 *Terraform template1.zip* 팔을일 Terraform 컨피그레이션 소스로 선택합니다.



단계 **3** **Transform version(**변형 버전**)** 드롭다운 목록에서 0.13.x 또는 0.14.x를 선택합니다.

단계 **4** 다음 단계에서 단계 1에 수집된 모든 세부 정보를 입력합니다.

참고 유효한 입력 매개변수를 입력하십시오. 입력하지 않으면 다음 단계에서 스택 구축이 실패할 수 있습니다.

단계 **5** 다음 단계에서 **Terraform Actions(Terraform** 작업**)** > **Apply(**적용**)**를 클릭합니다.

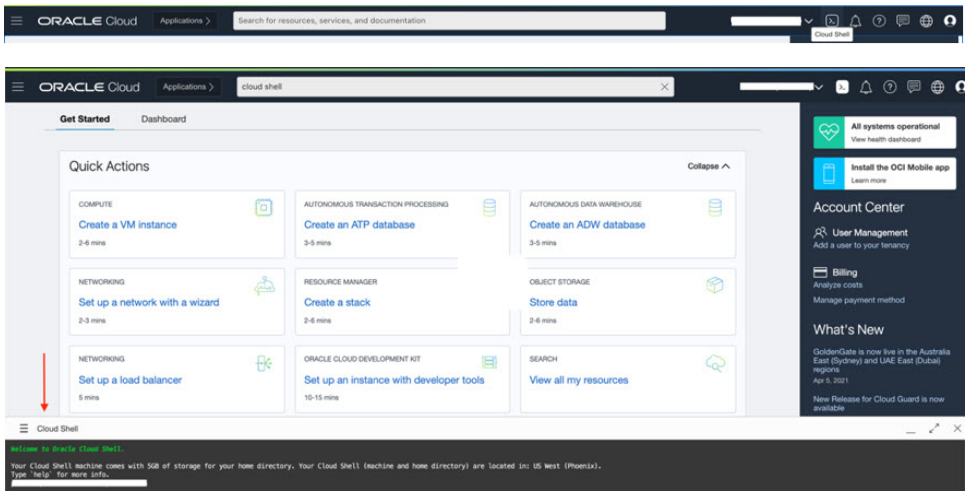
구축을 성공적으로 완료되면 Oracle Functions 구축을 진행합니다.

Oracle Functions 구축

참고 이 단계는 *Terraform* 템플릿 *1*을 구축한 후에만 수행해야 합니다.

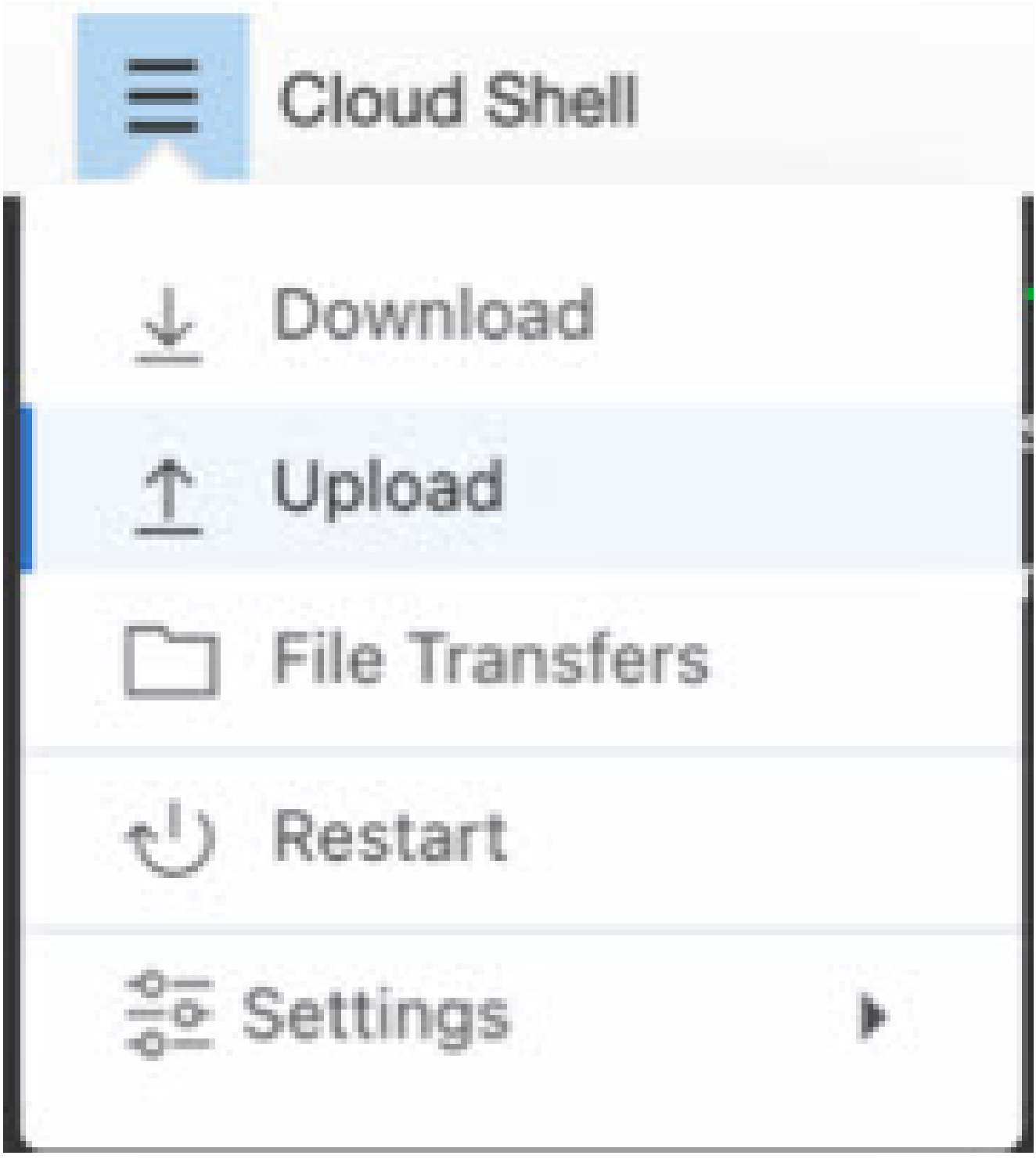
OCI에서 Oracle Functions는 OCI 컨테이너 레지스트리에 저장되는 Docker 이미지로 업로드됩니다. 구축할 때 Oracle Functions를 (Terraform 템플릿 1에서 생성된) OCI 애플리케이션 중 하나에 푸시해야 합니다.

단계 **1** OCI Cloud Shell을 엽니다.



단계 **2** *deploy_oracle_functions_cloudshell.py* 및 *Oracle-Functions.zip*을 업로드합니다.

Cloud Shell의 햄버거 메뉴에서 **Upload(**업로드**)**를 선택합니다.



단계 **3** **ls** 명령을 사용하여 파일을 확인합니다.

```
$ ls
Deploy_Oracle_Functions.py  Oracle-Functions.zip
```

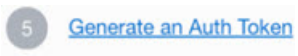
단계 **4** `python3 Deploy_Oracle_Functions.py -h`를 실행합니다. `deploy_oracle_functions_cloudshell.py` 스크립트에는 아래 그림에서처럼 help 인수를 사용하여 세부 정보를 찾을 수 있는 몇 가지 입력 매개변수가 필요합니다.

```
$ python3 Deploy_Oracle_Functions.py -h
usage: Deploy_Oracle_Functions.py [-h] -a  -r  -p  -c  -o  -t

*** Script to deploy Oracle Function for OCI ASAv Autoscale Solution ***

Instruction to find values of required arguments:
Application Name: Name of Application created by first Terraform Template
Region Identifier: OCI -> Administration -> Region Management
Profile Name: OCI -> Profile
Compartment OCID: OCI -> Identity -> Compartment -> Compartment Details
Object Storage Namespace: OCI -> Administration -> Tenancy Details
Authorization Token: OCI -> Identity -> Users -> User Details -> Auth Tokens -> Generate Token

optional arguments:
  -h, --help  show this help message and exit
  -a          Name of Application in OCI to which functions will be deployed
  -r          Region Identifier
  -p          Profile Name of User
  -c          Compartment OCID
  -o          Object Storage Namespace
  -t          Authorization Token for Docker Login (*Please Put in Quotes)
```

스크립트를 실행하려면 다음 인수를 전달해야 합니다.

표 ***22:*** 인수 및 세부 정보

인수	세부 사항
애플리케이션 이름	Terraform 템플릿 1 구축에 의해 생성된 OCI 애플리케이션의 이름입니다. 값은 템플릿 1에 제공된 "**autoscale_group_prefix**"와 "**_application**" 접미사를 결합하여 얻습니다.
지역 식별자	지역 식별자는 다양한 지역에 대해 OCI에서 고정된 지역 코드워드입니다. 예: 'us-phoenix-1'(Phoenix) 또는 "ap-melbourne-1"(Melbourne). 지역 식별자가 포함된 모든 지역의 목록을 가져오려면 **OCI** > **Administration(**관리**)** > **Region Management(**지역 관리**)**로 이동합니다.
Profile Name(프로필 이름**)**	OCI에서의 단순 사용자 프로파일 이름입니다. 예: *oracleidentitycloudservice/<user>@<mail>.com* 이름은 사용자의 프로파일 섹션에서 확인할 수 있습니다.

인수	세부 사항
구획 **OCID**	구획의 OCID(Oracle Cloud Identifier)입니다. 사용자가 OCI 애플리케이션을 보유한 구획 OCID입니다. **OCI** > **Identity(ID)** > **Compartment(**구획**)** > **Compartment Details(**구획 상세 정보**)**로 이동합니다.
개체 스토리지 네임스페이스	테넌시를 만들 때 생성되는 고유한 식별자입니다. **OCI** > **Administration(**관리**)** > **Tenancy Details(**테넌시 세부 정보**)**로 이동합니다.
권한 부여 토큰	Oracle-Functions를 OCI 컨테이너 레지스트리에 푸시할 권한을 부여하는 docker 로그인의 비밀번호로 사용됩니다. 구축 스크립트에서 따옴표로 묶인 토큰을 지정합니다. **OCI** > **Identity(ID)** > **Users(**사용자**)** > **User Details(**사용자 세부 정보**)** > **Auth Tokens(**인증 토큰**)** > **Generate Token(**토큰 생성**)**으로 이동합니다. 어떤 이유로든 User Details(사용자 세부사항)가 표시되지 않는다면 **Developer services(**개발자 서비스**)** > **Functions(**기능**)**를 클릭합니다. Terraform 템플릿 1에서 생성한 애플리케이션으로 이동합니다. **Getting Started(**시작하기**)**를 클릭하고 Cloud Shell Setup(클라우드 쉘 설정)을 선택하면 아래에서처럼 인증 토큰을 생성할 수 있는 링크가 표시됩니다. 

단계 **5** 유효한 입력 인수를 전달하여 `python3 Deploy_Oracle_Functions.py` 명령을 실행합니다. 모든 기능이 구축될 때까지 시간이 오래 걸릴 수 있습니다. 구축이 끝나면 파일을 제거하고 Cloud Shell을 닫아도 됩니다.

Terraform 템플릿-2 구축

템플릿 2는 알람, 기능 호출을 위한 ONS 항목 같은 알람 생성 관련 리소스를 구축합니다. 템플릿 2 구축은 Terraform 템플릿-1 구축과 유사합니다.

단계 **1** OCI 포털에 로그인합니다.

화면의 우측 상단에 지역이 표시됩니다. 원하는 지역에 있는지 정기적으로 확인합니다.

단계 **2** **Developer Service(**개발자 서비스**)** > **Resource Manager(**리소스 관리자**)** > **Stack(**스택**)** > **Create Stack(**스택 생성**)**을 선택합니다.

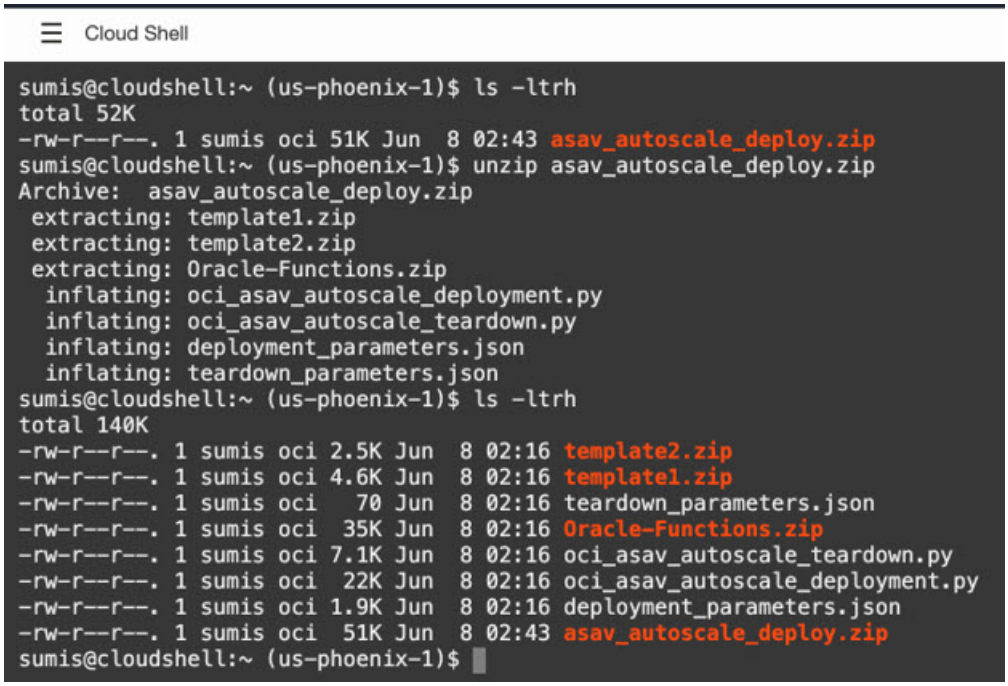
대상 폴더의 *Terraform template2.zip*을 Terraform 컨피그레이션의 소스로 선택합니다.

단계 3 다음 단계에서 **Terraform Actions(Terraform 작업)** > **Apply(적용)**를 클릭합니다.

Cloud Shell을 사용하여 자동 확장 구축

구축 오버헤드를 방지하기 위해 간편한 엔드 투 엔드 구축 스크립트를 호출하여 자동 확장 솔루션(terraform template1, template2 및 oracle functions)을 구축할 수 있습니다.

단계 1 대상 폴더의 *asav_autoscale_deploy.zip* 파일을 클라우드 쉘에 업로드하고 파일의 압축을 풉니다.



단계 2 `python3 make.py` 빌드 명령을 실행하기 전에 *deployment_parameters.json*에서 입력 매개변수를 업데이트했는지 확인합니다.

단계 3 자동 확장 솔루션 구축을 시작하려면 클라우드 쉘에서 `python3 oci_asav_autoscale_deployment.py` 명령을 실행합니다.

솔루션 구축이 끝나려면 약 10~15분 정도 걸립니다.

솔루션 구축 중에 오류가 발생하면 오류 로그가 저장됩니다.

구축 검증

모든 리소스가 구축되고 Oracle Functions가 알람 및 이벤트와 연결되어 있는지 확인합니다. 기본적으로 인스턴스 풀은 최소 및 최대 인스턴스 수가 0입니다. OCI UI에서 원하는 최소 및 최대 개수로 인스턴스 풀을 수정할 수 있습니다. 이렇게 하면 새 ASA 가상 인스턴스가 트리거됩니다.

인스턴스를 하나만 실행하고 관련 워크플로우를 확인한 다음 동작을 검증하여, 예상대로 작동하는지 확인하는 것이 좋습니다. 이 검증을 게시하면 ASA 가상의 실제 요구 사항을 구축할 수 있습니다.

참고 OCI 확장 정책 때문에 제거되지 않도록, ASA 가상 인스턴스의 최소 수를 **Scale-In protected(**축소 보호**)**로 지정합니다.

자동 확장 업그레이드

Autoscale 스택 업그레이드

이 릴리스에서는 업그레이드가 지원되지 않습니다. 스택을 재구축해야 합니다.

ASA 가상 **VM** 업그레이드

이 릴리스에서는 ASA 가상 VM 업그레이드가 지원되지 않습니다. 필요한 ASA 가상 이미지를 사용하여 스택을 재구축해야 합니다.

인스턴스 풀

1. 인스턴스 풀의 최소 및 최대 인스턴스 수를 변경하려면 다음을 수행합니다.

   Developer Services(개발자 서비스**)** > **Function(**기능**)** > **Application Name(created by Terraform Template 1)(**애플리케이션 이름**, Terraform** 템플릿 **1**에서 생성**)** > **Configuration(**구성**)**을 클릭합니다.

   min_instance_count 및 max_instance_count를 각각 변경합니다.

2. 인스턴스 삭제/종료는 축소와는 다릅니다. 축소 작업이 아니라 외부 작업 때문에 인스턴스 풀의 인스턴스가 삭제/종료된 경우, 인스턴스 풀은 복구를 위해 새 인스턴스를 자동으로 시작합니다.

3. Max_instance _count는 확장 작업에 대한 임계값 제한을 정의하지만, UI를 통해 인스턴스 풀의 인스턴스 수를 변경하면 이 제한을 초과할 수 있습니다. UI의 인스턴스 수가 OCI 애플리케이션에 설정된 max_instance_count보다 작은지 확인합니다. 그렇지 않다면 적절하게 임계값을 늘리십시오.

4. 애플리케이션에서 바로 인스턴스 풀의 인스턴스 수를 줄이면, 프로그래밍 방식으로 설정된 정리 작업이 수행되지 않습니다. 백엔드가 두 로드 밸런서 모두에서 드레인되거나 제거되지 않으므로, ASA 가상에 라이선스가 있다면 백엔드가 손실됩니다.

5. 몇 가지 이유 때문에, ASA 가상 인스턴스가 비정상적으로 응답하지 않고 일정 기간 SSH를 통해 연결할 수 없다면 인스턴스가 인스턴스 풀에서 강제로 제거되며 라이선스가 손실될 수 있습니다.

Oracle Functions

- Oracle Functions는 실제로는 docker 이미지입니다. 이러한 이미지는 OCI 컨테이너 레지스트리의 루트 디렉토리에 저장됩니다. 이러한 이미지는 삭제하면 안 됩니다. Autoscale 솔루션에서 사용하는 기능도 삭제되기 때문입니다.
- Terraform 템플릿 1에서 생성한 OCI 애플리케이션에는 Oracle Functions가 정상적으로 작동하는 데 필요한 중요한 환경 변수가 포함되어 있습니다. 반드시 필요한 경우가 아니면 이러한 환경 변수의 값이나 형식을 변경해선 안 됩니다. 변경 사항은 새 인스턴스에만 반영됩니다.

로드 밸런서 백엔드 집합

OCI에서, 인스턴스 풀에 대한 로드 밸런서 연결은 ASA 가상에서의 관리 인터페이스로 구성된 기본 인터페이스를 사용하는 경우에만 지원됩니다. 따라서 내부 인터페이스는 내부 로드 밸런서의 백엔드 집합에 연결되고 외부 인터페이스는 외부 로드 밸런서의 백엔드 집합에 연결됩니다. 이러한 IP는 자동으로 백엔드 집합에 추가되거나 집합에서 제거되지 않습니다. Autoscale 솔루션은 두 작업을 모두 프로그래밍 방식으로 처리합니다. 그러나 외부 활동, 유지 보수 또는 문제 해결을 위해 작업을 수동으로 수행해야 할 수도 있습니다.

요구 사항에 따라 리스너 및 백엔드 집합을 사용하여 로드 밸런서에서 추가 포트를 열 수 있습니다. 향후 인스턴스 IP는 백엔드 집합에 자동으로 추가되지만, 이미 존재하는 인스턴스 IP는 수동으로 추가해야 합니다.

로드 밸런서에 리스너 추가

포트를 로드 밸런서에 리스너로 추가하려면 **OCI** > **Networking(**네트워킹**)** > **Load Balancer(**로드 밸런서**)** > **Listener(**리스너**)** > **Create Listener(**리스터 생성**)**로 이동합니다.

백엔드 집합에 백엔드 등록

ASA 가상 인스턴스를 로드 밸런서에 추가하려면 ASA 가상 인스턴스 외부 인터페이스 IP를 외부 로드 밸런서의 백엔드 집합에서 백엔드로 구성해야 합니다. 내부 인터페이스 IP는 내부 로드 밸런서의 백엔드 집합에서 백엔드로 구성해야 합니다. 사용 중인 포트가 리스너에 추가되었는지 확인합니다.

OCI에서 자동 확장 구성 삭제

Terraform을 사용하여 구축된 스택한 OCI의 Resource Manager를 사용하여 동일한 방식으로 삭제할 수 있습니다. 스택을 삭제하면 스택에서 생성된 모든 리소스가 제거되고 이러한 리소스와 연결된 모든 정보가 영구적으로 제거됩니다.

참고 스택을 삭제할 때는 인스턴스 풀의 최소 인스턴스 수)을 0으로 설정한 다음 인스턴스가 종료될 때까지 대기하는 것이 좋습니다. 이렇게 하면 모든 인스턴스를 제거하고 잔여물을 남기지 않을 수 있습니다.

수동 삭제를 수행하거나 Cloud Shell을 사용하여 자동 확장 삭제 을 사용할 수 있습니다.

수동 삭제

엔드 투 엔드 Autoscale 솔루션 삭제는 Terraform 템플릿 2 스택 삭제, Oracle-Functions 삭제, Terraform 템플릿 1 스택 삭제 라는 3가지 단계로 구성됩니다.

Terraform 템플릿 2 스택 삭제

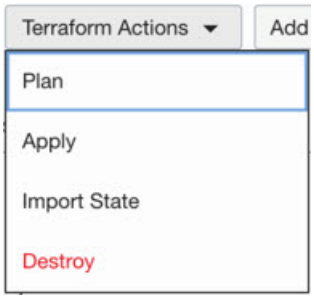
자동 확장 구성을 삭제하려면 Terraform 템플릿 2 스택 삭제부터 시작해야 합니다.

단계 **1** OCI 포털에 로그인합니다.

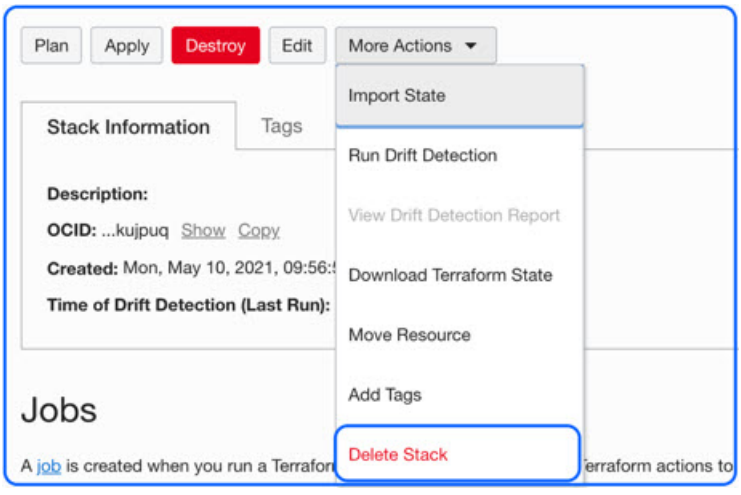
화면의 우측 상단에 지역이 표시됩니다. 원하는 지역에 있는지 정기적으로 확인합니다.

단계 **2** **Developer Services(**개발자 서비스**)** > **Resource Manager(**리소스 관리자**)** > **Stack(**스택**)**을 선택합니다.

단계 **3** Terraform 템플릿 2에서 생성한 스택을 선택한 다음, 아래 그램에서처럼 **Terraform Actions(Terraform** 작업**)** 드롭다운 메뉴에서 **Destroy(**삭제**)**를 클릭합니다.



Destroy Job(삭제 작업)이 생성됩니다. 리소스를 하나씩 제거하기 때문에 시간이 조금 걸립니다. 삭제 작업이 완료되면 아래 그림에서처럼 스택을 삭제할 수 있습니다.



단계 **4** 계속해서 Oracle 기능을 삭제합니다.

Oracle-Functions 삭제

Oracle-Function 구축은 Terraform 템플릿 스택 구축의 일부가 아니며, 클라우드 쉘을 사용하여 별도로 업로드됩니다. 따라서 Terraform 스택 삭제에서는 이 구축 삭제를 지원하지 않습니다. Terraform 템플릿 1에서 생성한 OCI 애플리케이션 내의 모든 Oracle-Functions를 삭제해야 합니다.

단계 **1** OCI 포털에 로그인합니다.

화면의 우측 상단에 지역이 표시됩니다. 원하는 지역에 있는지 정기적으로 확인합니다.

단계 **2** **Developer Services(**개발자 서비스**)** > **Functions(**기능**)**를 선택합니다. 템플릿 1 스택에서 생성된 애플리케이션 이름을 선택합니다.

단계 **3** 이 애플리케이션에서 각 기능을 방문하여 삭제합니다.

Terraform 템플릿 1 스택 삭제

참고 템플릿 1 스택 삭제는 모든 Oracle-Functions를 삭제한 후에만 성공합니다.

Terraform 템플릿 2 삭제와 동일합니다.

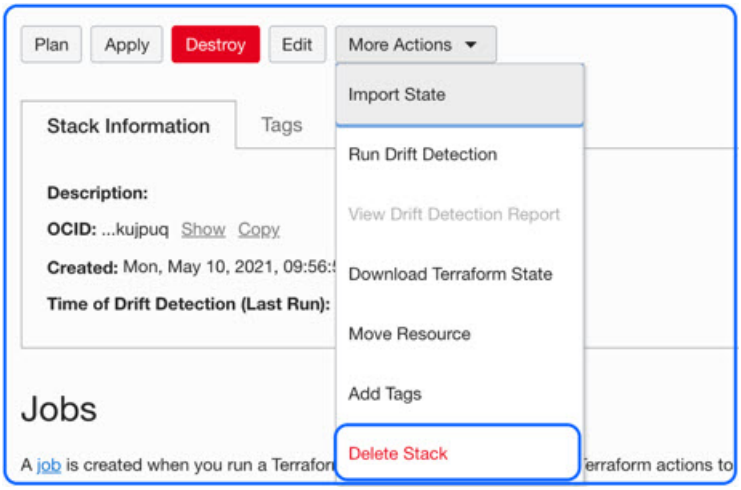
단계 **1** OCI 포털에 로그인합니다.

화면의 우측 상단에 지역이 표시됩니다. 원하는 지역에 있는지 정기적으로 확인합니다.

단계 **2** **Developer Services(**개발자 서비스**)** > **Resource Manager(**리소스 관리자**)** > **Stack(**스택**)**을 선택합니다.

단계 **3** Terraform 템플릿 2에서 생성한 스택을 선택한 다음 **Terraform Actions(Terraform** 작업**)**드롭다운 메뉴에서 **Destroy(**삭제**)**를 클릭합니다. Destroy Job(삭제 작업)이 생성됩니다. 리소스를 하나씩 제거하기 때문에 시간이 조금 걸립니다.

단계 **4** 제거 작업이 완료되면 아래 그림과 같이 **More Actions(**추가 작업**)** 드롭다운 메뉴에서 스택을 삭제할 수 있습니다.



Terraform 템플릿 1 스택을 삭제한 후에는 모든 리소스가 삭제되고 어떤 유형의 잔여물도 없는지 확인해야 합니다.

Cloud Shell을 사용하여 자동 확장 삭제

사용자는 스크립트를 사용하여 클라우드 쉘에서 `python3 oci_asav_autoscale_teardown.py` 명령을 실행하여 스택과 oracle 기능을 삭제할 수 있습니다. 스택을 수동으로 구축하는 경우 stack1 및 stack2의 stack ID를 업데이트하고, *teardown_parameters.json* 파일에서 애플리케이션 ID를 업데이트합니다.

12 장

Google Cloud Platform에 ASA 가상 구축

GCP(Google Cloud Platform)에서 ASA 가상을 구축할 수 있습니다.

GCP의 ASA 가상 구축 정보

GCP를 사용하면 Google과 동일한 인프라에서 애플리케이션, 웹사이트 및 서비스를 빌드, 구축 및 확장할 수 있습니다.

ASA 가상은 물리적 ASA 와 동일한 소프트웨어를 실행하여 가상 폼 팩터에서 검증된 보안 기능을 제공합니다. ASA 가상은 퍼블릭 GCP에서 구축될 수 있습니다. 그러면 시간이 경과함에 따라 해당 위치를 확장, 축소 또는 이동하는 가상 및 물리적 데이터 센터 워크로드를 보호하기 위한 구성이 가능하게 됩니다.

GCP 시스템 유형 지원

ASA 가상 필요에 따라 Google 가상 머신 유형 및 크기를 선택합니다.

ASA 가상에서는 다음 범용 *N1, N2* 및 컴퓨팅 최적화 *C2* GCP 머신 유형을 지원합니다.

표 ***23:*** 지원되는 컴퓨팅 최적화 시스템 유형

컴퓨팅 최적화 시스템 유형	속성	
	vCPUs	메모리**(GB)**
c2-standard-4	4	16
c2-standard-8	8	32

컴퓨팅 최적화 시스템 유형	속성	
	vCPUs	메모리**(GB)**
c2-standard-16	16	64

표 ***24:*** 지원되는 범용 시스템 유형

머신 유형	속성	
	vCPUs	메모리**(GB)**
n1-standard-4	4	15
n1-standard-8	8	30
n1-standard-16	16	60
n2-standard-4	4	16
n2-standard-8	8	32
n2-standard-16	16	64
n1-highcpu-8	8	7.2
n1-highcpu-16	16	14.4
n2-highcpu-8	8	8
n2-highcpu-16	16	16
n2-highmem-4	4	32
n2-highmem-8	8	64
n2-highmem-16	16	128

- ASA 가상에는 최소 3 개의 인터페이스가 필요합니다.
- 지원되는 최대 vCPU는 16개입니다.
- 메모리 최적화 머신 유형은 지원되지 않습니다.

GCP에서 계정을 생성하고, GCP Marketplace의 ASA 가상 방화벽(ASA 가상) 제품을 사용해서 ASA 가상 인스턴스를 실행한 다음 GCP 머신 유형을 선택합니다.

C2 컴퓨팅 최적화 머신 유형 제한

컴퓨팅 최적화 C2 머신 유형에는 다음과 같은 제한 사항이 있습니다.

- 컴퓨팅 최적화 머신 유형에는 지역 영구 디스크를 사용할 수 없습니다. 자세한 내용은 Google 문서 지역 영구 디스크 추가 또는 크기 조정을 참조하십시오.

- 범용 및 메모리 최적화 시스템 유형과는 다른 디스크 제한이 적용됩니다. 자세한 내용은 Google 문서 블록 스토리지 성능을 참고하십시오.
- 일부 영역 및 지역에서만 사용할 수 있습니다. 자세한 내용은 Google 문서 사용 가능한 지역 및 영역을 참조하십시오.
- 일부 CPU 플랫폼에서만 사용 가능합니다. 자세한 내용은 Google 문서 CPU 플랫폼을 참조하십시오.

ASA 가상 및 GCP의 사전 요건

- https://cloud.google.com에서 GCP 계정을 만듭니다.
- GCP 프로젝트를 생성합니다. Google 문서, 프로젝트 생성을 참조하십시오.
- ASA 가상에 라이선스를 부여합니다. ASA 가상 라이선스를 등록하기 전에는 저성능 모드에서 실행됩니다. 이 모드에서는 100개의 연결 및 100Kbps의 처리량만 허용됩니다. 라이선스: 스마트 소프트웨어 라이선싱을 참조하십시오.
- 인터페이스 요구 사항:
  - 관리 인터페이스 - ASA 가상을 ASDM에 연결할 때 사용합니다. 통과 트래픽에는 사용할 수 없습니다.
  - 내부 인터페이스 - ASA 가상을 내부 호스트에 연결하는 데 사용합니다.
  - 외부 인터페이스 - ASA 가상을 공용 네트워크에 연결하는 데 사용합니다.
- 통신 경로:
  - ASA 가상에 액세스하기 위한 공용 IP.
- ASA 가상 시스템 요구 사항은 Cisco Secure Firewall ASA호환성을 참조하십시오.

ASA 가상 및 GCP에 대한 지침 및 제한 사항

지원 기능

GCP의 ASA 가상은 다음 기능을 지원합니다.

- GCP VPC(Virtual Private Cloud)에 구축
- 인스턴스당 최대 16개의 vCPU
- 라우팅 모드(기본값)
- 라이선싱 – BYOL만 지원됩니다.

지원되지 않는 기능

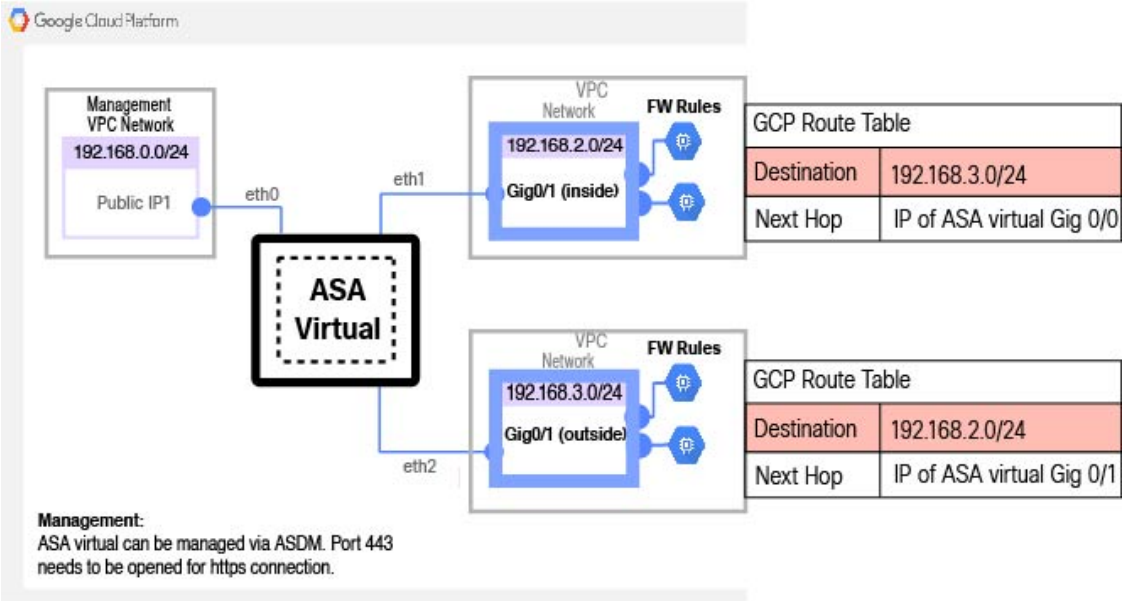
GCP의 ASA 가상은 다음을 지원하지 않습니다.

- IPv6
  - 인스턴스 수준 IPv6 설정은 GCP에서 지원되지 않습니다.
  - 로드 밸런서만이 IPv6 연결을 수락하고, IPv4를 통해 GCP 인스턴스로 프록시할 수 있습니다.
- 점보 프레임
- ASA 가상 기본 HA
- 자동 확장
- 투명/인라인/패시브 모드

GCP의 ASA 가상을 위한 네트워크 토폴로지 샘플

다음 그림은 Routed Firewall Mode의 ASA 가상에 대한 권장 네트워크 토폴로지와 ASA 가상에 대해 GCP에 구성된 3개의 서브넷(관리, 내부 및 외부)을 보여줍니다.

그림 ***54: GCP*** 구축에 대한 ***ASA*** 가상 샘플



Google Cloud Platform에 ASA 가상 구축

Google Cloud Platform에 ASA 가상 구축

GCP(Google Cloud Platform)에서 ASA 가상을 구축할 수 있습니다.

VPC 네트워크 생성

시작하기 전에

ASA 가상을 구축하려면 ASA 가상을 구축하기 전에 3개의 네트워크를 생성해야 합니다. 네트워크는 다음과 같습니다.

- 관리 서브넷의 관리 VPC
- 내부 서브넷의 내부 VPC
- 외부 서브넷의 외부 VPC

또한 ASA 가상로 트래픽 흐름을 허용하도록 경로 테이블 및 GCP 방화벽 규칙을 설정합니다. 경로 테이블 및 방화벽 규칙은 ASA 가상 자체에 구성된 규칙과 다릅니다. 연결된 네트워크 및 기능에 따라 GCP 경로 테이블 및 방화벽 규칙의 이름을 지정합니다. GCP의 ASA 가상을 위한 네트워크 토폴로지 샘플, 234 페이지의 내용을 참조하십시오.

단계 **1** GCP 콘솔에서 **Networking(**네트워킹**)** > **VPC network(VPC** 네트워크**)** > **VPC networks(VPC** 네트워크**)**를 선택하고 **Create VPC Network(VPC** 네트워크 생성**)**를 클릭합니다.

단계 **2** **Name(**이름**)** 필드에 VPC 네트워크를 설명하는 이름(예: *vpc-asiasouth-mgmt*)을 입력합니다.

단계 **3** **Subnet creation mode(**서브넷 생성 모드**)**에서 **Custom(**맞춤형**)**을 클릭합니다.

단계 **4** **New subnet(**새로운 서브넷**)** 아래의 **Name(**이름**)** 필드에 원하는 이름(예: *vpc-asiasouth-mgmt*)을 입력합니다.

단계 **5** **Region(**지역**)** 드롭 다운 목록에서 자신의 구축에 적합한 지역을 선택합니다. 세 개의 네트워크는 모두 같은 지역에 있어야 합니다.

단계 **6** **IP** 어드레스 레인지 필드에 CIDR 포맷, 예를 들면 10.10.0.0/24의 형식으로 첫 번째 네트워크의 서브넷을 입력합니다.

단계 **7** 기타 모든 설정은 기본값으로 하고 **Create(**생성**)**를 클릭합니다.

단계 **8** 1~7 단계를 반복하여 VPC에 나머지 2개의 네트워크를 생성합니다.

방화벽 규칙 생성

ASA 가상인스턴스를 구축하는 동안 (SSH 및 HTTPS 연결을 허용하도록) 관리 인터페이스에 대한 방화벽 규칙을 적용합니다. GCP에서 ASA 가상 인스턴스 생성, 236 페이지를 참조하십시오. 요구 사항에 따라 내부 및 외부 인터페이스에 대한 방화벽 규칙을 생성할 수도 있습니다.

단계 1 GCP 콘솔에서 **Networking(**네트워킹**)** > **VPC network(VPC** 네트워크**)** > **Firewall(**방화벽**)**을 선택하고 **Create Firewall Rule(**방화벽 규칙 생성**)**을 클릭합니다.

단계 2 **Name(**이름**)** 필드에 방화벽 규칙을 설명하는 이름(예: *vpc-asiasouth-inside-fwrule*)을 입력합니다.

단계 3 **Network(**네트워크**)** 드롭 다운 목록에서 방화벽 규칙을 생성할 VPC 네트워크의 이름(예: *asav-south-inside*)을 선택합니다.

단계 4 **Targets(**대상**)** 드롭 다운 목록에서 방화벽 규칙을 위해서 적용할 옵션(예: **All instances in the network**)을 선택합니다.

단계 5 **Source IP ranges(**소스 **IP** 범위**)** 필드에 소스 IP 주소 범위를 CIDR 형식으로 입력합니다(예: 0.0.0.0/0).

트래픽은 이들 IP 주소 범위 내의 소스로부터만 허용됩니다.

단계 6 **Protocols and ports(**프로토콜 및 포트**)** 아래에서 **Specified protocols and ports(**명시된 프로토콜 및 포트**)**를 선택합니다.

단계 7 보안 규칙을 추가합니다.

단계 8 **Create(**생성**)**를 클릭합니다.

GCP에서 ASA 가상 인스턴스 생성

아래 단계를 완료하면 GCP Marketplace에서 제공하는 Cisco ASA 가상 방화벽(ASA 가상) 제품을 사용하여 ASA 가상 인스턴스를 구축할 수 있습니다.

단계 1 GCP 콘솔로 로그인합니다.

단계 2 **Navigation(**탐색**)** 메뉴**(** > **Marketplace(**마켓플레이스**)**를 클릭합니다 .

단계 3 Marketplace에서 “Cisco ASA virtual firewall(ASAv)”을 검색하고 제품을 선택합니다.

단계 4 **Launch(**실행**)**를 클릭합니다.

단계 5 인스턴스의 고유한 **Deployment name(**구축 이름**)**을 추가합니다.

단계 6 ASA 가상를 구축할 **Zone(**영역**)**을 선택합니다.

단계 7 적절한 **Machine type(**머신 유형**)**을 선택합니다. 지원되는 머신 유형 목록은 GCP의 ASA 가상 구축 정보, 231 페이지를 참조하십시오.

단계 8 (선택 사항) **SSH** 키**(**선택 사항**)**에 있는 SSH 키 쌍의 공개 키를 붙여넣습니다.

키 쌍은 GCP가 저장하는 공용 키와 사용자가 저장하는 개인 키 파일로 구성됩니다. 이 두 키를 함께 사용하면 인스턴스에 안전하게 연결할 수 있습니다. 인스턴스에 연결할 때 필요한 만큼 키 쌍을 알고 있는 위치에 확실히 저장해야 합니다.

단계 9 이 인스턴스에 액세스하기 위해 프로젝트 전체 SSH 키를 허용할지 아니면 차단할지를 선택합니다. Google 문서 Allowing or blocking project-wide public SSH keys from a Linux instance를 참조하십시오.

단계 10 (선택 사항) **Startup script(**시작 스크립트**)**에 ASA 가상에 대한 day0 컨피그레이션을 입력합니다. day0 컨피그레이션은 ASA 가상의 첫 번째 부팅 중에 적용됩니다.

다음 예는 시작 스크립트 필드에 복사하여 붙여 넣은 Day0 컨피그레이션의 샘플을 보여줍니다.

ASA 명령에 대한 자세한 내용은 ASA 구성 가이드 및 ASA 명령 참조를 참조하십시오.

중요 이 예의 텍스트를 복사할 때는 서드파티 텍스트 편집기 또는 검증 엔진에서 스크립트를 검증하여 형식 오류를 방지하고 유효하지 않은 유니코드 문자를 제거해야 합니다.

```
!ASA Version 9.15.1

interface management0/0

management-only
nameif management
security-level 100
ip address dhcp setroute
no shut
!
same-security-traffic permit inter-interface
same-security-traffic permit intra-interface
!
crypto key generate rsa modulus 2048
ssh 0 0 management
ssh timeout 60
ssh version 2
username admin password cisco123 privilege 15
username admin attributes
service-type admin
! required config end
dns domain-lookup management
dns server-group DefaultDNS
name-server 8.8.8.8
```

단계 **11** 프로비저닝된 디스크 공간에는 기본 **Boot disk type(**부팅 디스크 유형**)**과 **Boot disk size in GB(**부팅 디스크 크기 **(GB)**를 그대로 유지합니다.

단계 **12** **Network interfaces(**네트워크 인터페이스**)**에서 인터페이스를 구성합니다.

- 관리
- 내부
- 외부

참고 인터페이스를 생성한 후엔 거기에 인터페이스를 추가할 수 없습니다. 부적절한 인터페이스 컨피그레이션으로 인스턴스를 생성했을 경우 해당 인스턴스를 삭제하고 적절한 인터페이스 컨피그레이션으로 다시 생성해야 합니다.

a) **Network(**네트워크**)** 드롭 다운 목록에서 VPC 네트워크(예 : *vpc-asiasouth-mgmt*)를 선택합니다.

b) **External IP(**외부 **IP)** 드롭 다운 목록에서 적절한 옵션을 선택합니다.

관리 인터페이스를 위해선 **External IP(**외부 **IP) - Ephemeral(**일회성**)**을 선택합니다. 이는 내부 및 외부 인터페이스의 경우 선택 사항입니다.

c) **Done(**완료**)**을 클릭합니다.

단계 **13** **Firewall(**방화벽**)**에서 방화벽 규칙을 적용합니다.

- 인터넷의 **TCP** 포트 **22** 트래픽**(SSH** 액세스**)** 허용 확인란을 선택하여 SSH를 허용합니다.

- HTTPS 연결을 허용하려면 **Allow HTTPS traffic from the Internet(ASDM 액세스)(**인터넷에서 **HTTPS** 트래픽 허용**(ASDM 액세스))** 확인란을 선택합니다.

단계 **14** **More(**더 보기**)**를 클릭하여 보기를 확장하고 **IP Forwarding(IP** 전달**)**이 **On(**켜짐**)**으로 설정되어 있는지 확인합니다.

단계 **15** **Deploy(**구축**)**를 클릭합니다.

GCP 콘솔의 VM 인스턴스 페이지에서 인스턴스 상세 정보를 확인합니다. 내부 IP 주소, 외부 IP 주소 그리고 인스턴스를 시작하고 중지할 수 있는 제어 기능을 확인할 수 있습니다. 인스턴스를 수정해야 하는 경우 인스턴스를 중지해야 합니다.

GCP에서 ASA 가상 인스턴스에 액세스

구축 중에 SSH(포트 22를 통한 TCP 연결)를 허용하는 방화벽 규칙을 이미 활성화했는지 확인합니다. 자세한 내용은 GCP에서 ASA 가상 인스턴스 생성, 236 페이지를 참조하십시오.

이 방화벽 규칙은 ASA 가상 인스턴스에 대한 액세스를 활성화하고 다음 방법을 사용하여 해당 인스턴스에 연결할 수 있도록 합니다.

- 외부 IP
  - 기타 SSH 클라이언트 또는 서드파티 도구
- 시리얼 콘솔
- Gcloud 명령줄

더 자세한 내용은 Google 문서 Connecting to instances를 참조하십시오.

참고 day0 컨피그레이션에 지정된 자격 증명을 사용하거나 인스턴스 시작 과정에서 생성된 SSH 키 쌍을 사용하여 ASA 가상 인스턴스에 로그인할 수 있습니다.

외부 IP를 사용하여 ASA 가상 인스턴스에 연결

ASA 가상 인스턴스는 내부 IP와 외부 IP로 할당됩니다. 외부 IP를 사용해서 ASA 가상 인스턴스에 액세스할 수 있습니다.

단계 **1** GCP 콘솔에서 **Compute Engine(**컴퓨팅 엔진**)** > **VM instances(VM** 인스턴스**)**를 선택합니다.

단계 **2** ASA 가상 인스턴스 이름을 클릭해서 **VM** 인스턴스 상세정보 페이지를 엽니다.

단계 **3** **Details(**상세정보**)** 탭 아래에서 **SSH** 필드를 위한 드롭 다운 메뉴를 엽니다.

단계 **4** **SSH** 드롭 다운 메뉴에서 원하는 옵션을 선택합니다.

다음 방법을 사용해서 ASA 가상 인스턴스에 연결할 수 있습니다.

- 기타 SSH 클라이언트 또는 서드파티 도구 - 더 자세한 내용은 Google 문서 Connecting using third-party tools을 참조하십시오.

참고 day0 컨피그레이션에 지정된 자격 증명을 사용하거나 인스턴스 시작 과정에서 생성된 SSH 키 쌍을 사용하여 ASA 가상 인스턴스에 로그인할 수 있습니다.

SSH를 사용하여 ASA 가상 인스턴스에 연결

Unix식 시스템에서 ASA 가상 인스턴스에 연결하려면 SSH를 사용해서 인스턴스에 연결합니다.

단계 1 다음 명령을 사용해서 파일 권한을 설정해서 본인만 파일을 읽을 수 있도록 합니다.

```
$ chmod 400 <private_key>
```

여기서 각 항목은 다음을 나타냅니다.

<private_key>는 액세스하고자 하는 인스턴스에 연결된 개인 키를 포함하고 있는 파일의 전체 경로와 이름입니다.

단계 2 다음 SSH 명령을 사용해서 인스턴스에 액세스합니다.

```
$ ssh -i <private_key> <username>@<public-ip-address>
```

여기서 각 항목은 다음을 나타냅니다.

<private_key>는 액세스하고자 하는 인스턴스에 연결된 개인 키를 포함하고 있는 파일의 전체 경로와 이름입니다.

<username>은 ASA 가상 인스턴스를 위한 사용자 이름입니다.

<public-ip-address>는 콘솔에서 가져온 인스턴스 IP 주소입니다.

직렬 콘솔을 사용해서 ASA 가상 인스턴스 연결

단계 1 GCP 콘솔에서 **Compute Engine(**컴퓨팅 엔진**)** > **VM instances(VM** 인스턴스**)**를 선택합니다.

단계 2 ASA 가상 인스턴스 이름을 클릭해서 **VM** 인스턴스 상세정보 페이지를 엽니다.

단계 3 **Details(**상세정보**)** 탭 아래에서 **Connect to serial console(**직렬 콘솔 연결**)**을 클릭합니다.

더 자세한 내용은 Google 문서 Interacting with the serial console을 참조하십시오.

Gcloud를 사용하여 ASA 가상 인스턴스에 연결

단계 1 GCP 콘솔에서 **Compute Engine(**컴퓨팅 엔진**)** > **VM instances(VM** 인스턴스**)**를 선택합니다.

단계 2 ASA 가상 인스턴스 이름을 클릭해서 **VM** 인스턴스 상세정보 페이지를 엽니다.

단계 3 **Details(**상세정보**)** 탭 아래에서 **SSH** 필드를 위한 드롭 다운 메뉴를 엽니다.

단계 4 **View gcloud command(gcloud** 명령 보기**)** > **Run in Cloud Shell(**클라우드 셸에서 구동**)**을 클릭합니다.

클라우드 셸 터미널 창이 열립니다. 더 자세한 내용은 Google 문서 gcloud command-line tool overview 그리고 gcloud compute ssh를 참조하십시오.

CPU 사용량 및 보고

CPU Utilization(CPU 사용률) 보고서에는 지정된 시간 내에 사용된 CPU의 백분율이 요약되어 있습니다. 일반적으로 코어는 사용량이 적은 시간에는 총 CPU 용량의 약 30~40%, 사용량이 많은 시간에는 약 60~70%로 작동합니다.

ASA Virtual의 vCPU 사용량

ASA virtual vCPU 사용량에서는 데이터 경로, 제어 지점, 외부 프로세스에 사용된 vCPU 양을 확인할 수 있습니다.

GCP에서 보고하는 vCPU 사용량에는 앞서 설명한 ASA virtual 사용량이 포함됩니다.

- ASA Virtual 유휴 시간
- ASA 가상 머신에 사용된 %SYS 오버헤드
- vSwitch, vNIC, pNIC 간 패킷 이동의 오버헤드. 이 오버헤드가 상당히 클 수 있습니다.

CPU 사용량의 예

show cpu usage 명령을 사용하여 CPU 사용률 통계를 표시할 수 있습니다.

예

```
Ciscoasa#show cpu usage

CPU utilization for 5 seconds = 1%; 1 minute: 2%; 5 minutes: 1%
```

다음은 보고된 vCPU 사용량이 상당한 차이를 보이는 예입니다.

- ASA Virtual 보고서: 40%
- DP: 35%

- 외부 프로세스: 5%
- ASA(ASA Virtual 보고서): 40%
- ASA 유휴 폴링: 10%
- 오버헤드: 45%

이 오버헤드는 하이퍼바이저 기능을 수행하고 vSwitch를 사용하여 NIC와 vNIC 간에 패킷을 이동하는 데 사용됩니다.

GCP CPU 사용량 보고

GCP 콘솔에서 인스턴스 이름을 클릭한 다음 **Monitoring(**모니터링**)** 탭을 클릭합니다. CPU 사용률을 확인할 수 있습니다.

Compute Engine에서는 사용량 내보내기 기능을 사용하여 Compute Engine 사용량에 관한 자세한 보고서를 Google Cloud Storage 버킷으로 내보낼 수 있습니다. 사용량 보고서는 리소스의 수명에 관한 정보를 제공합니다. 예를 들어 프로젝트에서 `n2-standard-4` 머신 유형을 실행 중인 VM 인스턴스의 수와 각 인스턴스가 실행된 기간을 확인할 수 있습니다. 또한 영구 디스크의 스토리지 공간과 다른 Compute Engine 기능에 대한 정보도 검토할 수 있습니다.

ASA Virtual 및 GCP 그래프

ASA Virtual과 GCP의 CPU % 수치가 다릅니다.

- GCP 그래프 수치가 항상 ASA Virtual 수치보다 높습니다.
- GCP에서는 이를 %CPU 사용량, ASA Virtual에서는 %CPU 활용이라고 부릅니다.

용어 “%CPU utilization”과 “%CPU usage”의 의미는 서로 다릅니다.

- CPU utilization은 물리적 CPU의 통계를 제공합니다.
- CPU usage는 논리적 CPU의 통계로서 CPU 하이퍼스레딩을 기반으로 합니다. 그러나 단 하나의 vCPU가 사용되므로 하이퍼스레딩은 켜져 있지 않습니다.

GCP는 CPU % 사용량을 다음과 같이 계산합니다.

활발하게 사용 중인 가상 CPU의 양 - 총 가용 CPU 기준 백분율로 표시

이 계산은 게스트 운영 체제가 아닌 호스트의 관점에서 본 CPU 사용량입니다. 그리고 가상 머신에 있는 사용 가능한 모든 가상 CPU의 평균 CPU 사용률입니다.

예를 들어, 가상 CPU 1개를 사용하는 가상 시스템이 4개의 물리적 CPU를 가진 호스트에서 실행되는 중이고 CPU usage가 100%라면 가상 머신에서 하나의 물리적 CPU를 온전히 사용하는 것입니다. 가상 CPU 사용량 계산: 사용량(MHz) / 가상 CPU 수 x 코어 주파수

13 장

GCP에 ASA 가상 Auto Scale 솔루션 구축

GCP의 ASA 가상용 Auto Scale 솔루션

다음 섹션에서는 GCP의 자동 확장 솔루션 구성 요소가 ASA 가상에서 작동하는 방식을 설명합니다.

Auto Scale 솔루션

ASA 가상 Auto Scale for GCP는 GCP에서 제공하는 서버리스 인프라(Cloud Functions, 로드 밸런서, Pub/Sub, 인스턴스 그룹 등)를 사용하는 완벽한 서버리스 구현입니다.

ASA 가상 Auto Scale for GCP 구현의 몇 가지 주요 기능은 다음과 같습니다.

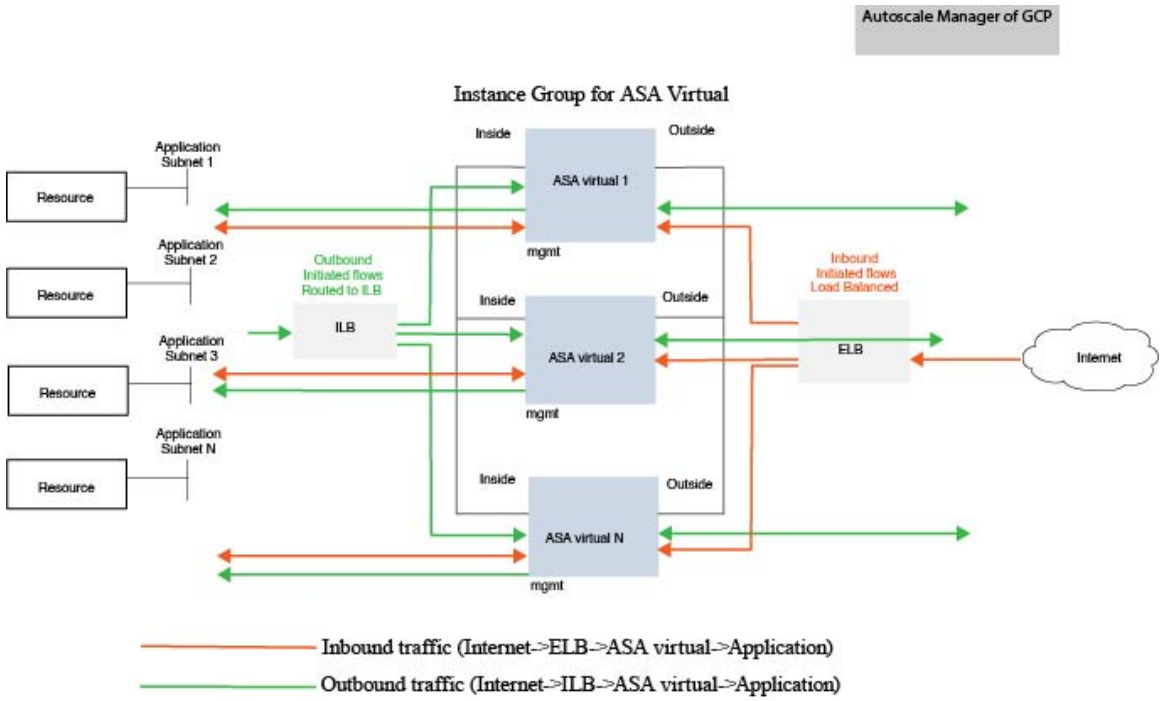
- GCP Deployment Manager 템플릿 기반 구축.
- CPU 기반의 메트릭 확장 지원.
- ASA 가상 구축 및 다중 가용성 영역 지원
- 확장된 ASA 가상 인스턴스에 자동으로 적용되는 완전히 자동화된 컨피그레이션.
- 로드 밸런서 및 다중 가용성 영역 지원
- Cisco에서는 구축을 쉽게 수행할 수 있도록 Auto Scale for GCP 구축 패키지를 제공합니다.

Auto Scale 사용 사례

ASA 가상 Auto Scale for GCP는 GCP Azure 내부 로드 밸런서(ILB)와 GCP 외부 로드 밸런서(ELB) 사이에 ASA 가상 인스턴스 그룹을 배치하는 자동화된 수평 확장 솔루션입니다.

- 인터넷에서 인스턴스 그룹의 ASA 가상 인스턴스로 트래픽을 분산합니다. 그러면 방화벽이 애플리케이션에 트래픽을 전달합니다.
- ILB는 애플리케이션의 아웃바운드 인터넷 트래픽을 인스턴스 그룹의 ASA 가상 인스턴스로 분산합니다. 그러면 방화벽이 트래픽을 인터넷으로 전달합니다.
- 네트워크 패킷은 단일 연결에서 내부 및 외부 로드 밸런서를 모두 통과하지 않습니다.
- 확장 집합의 ASA 가상 인스턴스 수는 로드 조건에 따라 자동으로 조정 및 구성됩니다.

그림 ***55: ASA*** 가상 ***Auto Scale*** 사용 사례



범위

이 문서에서는 ASA 가상 Auto Scale for GCP 솔루션의 서버리스 구성 요소를 구축하는 자세한 절차를 설명합니다.

중요
- 구축을 시작하기 전에 전체 문서를 읽어보십시오.
- 구축을 시작하기 전에 전제 조건이 충족되었는지 확인합니다.
- 여기에 설명된 대로 단계 및 실행 순서를 따라야 합니다.

구축 패키지 다운로드

ASA 가상 Auto Scale for GCP 솔루션은 GCP에서 제공하는 서버리스 인프라(Cloud Functions, 로드 밸런서, Pub/Sub, 인스턴스 그룹 등)를 사용하는 GCP 구축 관리자 템플릿 기반 구축입니다.

ASA 가상 Auto Scale for GCP 솔루션을 시작하는 데 필요한 파일을 다운로드합니다. 사용자 ASA 버전의 구축 스크립트 및 템플릿은 GitHub 리포지토리에서 제공됩니다.

주의 Cisco에서 제공하는 자동 확장용 구축 스크립트 및 템플릿은 오픈 소스 예시로 제공되며 일반적인 Cisco TAC 지원 범위에서는 다루지 않습니다.

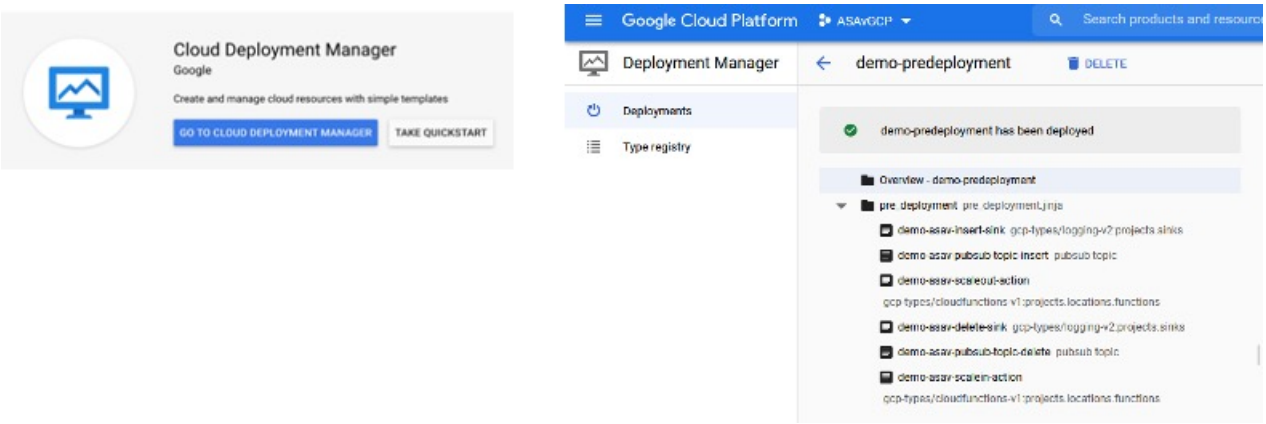
Auto Scale 솔루션 구성 요소

다음 구성 요소가 ASA 가상 Auto Scale for GCP 솔루션을 구성합니다.

구축 관리자

- 컨피그레이션을 코드로서 처리하고 반복 가능한 구축을 수행합니다. Google Cloud Deployment Manager를 사용하면 YAML을 이용해 애플리케이션에 필요한 모든 리소스를 선언적 형식으로 지정할 수 있습니다. 또한 Python 또는 Jinja2 템플릿을 사용하여 구성을 매개변수화하고 공통 구축 패러다임을 재사용할 수 있습니다.
- 리소스를 정의하는 구성 파일을 생성합니다. 이러한 리소스를 생성하는 프로세스를 반복하여 일관된 결과를 얻을 수 있습니다. 자세한 내용은 https://cloud.google.com/deployment-manager/docs를 참조하십시오.

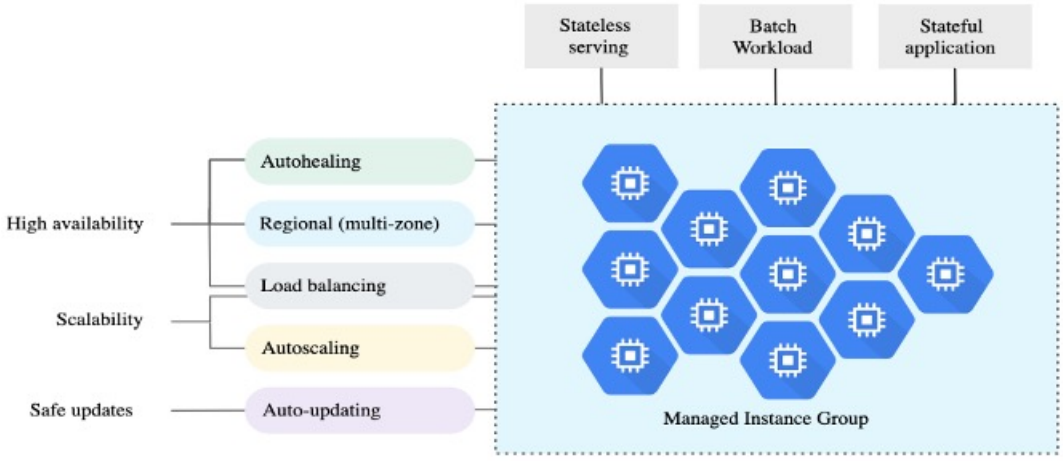
그림 *56: Deployment Manager* 보기



GCP의 Managed Instance Group

MIG(Managed Instance Group)는 사용자가 지정하는 인스턴스 템플릿 및 선택적 스테이트풀 구성을 기반으로 각 매니지드 인스턴스를 생성합니다. 자세한 내용은 https://cloud.google.com/compute/docs/instance-groups를 참조하십시오.

그림 *57:* 인스턴스 그룹 기능

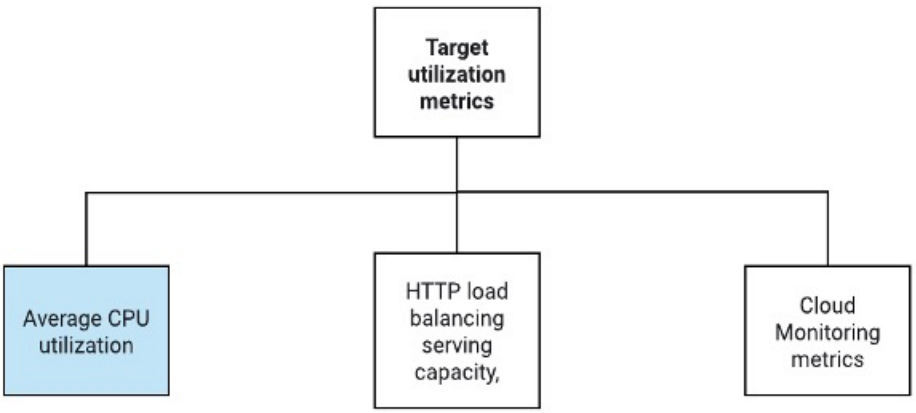


목표 사용률 메트릭

- 다음 다이어그램은 목표 사용률 메트릭을 보여줍니다. 자동 확장 결정에는 평균 CPU 사용률 메트릭만 사용합니다.
- Autoscaler는 선택한 사용률 메트릭을 기반으로 사용량 정보를 지속적으로 수집하고, 실제 사용률을 원하는 목표 사용률과 비교하며, 이 정보를 사용하여 그룹에서 인스턴스를 제거해야 하는지(축소) 추가해야 하는지(확장)를 결정합니다.

- 목표 사용률 수준은 VM(가상 머신) 인스턴스를 유지 관리할 수준입니다. 예를 들어 CPU 사용률을 기준으로 확장하는 경우 목표 사용률 레벨을 75%로 설정할 수 있으며, 이 경우 Autoscaler는 지정된 인스턴스 그룹의 CPU 사용률을 75% 이하로 유지합니다. 각 메트릭의 사용률 수준은 자동 확장 정책에 따라 다르게 해석됩니다. 자세한 내용은 https://cloud.google.com/compute/docs/autoscaler를 참조하십시오.

그림 *58:* 목표 사용률 메트릭



서버리스 클라우드 기능

Instance Group Manager에 인스턴스가 나타나면, 사용자는 SSH 비밀번호를 설정하고, 비밀번호를 활성화하고, 호스트 이름을 변경하기 위한 Google Cloud 기능을 사용합니다.

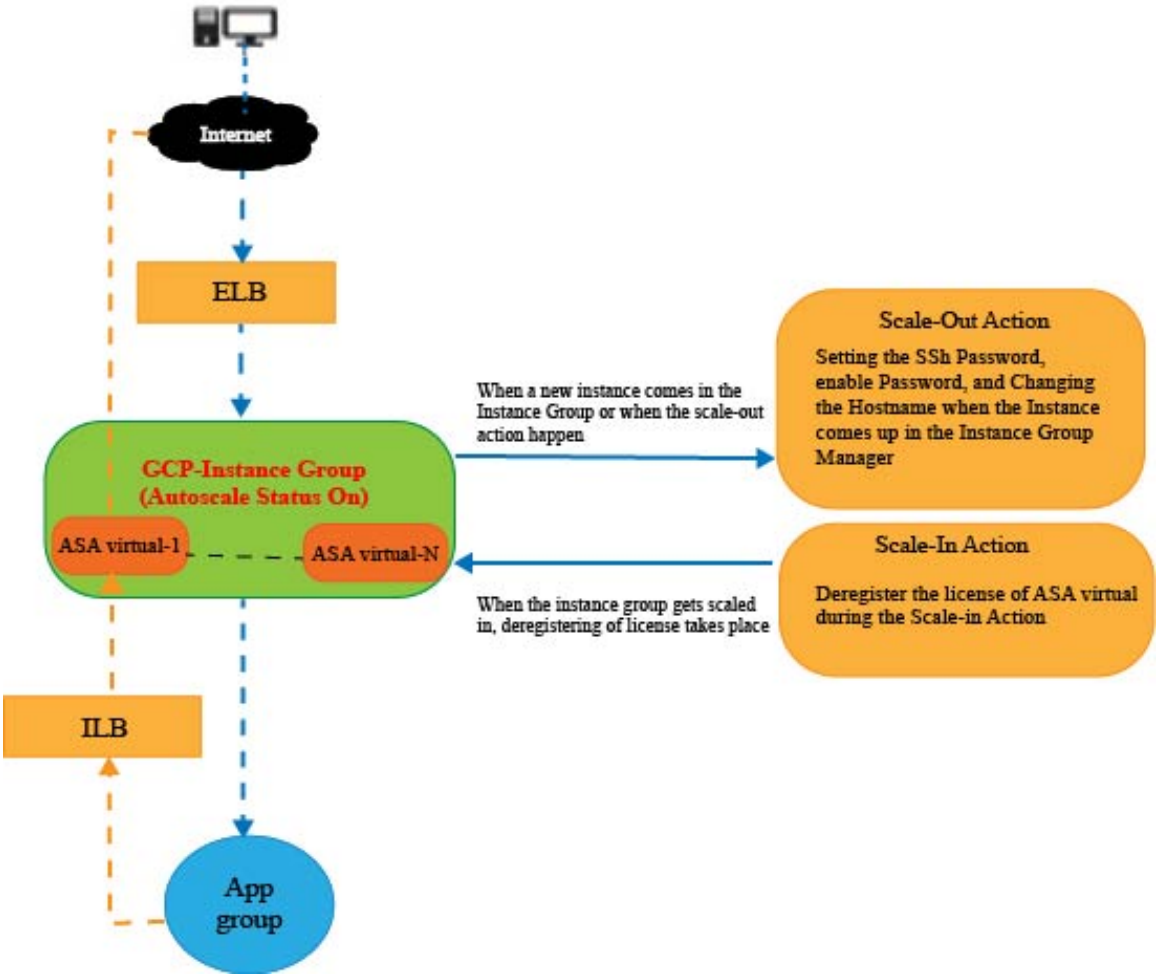
- 확장 중에 인스턴스 그룹에 새 ASA 가상 인스턴스가 나타나면, 확장 프로세스를 항상 모니터링할 수는 없으므로 SSH 비밀번호를 설정하고 비밀번호를 활성화한 다음 호스트 이름을 변경해야 합니다.
- 클라우드 기능은 확장 프로세스 중에 Cloud Pub/Sub 주제를 통해 트리거됩니다. 확장하는 동안 인스턴스를 추가할 때만 사용 가능한 필터가 있는 Log Sink도 있습니다.

Cloud Functions를 사용하여 서버리스 라이선스 등록 해제

- 축소 중에 인스턴스가 삭제되는 동안, ASA 가상 인스턴스에서 라이선스를 등록 해제해야 합니다.
- 클라우드 기능은 Cloud Pub/Sub 주제를 통해 트리거됩니다. 특히 삭제 프로세스의 경우, 축소하는 동안 인스턴스 삭제에만 사용 가능한 필터가 있는 Log Sink가 있습니다.
- Cloud Function은 트리거되면 SSH를 통해 삭제 중인 ASA 가상 인스턴스에 SSH를 수행하고 라이선스 등록 취소를 위한 명령을 실행합니다.

Autoscale 솔루션의 대략적인 개요

그림 *59: Autoscale* 솔루션 개요



Auto Scale 솔루션 사전 요건

GCP 리소스

GCP 프로젝트

이 솔루션의 모든 구성 요소를 구축하려면 기존 또는 새로 생성된 프로젝트가 필요합니다.

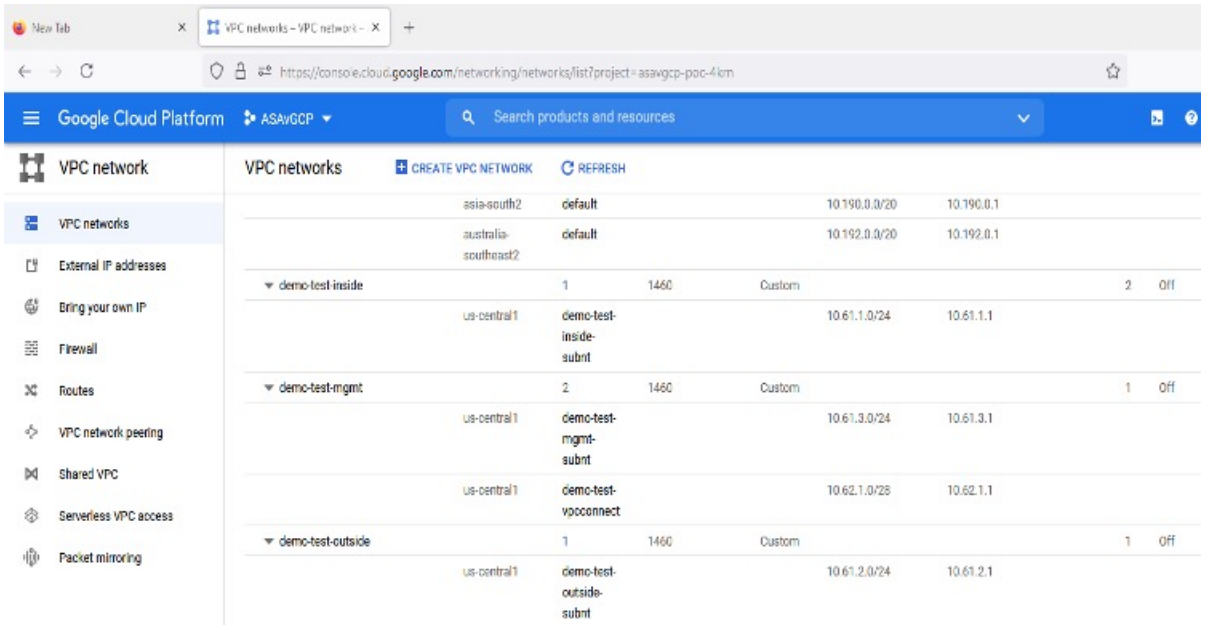
네트워킹

VPC 3개가 사용 가능한지/생성되었는지 확인합니다. Auto Scale 구축에서는 네트워킹 리소스를 생성, 변경 또는 관리하지 않습니다.

ASA 가상에는 네트워크 인터페이스 3개가 필요하므로 가상 네트워크에는 다음을 위한 서브넷 3개가 필요합니다.

- 관리 트래픽
- 내부 트래픽
- 외부 트래픽

그림 ***60: VPC*** 네트워크 보기



방화벽

VPC 간 통신을 허용하고 상태 프로브도 허용하는 방화벽 규칙을 만들어야 합니다. 나중에 구축 관리자 템플릿에서 사용할 방화벽 태그를 기록해 두어야 합니다.

서브넷이 연결된 네트워크 보안 그룹에서 다음 포트를 열어야 합니다.

- SSH(TCP/22) - 로드 밸런서와 ASA 가상 사이의 상태 프로브에 필요합니다. 서버리스 함수와 ASA 가상 간의 통신에 필요합니다.
- 애플리케이션 전용 프로토콜/포트 - 모든 사용자 애플리케이션(예: TCP/80)에 필요합니다.

ASA 컨피그레이션 파일 준비

deployment manager jinja 컨피그레이션 파일에 넣을 ASA 가상 컨피그레이션 파일을 준비합니다. 이 컨피그레이션은 ASA 가상 프로젝트의 인스턴스 템플릿에서 시작 스크립트로 사용합니다.

컨피그레이션 파일에는 최소한 다음이 포함되어야 합니다.

- 모든 인터페이스에 DHCP IP 할당을 설정합니다.

- GCP 로드 밸런서는 nic0으로만 트래픽을 전달하므로 Nic0은 'outside(외부)'로 표시되어야 합니다.
- Nic0은 IP 전달만 지원하므로 ASA 가상에 대한 SSH에 사용됩니다.
- ASA 컨피그레이션의 외부 인터페이스에서 SSH를 활성화합니다.
- 외부에서 내부 인터페이스로 트래픽을 전달하도록 NAT 컨피그레이션을 생성합니다.
- 원하는 트래픽을 허용하는 액세스 정책을 생성합니다.
- 리소스 상태의 경우, 적절한 NAT 규칙을 사용하여 관련 상태 프로브를 메타데이터 서버로 리디렉션해야 합니다.

다음은 참조용으로만 사용할 수 있는 샘플 ASA 컨피그레이션 파일입니다.

```
!ASA Version 9.15.1.10
!Interface Config
interface G0/0
nameif inside
security-level 100
ip address dhcp setroute
no shutdown


interface G0/1
nameif management
security-level 50
ip address dhcp setroute
no shutdown

interface M0/0
no management-only
nameif outside
security-level 0
ip address dhcp setroute
no shutdown
!
same-security-traffic permit inter-interface
!
!Due to some constraints in GCP,
!"GigabitEthernet0/0" will be used as a Management interface
!"Management0/0" will be used as a data interface
crypto key generate rsa modulus 2048
ssh 0.0.0.0 0.0.0.0 management
ssh version 2
ssh timeout 60
aaa authentication ssh console LOCAL
ssh authentication publickey {{ properties["publicKey"] }}
username admin privilege 15
username admin attributes
service-type admin

! required config end
dns domain-lookup management
dns server-group DefaultDNS
name-server 8.8.8.8
!
access-list all extended permit ip any any
access-list out standard permit any4
access-group all global
```

```
! Objects
object network metadata
host 169.254.169.254
object network ilb
host $(ref.{{ properties["resourceNamePrefix"] }}-ilb-ip.address)
object network hc1
subnet 35.191.0.0 255.255.0.0
object network hc2
subnet 130.211.0.0 255.255.63.0
object network elb
host $(ref.{{ properties["resourceNamePrefix"] }}-elb-ip.address)
object network appServer
host 10.61.2.3
object network defaultGateway
subnet 0.0.0.0 0.0.0.0
! Nat Rules
nat (inside,outside) source dynamic hc1 ilb destination static ilb metadata
nat (inside,outside) source dynamic hc2 ilb destination static ilb metadata
nat (inside,outside) source dynamic defaultGateway interface
 !
 object network appServer
nat (inside,outside) static $(ref.{{ properties["resourceNamePrefix"] }}-elb-ip.address)
object network defaultGateway
nat (outside,inside) dynamic interface
! Route Add
route inside 0.0.0.0 0.0.0.0 10.61.1.1 2
route management 0.0.0.0 0.0.0.0 10.61.3.1 3
license smart register idtoken <licenseIDToken>
```

GCP 클라우드 기능 패키지 구축

ASA 가상 GCP Auto Scale 솔루션에서는 클라우드 기능을 압축된 ZIP 패키지 형식으로 전달하는 아카이브 파일 2개를 빌드해야 합니다.

- `scalein-action.zip`
- `scaleout-action.zip`

`scalein-action.zip` 및 `scaleout-action.zip` 패키지를 구축하는 자세한 방법은 Auto Scale 구축 지침을 참조하십시오.

이러한 기능은 특정 작업을 수행하기 위해 가능한 한 개별적이며, 개선 사항 및 새로운 릴리스 지원을 위해 필요에 따라 업그레이드할 수 있습니다.

입력 매개변수

다음 표에서는 템플릿 매개 변수를 정의하고 일 예를 제공합니다. 이러한 값을 결정하고 나면 GCP 프로젝트에 GCP 템플릿을 구축할 때 이러한 매개변수를 사용하여 ASA 가상 디바이스를 생성할 수 있습니다.

표 25: 템플릿 매개변수

매개 변수 이름	허용되는 값 / 유형	설명	리소스 생성 유형
resourceNamePrefix	문자열	모든 리소스는 이 접두사를 포함하는 이름으로 생성됩니다. 예: demo-test	신규
region	GCP에서 지원하는 유효한 지역 [문자열]	프로젝트가 구축될 지역의 이름입니다. 예: us-central1	
serviceAccountMailId	문자열 [이메일 ID]	서비스 계정을 식별하는 이메일 주소입니다.	
vpcConnectorName	문자열	서버리스 환경과 VPC 네트워크 간의 트래픽을 처리하는 커넥터의 이름입니다. 예: demo-test-vpc-connector	
bucketName	문자열	클라우드 함수 ZIP 패키지가 업로드될 GCP 스토리지 버킷의 이름입니다. 예: demo-test-bkt	
cpuUtilizationTarget	10진수 (0,1]	Autoscaler가 유지해야 하는 인스턴스 그룹에 있는 VM의 평균 CPU 사용률입니다. 예: 0.5	
healthCheckFirewallRuleName	문자열	상태 검사 프로브 IP 범위의 패킷을 허용하는 방화벽 규칙의 태그입니다. 예: demo-test-healthallowall	기존

매개 변수 이름	허용되는 값 / 유형	설명	리소스 생성 유형
insideFirewallRuleName	문자열	내부 VPC에서 통신을 허용하는 방화벽 규칙의 태그입니다. 예: demo-test-inside-allowall	기존
insideVPCName	문자열	내부 VPC의 이름입니다. 예: demo-test-inside	기존
insideVPCSubnet	문자열	내부 서브넷의 이름입니다. 예: demo-test-inside-subnt	기존
머신 유형	문자열	ASA 가상 VM의 머신 유형입니다. 예: e2-standard-4	
maxASACount	정수	인스턴스 그룹에서 허용되는 최대 ASA 가상 인스턴스 수입니다. 예: 3	
mgmtFirewallRuleName	문자열	관리 VPC에서의 통신을 허용하는 방화벽 규칙의 태그입니다. 예: demo-test-mgmt-allowall	
mgmtVPCName	문자열	관리 VPC의 이름입니다. 예: demo-test-mgmt	
mgmtVPCSubnet	문자열	관리 서브넷의 이름입니다. 예: demo-test-mgmt-subnt	

매개 변수 이름	허용되는 값 / 유형	설명	리소스 생성 유형
minASACount	정수	지정된 시간에 인스턴스 그룹에서 사용 가능한 최소 ASA 가상 인스턴스 수입니다. 예: 1	
outsideFirewallRuleName	문자열	외부 VPC에서의 통신을 허용하는 방화벽 규칙의 태그입니다. 예: demo-test-outside-allowall	
outsideVPCName	문자열	외부 VPC의 이름입니다. 예: demo-test-outside	
outsideVPCSubnet	문자열	외부 서브넷의 이름입니다. 예: demo-test-outside-subnt	
publicKey	문자열	ASA 가상 VM의 SSH 키입니다.	
sourceImageURL	문자열	프로젝트에서 사용할 ASA 가상의 이미지입니다. 예: https://www.googleapis.com/compute/v1/projects/cisco-public/global/images/cisco-asav-9-15-1-15	
애플리케이션 서버 IP 주소	문자열	내부 Linux 머신의 내부 IP 주소입니다. 예: 10.61.1.2	
내부 VPC 게이트웨이 IP 주소	문자열	내부 VPC의 게이트웨이입니다. 예: 10.61.1.1	

매개 변수 이름	허용되는 값 / 유형	설명	리소스 생성 유형
관리 VPC 게이트웨이 IP 주소	문자열	관리 VPC의 게이트웨이입니다. 예: 10.61.3.1	

Auto Scale 솔루션 구축

단계 **1** 로컬 폴더에 Git 리포지토리를 복제합니다.

```
git clone git_url -b branch_name
```

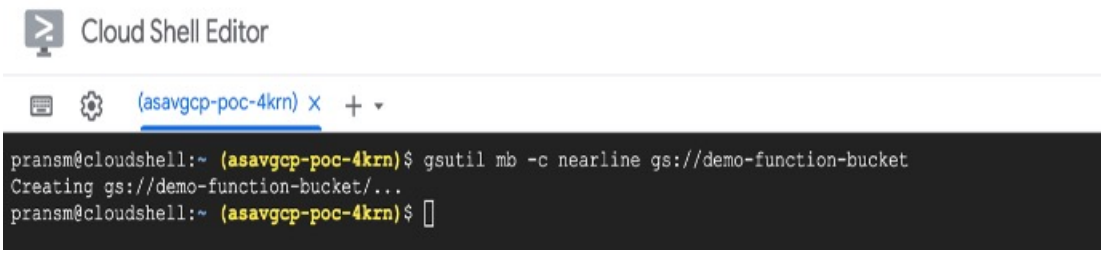
예제:

```
Last login: Thu Jun  3 13:01:32 on ttys002
(base) pransm@PRANSM-M-F9KA ~ % git clone https://bitbucket-eng-bgl1.cisco.com/bitbucket/scm/vcb/cloud_autoscale.git -b saaarwar_asa_autoscale_public_key
Cloning into 'cloud_autoscale'...
remote: Enumerating objects: 1604, done.
remote: Counting objects: 100% (1604/1604), done.
remote: Compressing objects: 100% (1507/1507), done.
remote: Total 1604 (delta 759), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (1604/1604), 58.35 MiB | 8.54 MiB/s, done.
Resolving deltas: 100% (759/759), done.
(base) pransm@PRANSM-M-F9KA ~ %
```

단계 **2** gcloud CLI에서 버킷을 생성합니다.

```
gsutil mb -c nearline gs://bucket_name
```

예제:



단계 **3** 압축된 Zip 패키지를 빌드합니다.

a) `scalein_action` 및 `scaleout_action` 폴더에서 다음 파일로 구성된 압축된 Zip 패키지를 생성합니다.

- main.py
- basic_functions.py

requirements.txt

b) 압축된 Zip 패키지의 이름을 `scaleout-action.zip` 및 `scalein-action.zip`으로 변경합니다.

참고 폴더 내부로 이동하여 파일을 선택하고 마우스 오른쪽 버튼으로 클릭한 다음 'compress | archive'를 선택하여, GCP가 읽을 수 있는 .zip 파일을 만듭니다.

단계 4 압축된 Zip 패키지(`scaleout-action.zip` 및 `scalein-action.zip`)를 Cloud Editor 작업 공간에 업로드합니다.

단계 5 구축 관리자 템플릿의 파일을 Cloud Editor 작업 공간에 업로드합니다.

- asav_autoscale.jinja
- asav_autoscale_params.yaml
- pre_deployment.jinja
- pre_deployment.yaml

단계 6 압축된 Zip 패키지를 버킷 스토리지에 복사합니다.

- `gsutil cp scaleout-action.zip gs://bucket_name`
- `gsutil cp scalein-action.zip gs://bucket_name`

예제:

```
pransm@cloudshell:~ (asavgcp-poc-4krn)$ gsutil cp scaleout-action.zip gs://demo-function-bucket
Copying file://scaleout-action.zip [Content-Type=application/zip]...
/ [1 files][  3.3 KiB/  3.3 KiB]
Operation completed over 1 objects/3.3 KiB.
pransm@cloudshell:~ (asavgcp-poc-4krn)$ gsutil cp scalein-action.zip gs://demo-function-bucket
Copying file://scalein-action.zip [Content-Type=application/zip]...
/ [1 files][  3.3 KiB/  3.3 KiB]
Operation completed over 1 objects/3.3 KiB.
pransm@cloudshell:~ (asavgcp-poc-4krn)$
```

단계 7 내부, 외부 및 관리 인터페이스용 VPC와 서브넷을 생성합니다.

관리 VPC에는 /28 서브넷이 있어야 합니다(예: 10.8.2.0/28).

단계 8 내부, 외부 및 관리 인터페이스를 위해 방화벽 규칙 3개가 필요합니다. 상태 검사 프로브를 허용하는 방화벽 규칙도 있어야 합니다.

단계 9 사전 구축 및 ASA 가상 자동 확장 구축을 위한 Jinja 및 YAML 파일에서 매개변수를 업데이트합니다.

a) `asav_autoscale_params.yaml` 파일을 열고 다음 매개변수를 업데이트합니다.

- **resourceNamePrefix**: <resourceNamePrefix>
- **region**: <region>
- **serviceAccountMailId**: <serviceAccountMailId>
- **publicKey**: <publicKey>
- **insideVPCName**: <Inside-VPC-Name>
- **insideVPCSubnet**: <Inside-VPC-Subnet>
- **outsideVPCName**: <Outside-VPC-Name>
- **outsideVPCSubnet**: <Outside-VPC-Subnet>
- **mgmtVPCName**: <Mgmt-VPC-Name>

- **mgmtVPCSubnet**: <Mgmt-VPC-Subnet>
- **insideFirewallRuleName**: <Inside-Network-Firewall-Tag>
- **outsideFirewallRuleName**: <Outside-Network-Firewall-Tag>
- **mgmtFirewallRuleName**: <Mgmt-Network-Firewall-Tag>
- **healthCheckFirewallRuleName**: <HealthCheck-IP-Firewall-Tag>
- **machineType**: <machineType>

참고 ASA 가상 Auto Scale의 경우 **cpuUtilizationTarget: 0.5** 매개변수가 설정되며, 요구 사항에 맞게 수정할 수 있습니다.

이 값은 모든 ASA 가상 인스턴스 그룹의 CPU 사용량이 50%임을 나타냅니다.

b) `asav_autoscale.jinja` 파일을 열고 다음 매개변수를 업데이트합니다.

- **host**: <Application server IP address>
- **route inside 0.0.0.0 0.0.0.0**: <Inside VPC Gateway IP address> 2
- **route management 0.0.0.0 0.0.0.0**: <Management VPC Gateway IP address> 3
- **license smart register idtoken**: <licenseIDToken>

c) `pre_deployment.yaml` 파일을 열고 다음 매개변수를 업데이트합니다.

- **resourceNamePrefix**: <resourceNamePrefix>
- **region**: <region>
- **serviceAccountMailId**: <serviceAccountMailId>
- **vpcConnectorName**: <VPC-Connector-Name>
- **bucketName**: <bucketName>

단계 **10** Secret Manager GUI를 사용하여 다음을 위한 3가지 암호를 생성합니다. https://console.cloud.google.com/security/secret-manager의 내용을 참조하십시오.

- asav-en-password
- asav-new-password
- asav-private-key


Secret Manager lets you store, manage, and secure access to your application secrets.
Learn more

Filter Enter property name or value

Name	Location	Encryption	Labels	Created	Expiration	Actions
asav-en-password	Automatically replicated	Google-managed	None	4/26/21, 3:35 PM		
asav-new-password	Automatically replicated	Google-managed	None	4/26/21, 3:36 PM		
asav-private-key	Automatically replicated	Google-managed	None	4/26/21, 3:35 PM		

단계 **11** VPC 커넥터를 만듭니다.

```
gcloud beta compute networks vpc-access connectors create <vpc-connector-name>
--region <region> --subnet=</28 subnet name>
```

예제:

```
gcloud beta compute networks vpc-access connectors create demo-vpc-connector
--region us-central1 --subnet=outside-connect-28
Create request issued for: [demo-vpc-connector]
Waiting for operation [projects/asavgcp-poc-4krn/locations/us-central1/operations/
10595de7-837f-4c19-9396-0c22943ecf15] to complete...done.
Created connector [demo-vpc-connector].
```

단계 **12** 사전 구축 YAML 컨피그레이션을 구축합니다.

```
gcloud deployment-manager deployments create <pre-deployment-name>
--config pre_deployment.yaml
```

예제:

```
gcloud deployment-manager deployments create demo-predeployment
--config pre_deployment.yaml
The fingerprint of the deployment is b'9NOy0gsTPgg16SqUEVsBjA=='
Waiting for create [operation-1624383045917-5c55e266e596d-4979c5b6-66d1025c]...done.
Create operation operation-1624383045917-5c55e266e596d-4979c5b6-66d1025c
completed successfully
```

NAME	TYPE	STATE
demo-asav-delete-sink	gcp-types/logging-v2:projects.sinks	COMPLETED
demo-asav-insert-sink	gcp-types/logging-v2:projects.sinks	COMPLETED
demo-asav-pubsub-topic-delete	pubsub.v1.topic	COMPLETED
demo-asav-pubsub-topic-insert	pubsub.v1.topic	COMPLETED
demo-asav-scalein-action	gcp-types/cloudfunctions-v1:projects.locations.functions	COMPLETED
demo-asav-scaleout-action	gcp-types/cloudfunctions-v1:projects.locations.functions	COMPLETED

단계 **13** ASA 가상 Auto Scale 구축을 만듭니다.

```
gcloud deployment-manager deployments create <deployment-name>
--config asav_autoscale_params.yaml
```

예제:

```
gcloud deployment-manager deployments create demo-asav-autoscale
--config asav_autoscale_params.yaml
The fingerprint of the deployment is b'1JCQi7Il-laWOY7vOLza0g=='
Waiting for create [operation-1624383774235-5c55e51d79d01-1a3acf92-4f3daf16]...done.
Create operation operation-1624383774235-5c55e51d79d01-1a3acf92-4f3daf16
completed successfully.
```

NAME	TYPE	STATE
demo-asav-autoscaler	compute.v1.regionAutoscaler	COMPLETED
demo-asav-backend-service-elb	compute.v1.regionBackendService	COMPLETED
demo-asav-backend-service-ilb	compute.v1.regionBackendService	COMPLETED
demo-asav-fr-elb	compute.v1.forwardingRule	COMPLETED
demo-asav-fr-ilb	compute.v1.forwardingRule	COMPLETED
demo-asav-hc-elb	compute.v1.regionHealthChecks	COMPLETED
demo-asav-hc-ilb	compute.v1.healthCheck	COMPLETED
demo-asav-health-check	compute.v1.healthCheck	COMPLETED
demo-asav-instance-group	compute.v1.regionInstanceGroupManager	COMPLETED
demo-asav-instance-template	compute.v1.instanceTemplate	COMPLETED
demo-elb-ip	compute.v1.address	COMPLETED

단계 **14** ILB가 내부 애플리케이션에서 인터넷으로 패킷을 전달할 경로를 만듭니다.

```
gcloud beta compute routes create <ilb-route-name>
--network=<inside-vpc-name> --priority=1000 --destination-range=0.0.0.0/0
--next-hop-ilb=<ilb-forwarding-rule-name> --next-hop-ilb-region=<region>
```

예제:

```
gcloud beta compute routes create demo-ilb --network=sdt-test-asav-inside
--priority=1000 --destination-range=0.0.0.0/0 --next-hop-ilb=demo-asav-fr-ilb
--next-hop-ilb-region=us-central1
Created [https://www.googleapis.com/compute/beta/projects/asavgcp-poc-4krn/global
/routes/demo-ilb].
```

NAME	NETWORK	DEST_RANGE	NEXT_HOP	PRIORITY
demo-ilb	sdt-test-asav-inside	0.0.0.0/0	10.7.1.60	1000

단계 **15** Cloud Router 및 Cloud NAT를 만듭니다.

```
gcloud compute routers create <cloud-router-name>
--project=<project-name> --region <region> --network=<outside-vpc-name>
--advertisement-mode=custom
```

```
gcloud compute routers nats create <cloud-nat-name>
--router=<cloud-router-name> --nat-all-subnet-ip-ranges --auto-allocate-nat-external-ips
--region=<region>
```

예제:

```
gcloud compute routers create demo-cloud-router --project=asavgcp-poc-4krn
--region us-central1 --network=sdt-test-asav-outside --advertisement-mode=custom
Creating router [demo-cloud-router]...done.
```

NAME	REGION	NETWORK
demo-cloud-router	us-central1	sdt-test-asav-outside

```
gcloud compute routers nats create demo-cloud-nat
--router=demo-cloud-router --nat-all-subnet-ip-ranges
--auto-allocate nat-external-ips --region=us-central1
Creating NAT [demo-cloud-nat] in router [demo-cloud-router]...done.
```

Auto Scale 논리

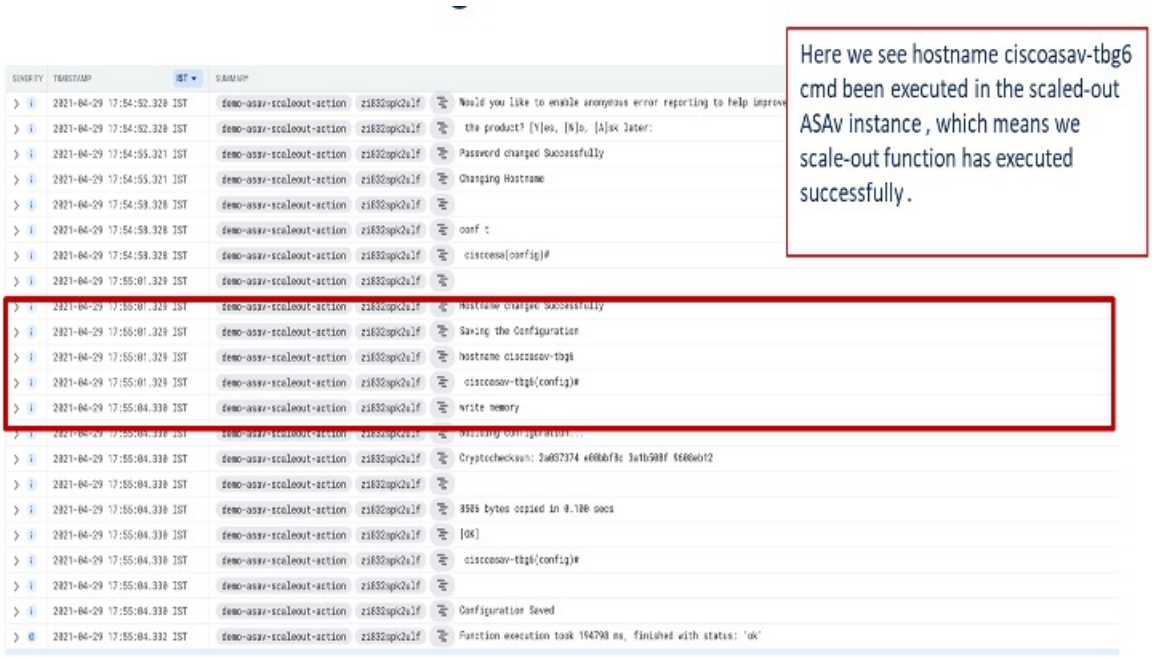
- 자동 확장기는 목표 CPU 사용률을 인스턴스 그룹에서 시간 경과에 따른 모든 vCPU의 평균 사용량의 일부로 취급합니다.
- 총 vCPU의 평균 사용률이 목표 사용률을 초과하면, 자동 확장기는 VM 인스턴스를 추가합니다. 총 vCPU의 평균 사용률이 목표 사용률 미만이라면, 자동 확장기는 인스턴스를 제거합니다.
- 예를 들어 목표 사용률을 0.75로 설정하면 자동 확장기는 인스턴스 그룹 내 모든 vCPU의 평균 사용률을 75%로 유지합니다.
- 확장 결정에는 CPU 사용률 메트릭만 사용합니다.
- 위의 논리는 로드 밸런서가 모든 ASAv에 연결을 동일하게 분산하려 한다는 가정을 기반으로 하며, 평균적으로 모든 ASAv를 동일하게 로드해야 합니다.

Auto Scale 로깅 및 디버깅

클라우드 기능의 로그는 다음과 같이 볼 수 있습니다.

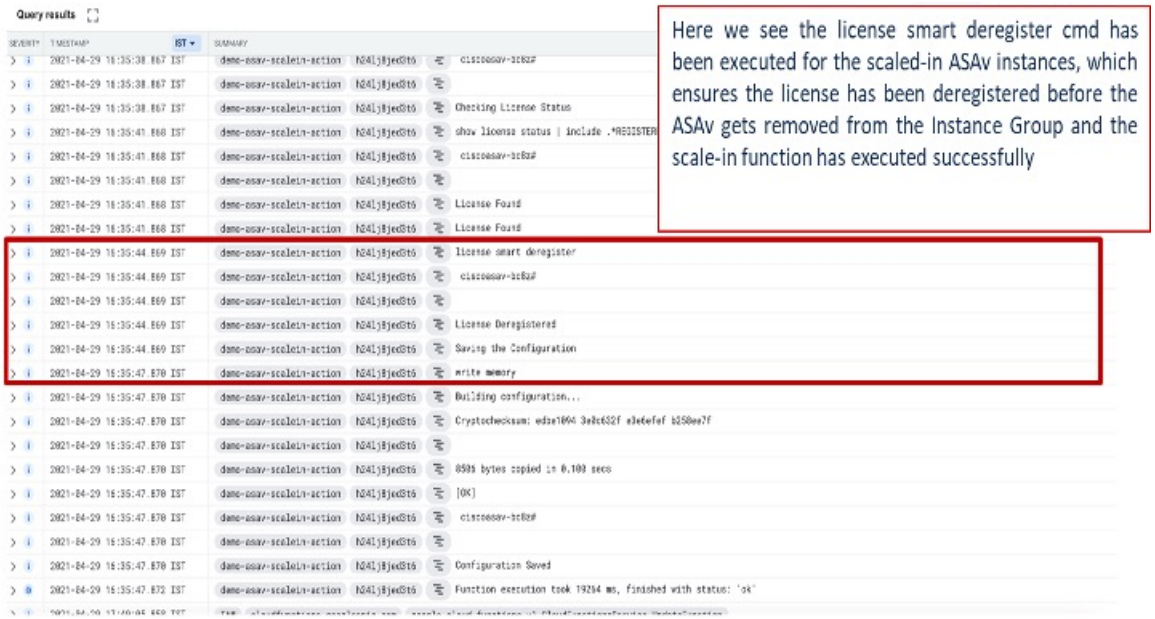
- 확장 기능 로그

  그림 ***61:*** 확장 기능 로그



- 축소 기능 로그

그림 62: 축소 기능 로그



Auto Scale 지침 및 제한 사항

- IPv4만 지원됩니다.
- 지원되는 라이선싱은 BYOL 전용입니다. PAYG는 GCP의 ASA 가상에서는 사용할 수 없습니다.
- 외부 로드 밸런서는 템플릿에 의해 생성되므로, 로드 밸런서의 공용 IP에 대한 특정 DNS 요구 사항은 범위를 벗어납니다.
- 애플리케이션이 사용자가 생성한 로드 밸런서 뒤에 있으며, ASA 가상은 모든 트래픽을 (트래픽을 특정 애플리케이션 IP로 직접 전송하는 대신) 이 로드 밸런서로 라우팅한다고 가정합니다.
- 태그, 리던던시 및 로드 밸런서 선호도 구성의 필요성에 대한 세부 정보는 고려되지 않습니다.
- ASA 가상 자격 증명은 다음과 같이 표시됩니다.
  - 서버리스 코드의 텍스트를 지웁니다.
  - 인스턴스 그룹의 모든 인스턴스에 표시됩니다.
  - 공유 GCP 계정을 사용하는 경우 인스턴스 템플릿에 표시됩니다.

민감한 데이터는 GCP의 공개 키 서비스를 사용하여 보호할 수 있습니다.

중요 Cisco에서는 라이선싱 서버를 이용한 ASA 가상 등록을 주기적으로 추적하여 확장 ASA가 라이선싱 서버에 정상적으로 등록되고 있는지, 그리고 축소 ASA 가상 인스턴스가 라이선스 서버에서 제거되는지 확인하는 것을 권장합니다.

Auto Scale 문제 해결

다음은 일반적인 오류 시나리오 및 ASA 가상 Auto Scale for GCP에 대한 디버깅 팁입니다.

- `main.py`를 찾을 수 없음 - Zip 패키지가 파일을 통해서만 생성되었는지 확인합니다. 클라우드 기능으로 이동하여 파일 트리를 확인할 수 있습니다. 어떤 폴더도 없어야 합니다.
- 템플릿을 구축하는 동안 오류 발생 - "<>" 안에 jimja 및 .yaml을 포함한 모든 매개변수 값이 입력되어 있으며, 동일한 이름의 구축이 이미 존재하는지 확인하십시오.
- Google Function에서 ASA 가상에 연결할 수 없음 - VPC 커넥터가 생성되었으며 YAML 매개변수 파일에 동일한 이름이 언급되는지 확인합니다.
- ASA 가상 SSH 과정에서 인증 실패 - 공개 및 개인 키 쌍이 올바른지 확인합니다.
- 라이선스 등록 실패 - 라이선스 ID 토큰이 올바른지 확인합니다. 또한 클라우드 NAT가 생성되었으며 ASA 가상에서 tools.cisco.com에 연결할 수 있는지 확인합니다.

14 장

OpenStack에 ASA 가상 구축

OpenStack에 ASA 가상을 구축할 수 있습니다.

OpenStack에서의 ASA 가상 구축 정보

OpenStack 환경에서 ASA 가상을 구축할 수 있습니다. OpenStack은 퍼블릭 및 프라이빗 클라우드용 클라우드 컴퓨팅 플랫폼을 구축 및 관리하기 위한 소프트웨어 툴 집합으로, KVM 하이퍼바이저와 긴밀하게 통합되어 있습니다.

ASA 가상에 대한 OpenStack 플랫폼 지원을 활성화하면 오픈 소스 클라우드 플랫폼에서 ASA 가상을 실행할 수 있습니다. OpenStack은 KVM 하이퍼바이저를 사용하여 가상 리소스를 관리합니다. ASA 가상 디바이스는 KVM 하이퍼바이저에서 이미 지원됩니다. 따라서 추가 커널 패키지 또는 드라이버 없이도 OpenStack 지원을 활성화할 수 있습니다.

ASA 가상 및 OpenStack에 대한 사전 요건

- software.cisco.com에서 ASA 가상 qcow2 파일을 다운로드하고 Linux 호스트에 넣습니다.

  http://www.cisco.com/go/asa-software

- ASA 가상은 오픈 소스 OpenStack 환경 및 Cisco VIM 관리 OpenStack 환경에서의 구축을 지원합니다.

  OpenStack 지침에 따라 OpenStack 환경을 설정합니다.

  - 오픈 소스 OpenStack 문서를 참조하십시오.

    Stein 릴리스 - https://docs.openstack.org/project-deploy-guide/openstack-ansible/stein/overview.html

Queens 릴리스 - https://docs.openstack.org/project-deploy-guide/openstack-ansible/queens/overview.html

- Cisco VIM(Virtualized Infrastructure Manager) OpenStack 문서: Cisco Virtualized Infrastructure Manager 설명서, 3.4.3~3.4.5를 참조하십시오.

- ASA 가상에 라이선스를 부여합니다. ASA 가상 라이선스를 등록하기 전에는 저성능 모드에서 실행됩니다. 이 모드에서는 100개의 연결 및 100Kbps의 처리량만 허용됩니다. 라이선스: 스마트 소프트웨어 라이선싱을 참조하십시오.
- 인터페이스 요건:
  - 관리 인터페이스
  - 내부 및 외부 인터페이스
- 통신 경로:
  - 관리 인터페이스 - ASA 가상을 ASDM에 연결할 때 사용합니다. 트래픽에는 사용할 수 없습니다.
  - 내부 인터페이스(필수)—ASA 가상를 내부 호스트에 연결하는 데 사용합니다.
  - 외부 인터페이스(필수)—ASA 가상를 공용 네트워크에 연결하는 데 사용합니다.
- 통신 경로:
  - ASA 가상에 액세스하기 위한 부동 IP.
- 최소 지원 ASA 가상 버전:
  - ASA 9.16.1
- OpenStack 요구 사항은 OpenStack 요구 사항을 참조하십시오.
- ASA 가상 시스템 요구 사항은 Cisco Secure Firewall ASA호환성을 참조하십시오.

ASA 가상 및 OpenStack에 대한 지침 및 제한 사항

지원 기능

OpenStack의 ASA 가상는 다음 기능을 지원합니다.

- OpenStack 환경의 컴퓨팅 노드에서 실행 중인 KVM 하이퍼바이저의 ASA 가상 구축.
- OpenStack CLI
- Heat 템플릿 기반 구축
- OpenStack Horizon 대시보드

- 라우팅 모드(기본값)
- 라이선싱 – BYOL만 지원됩니다.
- CLI 및 ASDM을 사용하여 ASA 가상 관리
- 드라이버 - VIRTIO, VPP 및 SRIOV

지원되지 않는 기능

OpenStack의 ASA 가상는 다음을 지원하지 않습니다.

- 자동 확장
- OpenStack Stein 및 Queens 릴리스 이외의 OpenStack 릴리스
- Ubuntu 18.04 버전 및 RHEL(Red Hat Enterprise Linux) 7.6 이외의 운영 체제

OpenStack 요구 사항

OpenStack 환경은 다음의 지원되는 하드웨어 및 소프트웨어 요구 사항을 준수해야 합니다.

표 *26:* 하드웨어 및 소프트웨어 요건

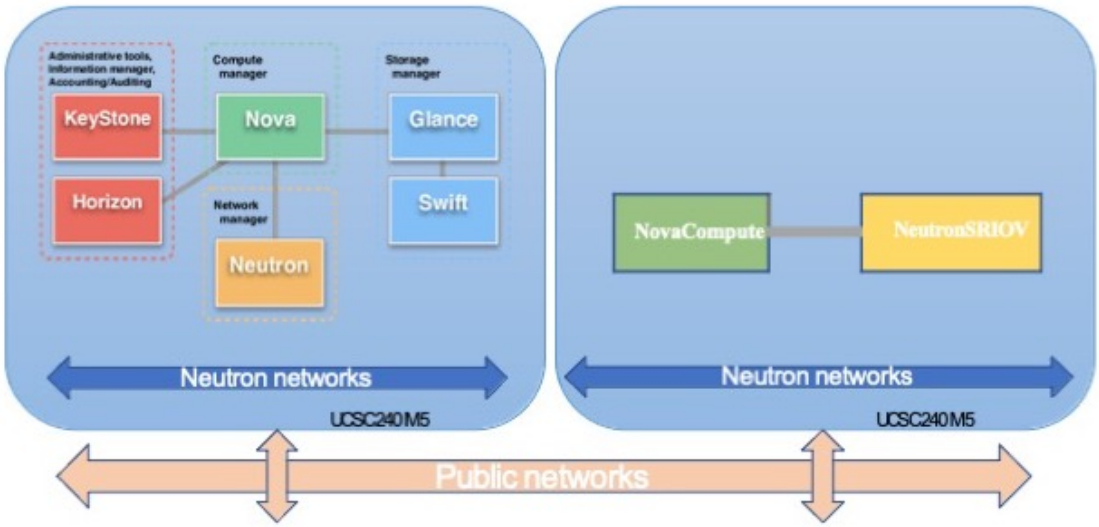
카테고리	지원되는 버전	참고
서버	UCS C240 M5	os-controller 및 os-compute 노드에 대해 각각 하나씩, 2개의 UCS 서버가 권장됩니다.
드라이버	VIRTIO, IXGBE, I40E	다음은 지원되는 드라이버입니다.
운영 체제	Ubuntu Server 18.04	이는 UCS 서버의 권장 OS입니다.
OpenStack 버전	Stein 릴리스	다양한 OpenStack 릴리스에 대한 세부 정보는 다음에서 확인할 수 있습니다. https://releases.openstack.org/

표 *27: Cisco VIM Managed OpenStack*의 하드웨어 및 소프트웨어 요구 사항

카테고리	지원되는 버전	참고
서버 하드웨어	UCS C220-M5/UCS C240-M4	5개의 UCS 서버를 사용하는 것이 좋습니다. os-controller에는 각각 3개, os-compute 노드에는 2개 이상입니다.
동인	VIRTIO, SRIOV 및 VPP	다음은 지원되는 드라이버입니다.
Cisco VIM 버전	Cisco VIM 3.4.4 지원되는 모델: • 운영 체제 - Red Hat Enterprise Linux 7.6 • OpenStack 버전 - OpenStack 13.0(Queens 릴리스)	자세한 내용은 Cisco Virtualized Infrastructure Manager 문서의 3.4.3~3.4.5를 참고하십시오. 다양한 OpenStack 릴리스에 대한 세부 정보는 https://releases.openstack.org/에서 확인할 수 있습니다.
	Cisco VIM 4.2.1 지원되는 모델: • 운영 체제 - Red Hat Enterprise Linux 8.2 • OpenStack 버전 - OpenStack 16.1(Train 릴리스)	자세한 내용은 Cisco Virtualized Infrastructure Manager 문서의 4.2.1을 참고하십시오. 다양한 OpenStack 릴리스에 대한 세부 정보는 https://releases.openstack.org/에서 확인할 수 있습니다.

그림 *63: OpenStack* 플랫폼 토폴로지

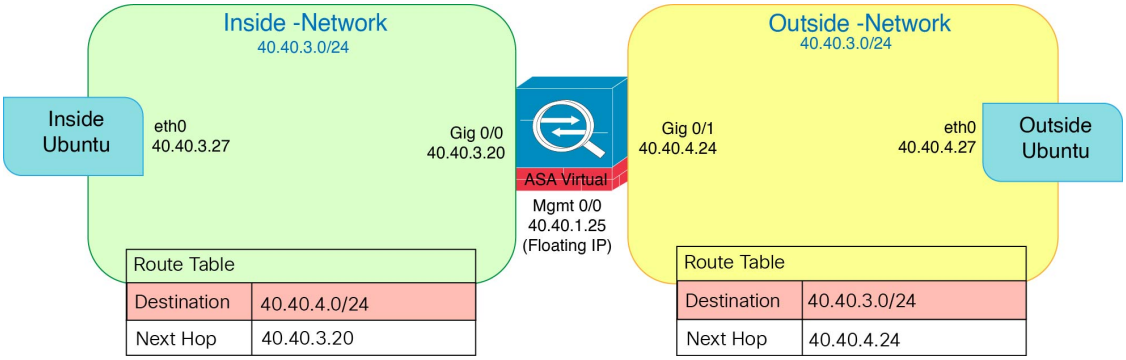
OpenStack 플랫폼 토폴로지는 UCS 서버 2개에 대한 일반 OpenStack 설정을 보여줍니다.



OpenStack 기반 ASA 가상의 샘플 네트워크 토폴로지

다음 그림은 Routed Firewall Mode의 ASA 가상에 대한 권장 네트워크 토폴로지와 ASA 가상에 대해 OpenStack에 구성된 3개의 서브넷(관리, 내부 및 외부)을 보여줍니다.

그림 ***64: OpenStack*** 구축에 대한 ***ASA*** 가상 샘플



OpenStack에 ASA 가상 구축

Cisco는 ASA 가상 구축을 위한 샘플 히트 템플릿을 제공합니다. OpenStack 인프라 리소스를 생성하는 단계는 히트 템플릿(`deploy_os_infra.yaml`) 파일에 통합되어 네트워크, 서브넷 및 라우터 인터페이스를 생성합니다. ASA 가상 구축 단계는 개략적으로 다음 섹션으로 분류됩니다.

- OpenStack Glance 서비스에 ASA 가상 qcow2 이미지를 업로드합니다.
- 네트워크 인프라를 만듭니다.
  - 네트워크

- 서브넷
- 라우터 인터페이스

- ASA 가상 인스턴스를 만듭니다.
  - Flavor
  - 보안 그룹
  - 부동 IP
  - Instance

다음 단계를 사용하여 OpenStack에서 ASA 가상를 구축할 수 있습니다.

OpenStack에 ASA 가상 이미지 업로드

qcow2 이미지(`asav-<version>.qcow2`)를 OpenStack 컨트롤러 노드에 복사한 다음 OpenStack Glance 서비스에 이미지를 업로드합니다.

시작하기 전에

Cisco.com에서 ASA 가상 qcow2 파일을 다운로드하고 이를 Linux 호스트에 넣습니다.

http://www.cisco.com/go/asa-software

참고 Cisco.com 로그인 및 Cisco 서비스 계약이 필요합니다.

단계 **1** qcow2 이미지 파일을 OpenStack 컨트롤러 노드에 복사합니다.

단계 **2** OpenStack Glance 서비스에 ASA 가상 이미지를 업로드합니다.

```
root@ucs-os-controller:$ openstack image create <image_name> --public --disk-
format qcow2 --container-format bare --file ./<asav_qcow2_file>
```

단계 **3** ASA 가상 이미지 업로드에 성공했는지 확인합니다.

root@ucs-os-controller:$ openstack image list

예제:

```
root@ucs-os-controller:$ openstack image
list+--------------------------------------+------------------+--------+
| ID                                   | Name               | Status
|+--------------------------------------+------------------+--------+
| 06dd7975-0b6e-45b8-810a-4ff98546a39d | asav-<version>-image | active
|+--------------------------------------+------------------+--------+
```

업로드된 이미지 및 이미지 상태가 표시됩니다.

다음에 수행할 작업

deploy_os_infra.yaml 템플릿을 사용하여 네트워크 인프라를 생성합니다.

OpenStack 및 ASA 가상의 네트워크 인프라 생성

시작하기 전에

Heat 템플릿 파일은 네트워크 인프라 및 ASA 가상에 대한 필수 구성 요소(예: 버전, 네트워크, 서브넷, 라우터 인터페이스 및 보안 그룹 규칙)를 생성하는 데 필요합니다.

- deploy_os_infra.yaml
- env.yaml

ASA 가상 버전의 템플릿은 GitHub 리포지토리에서 제공됩니다.

- https://github.com/CiscoDevNet/cisco-asav

중요 Cisco에서 제공하는 템플릿은 오픈 소스 예시로 제공되며 일반적인 Cisco TAC 지원 범위에서는 다루지 않습니다. GitHub에서 정기적으로 업데이트 및 ReadMe 지침을 확인하십시오.

단계 1 인프라 히트 템플릿 파일을 구축합니다.

root@ucs-os-controller:$ openstack stack create<stack-name> -e<environment files name> -t<deployment file name>

예제:

```
root@ucs-os-controller:$ openstack stack create infra-stack -e env.yaml -t deploy_os_infra.yaml
```

단계 2 인프라 스택이 성공적으로 생성되었는지 확인합니다.

root@ucs-os-controller:$ openstack stack list

다음에 수행할 작업

OpenStack에서 ASA 가상 인스턴스를 생성합니다.

OpenStack에서 ASA 가상 인스턴스 생성

샘플 ASA 가상 히트 템플릿을 사용하여 OpenStack에 ASA 가상을 구축합니다.

시작하기 전에

OpenStack에 ASA 가상를 구축하려면 히트 템플릿이 필요합니다.

- deploy_asav.yaml

ASA 가상 버전의 템플릿은 GitHub 리포지토리에서 제공됩니다.

- https://github.com/CiscoDevNet/cisco-asav

중요 Cisco에서 제공하는 템플릿은 오픈 소스 예시로 제공되며 일반적인 Cisco TAC 지원 범위에서는 다루지 않습니다. GitHub에서 정기적으로 업데이트 및 ReadMe 지침을 확인하십시오.

단계 **1** ASA 가상 히트 템플릿 파일(`deploy_asav.yaml`)을 구축하여 ASA 가상 인스턴스를 생성합니다.

root@ucs-os-controller:$ openstack stack create asav-stack -e env.yaml-t deploy_asav.yaml

예제:

```
+---------------------+--------------------------------------+
| Field               | Value                                |
+---------------------+--------------------------------------+
| id                  | 14624af1-e5fa-4096-bd86-c453bc2928ae |
| stack_name          | asav-stack                           |
| description         | ASAvtemplate                         |
| creation_time       | 2020-12-07T14:55:05Z                 |
| updated_time        | None                                 |
| stack_status        | CREATE_IN_PROGRESS                   |
| stack_status_reason | Stack CREATE started                 |
+---------------------+--------------------------------------+
```

단계 **2** ASA 가상 스택이 성공적으로 생성되었는지 확인합니다.

root@ucs-os-controller:$ openstack stack list

예제:

```
+--------------------------------------+-----------+----------------------------------+--------------+----------------------+-----+
| ID                                   | Stack Name  | Project                          | Stack Status
   | Creation Time     | Updated Time |
+--------------------------------------+-----------+----------------------------------+--------------+----------------------+-----+
| 14624af1-e5fa-4096-bd86-c453bc2928ae | asav-stack  | 13206e49b48740fdafca83796c6f4ad5 | CREATE_COMPLETE
 | 2020-12-07T14:55:05Z | None         |
| 198336cb-1186-45ab-858f-15ccd3b909c8 | infra-stack | 13206e49b48740fdafca83796c6f4ad5 | CREATE_COMPLETE
 | 2020-12-03T10:46:50Z | None         |
+--------------------------------------+-----------+----------------------------------+--------------+----------------------+-----+
```

15 장

Cisco HyperFlex에 ASAv 구축

HyperFlex 시스템은 위치에 상관없이 모든 애플리케이션에 대한 하이퍼컨버전스를 제공합니다. Cisco Intersight 클라우드 운영 플랫폼을 통해 관리되는 Cisco UCS(Unified Computing System) 기술을 이용하는 HyperFlex는 모든 장소에서 애플리케이션과 데이터를 가동하고, 코어 데이터 센터에서 엣지 및 퍼블릭 클라우드로의 운영을 최적하며, 결과적으로 가속 DevOps 방식을 통해 민첩성을 개선할 수 있습니다.

이 장에서는 기능 지원, 시스템 요구 사항, 지침, 제한 사항 같은, Cisco HyperFlex 환경 내에서의 Firepower Threat Defense Virtual 기능에 대해 설명합니다.

중요 ASAv의 최소 메모리 요구 사항은 2GB입니다. 현재 ASAv가 2GB 미만의 메모리로 실행되는 경우에는 ASAv VM의 메모리를 늘리지 않고는 이전 버전에서 9.13(1) 이상으로 업그레이드할 수 없습니다. 최신 버전의 새 ASAv VM을 재구축할 수도 있습니다.

Cisco HyperFlex의 ASAv 지침 및 제한 사항

VMware vCenter 서버에서 ASAv Cisco HyperFlex의 여러 인스턴스를 생성하고 구축할 수 있습니다. ASAv 구축에 사용되는 특정 하드웨어는 구축된 인스턴스 수 및 사용 요구 사항에 따라 달라질 수 있습니다. 생성하는 각 가상 어플라이언스는 호스트 머신에서 최소 리소스 할당(메모리, CPU 수 및 디스크 공간)을 필요로 합니다.

중요 ASAv는 8GB 디스크 스토리지 크기로 구축됩니다. 디스크 공간의 리소스 할당은 변경할 수 없습니다.

ASAv를 구축하기 전에 다음 지침 및 제한 사항을 검토하십시오.

권장 vNIC

최적의 성능을 누리려면 vmxnet3 vNIC를 사용하는 것이 좋습니다. 이 vNIC는 10Gbps 작업을 지원하지만 CPU 사이클이 필요한 반가상화 네트워크 드라이버입니다. 또한 vmxnet3를 사용할 때는 LRO(Large Receive Offload)를 비활성화하여 TCP 성능 저하를 방지해야 합니다.

OVF 파일 지침

- asav-vi.ovf - vCenter에서 구축하는 경우
- ASAv OVF 구축은 현지화(영어 이외의 언어 모드로 구성 요소 설치)를 지원하지 않습니다. 사용자 환경의 VMware vCenter와 LDAP 서버가 ASCII 호환 모드로 설치되어 있는지 확인해 주십시오.
- ASAv를 설치하고 VM 콘솔을 사용하려면 먼저 키보드를 영어(미국)로 설정해야 합니다.

고가용성을 위한 장애 조치 지침

페일오버 구축의 경우, 대기 유닛에 동일한 라이선스 자격이 있는지 확인합니다. 예를 들어 두 유닛 모두 2Gbps 엔타이틀먼트가 있어야 합니다.

중요 ASAv를 사용하여 고가용성 쌍을 생성할 때는 각 ASAv에 동일한 순서로 데이터 인터페이스를 추가해야 합니다. 각 ASAv에 동일한 인터페이스를 추가했지만 순서가 다르면 ASAv 콘솔에서 오류가 표시될 수 있습니다. 페일오버 기능도 영향을 받을 수 있습니다.

IPv6 지침

VMware vSphere Web Client를 사용하여 ASAv OVF 파일을 처음 구축할 때는 관리 인터페이스의 IPv6 주소를 지정할 수 없습니다. 나중에 ASDM 또는 CLI를 사용하여 IPv6 주소를 추가할 수 있습니다.

vMotion 지침

- VMware에서는 vMotion을 사용하는 경우 공유 스토리지만 사용해야 합니다. 호스트 클러스터가 있는 경우 ASAv를 구축하는 동안 특정 호스트에 로컬로 스토리지를 프로비저닝하거나 공유 호스트에 스토리지를 프로비저닝할 수 있습니다. 그러나 ASAv에서 다른 호스트에 대한 vMotion을 실행하려고 하는 경우, 로컬 스토리지를 사용하면 오류가 발생합니다.

처리량 및 라이선싱을 위한 메모리 및 vCPU 할당

- ASAv에 할당된 메모리의 크기는 처리량 수준에 따라 지정됩니다. 다른 처리량 수준에 대한 라이선스를 요청할 때가 아니라면, **Edit Settings(**설정 수정**)** 대화 상자에서 메모리 설정이나 vCPU 하드웨어 설정을 변경하지 마십시오. 프로비저닝 부족은 성능에 영향을 줄 수 있습니다.

참고 메모리 또는 vCPU 하드웨어 설정을 변경해야 하는 경우 ASA 가상에 대한 라이선싱, 1 페이지에 나와 있는 값만 사용해야 합니다. VMware 권장 메모리 컨피그레이션 최소값, 기본값, 최대값을 사용하지 마십시오.

CPU 예약

- 기본적으로 ASAv를 위해 예약된 CPU는 1000MHz입니다. 공유, 예약 및 제한 설정을 사용하여 ASAv에 할당된 CPU 리소스의 양을 변경할 수 있습니다. **Edit Settings(**설정 편집**)** > **Resources(**리소스**)** > **CPU**. ASAv가 낮은 설정을 사용하는 필수 트래픽 로드 이하인 동안 필요한 목적을 수행할 수 있는 경우, 1000MHz에서 CPU 예약 설정을 낮게 설정하는 작업을 수행할 수 있습니다. ASAv에서 사용된 CPU 양은 실행 중인 하드웨어 플랫폼과 수행 중인 작업의 유형 및 양에 따라 달라집니다.

  CPU Usage(CPU 사용량(MHz)) 차트에서 모든 가상 머신에 대한 호스트의 CPU 사용량 관점을 확인할 수 있습니다. 이 차트는 가상 머신의 Performance(성능) 탭의 Home(홈) 보기에 있습니다. ASAv가 일반적인 트래픽 볼륨을 처리 중인 경우 CPU 사용량에 대한 벤치마크를 설정하면 CPU 예약을 조정할 때 해당 정보를 입력으로 사용할 수 있습니다.

  자세한 내용은 CPU 성능 개선 사항 조언 링크를 참조하십시오.
- ASAv**show vm** > **show cpu**

  명령 또는 ASDM

  Home(홈**)** > **Device Dashboard(**디바이스 대시보드**)** > **Device Information(**디바이스 정보**)** > **Virtual Resources(**가상 리소스**)**

  탭 또는

  Monitoring(모니터링**)** > **Properties(**속성**)** > **System Resources Graphs(**시스템 리소스 그래픽**)** > **CPU** 창을 사용하여 리소스 할당과 초과 또는 부족 프로비저닝 상태인 리소스를 확인할 수 있습니다.

투명 모드의 **UCS B** 및 **C** 시리즈 하드웨어 지침

Cisco UCS B(컴퓨팅 노드) 및 C(컨버지드 노드) 시리즈 하드웨어에서 투명 모드로 실행되는 일부 ASAv 컨피그레이션에서 MAC 플랩이 관찰되었습니다. MAC 주소가 다른 위치에서 표시되면 패킷이 손실됩니다.

다음 지침은 VMware 환경에서 투명 모드로 ASAv를 구축할 때 MAC 플랩을 방지하는 데 도움이 됩니다.

- VMware NIC 팀 구성 - UCS B 또는 C 시리즈에서 투명 모드로 ASAv를 구축하는 경우, 내부 및 외부 인터페이스에 사용하는 포트 그룹에는 활성 업링크 1개만 있어야 하며, 이 업링크는 동일해야 합니다. vCenter에서 VMware NIC 팀 구성.

- ARP 검사 - ASAv에서 ARP 검사를 활성화하고, ARP 검사를 수신해야 하는 인터페이스에서 MAC 및 ARP 항목을 정적으로 구성합니다. ARP 검사 및 검사를 활성화하는 방법은 Cisco ASA 시리즈 일반 운영 구성 가이드를 참조하십시오.

ASAv 및 HyperFlex의 시스템 요구 사항

HyperFlex HX-Series용 구성 및 클러스터

컨피그레이션	클러스터
HX220c 통합 노드	• 플래시 클러스터 • 최소 3개의 노드 클러스터(데이터베이스, VDI, VSI)
HX240c 통합 노드	• 플래시 클러스터 • 최소 3개의 노드 클러스터(VSI: IT/Biz Apps, Test/Dev)
HX220C 및 Edge(VDI, VSI, ROBO) HX240C(VDI, VSI, 테스트/개발)	• 하이브리드 클러스터 • 최소 3개의 노드 클러스터
B200 + C240/C220	컴퓨팅 바운드 애플리케이션/VDI

HyperFlex HX 시리즈용 구축 옵션:

- 하이브리드 클러스터
- 플래시 클러스터
- HyperFlex HX Edge
- SED 드라이브
- NVME 캐시
- GPU

HyperFlex HX 클라우드 기반 관리 옵션은 Cisco HyperFlex 시스템 설치 가이드의 *HyperFlex Fabric Interconnect* 연결 클러스터 구축 섹션을 참조하십시오.

HyperFlex 구성 요소 및 버전

구성 요소	**Version**(버전)
VMware vSphere	7.0.2-18426014
HyperFlex Data Platform	4.5.2a-39429

지원 기능

- 구축 모드 - 라우팅(독립형), 라우팅(HA) 및 투명
- ASAv 기본 HA
- 점보 프레임
- VirtIO
- HyperFlex 데이터 센터 클러스터(확장된 클러스터 제외)
- HyperFlex Edge 클러스터
- HyperFlex All NVMe, All Flash 및 Hybrid 컨버지드 노드
- HyperFlex 컴퓨팅 전용 노드

지원되지 않는 기능

SR-IOV로 실행되는 ASAv는 HyperFlex에서 검증되지 않았습니다.

참고 HyperFlex는 SR-IOV를 지원하지만, MLOM VIC 외에 PCI-e NIC도 필요합니다.

Cisco HyperFlex에 ASAv를 구축하는 방법

단계	작업	추가 정보
1	지침 및 제한 사항을 검토합니다.	Cisco HyperFlex의 ASAv 지침 및 제한 사항, 271 페이지
2	사전 요구 사항을 검토합니다.	ASAv 및 Cisco HyperFlex 사전 요구 사항, 275 페이지
3	cisco.com에서 OVF 파일을 다운로드합니다.	ASAv 소프트웨어 다운로드 및 압축 풀기, 276 페이지
4	Cisco HyperFlex에 ASAv를 구축합니다.	Cisco HyperFlex의 ASAv를 vSphere vCenter에 구축, 277 페이지
5	ASAv 콘솔에 액세스합니다.	ASAv 콘솔 액세스, 279 페이지

ASAv 및 Cisco HyperFlex 사전 요구 사항

VMware vSphere Web Client, vSphere 독립형 클라이언트 또는 OVF 툴을 사용하여 Cisco HyperFlex에서 ASAv를 구축할 수 있습니다. 시스템 요구 사항은 Cisco ASA 호환성을 참조하십시오.

vSphere 표준 스위치에 대한 보안 정책

vSphere 스위치의 경우 계층 2 보안 정책을 수정하고 ASAv 인터페이스에서 사용하는 포트 그룹에 대한 보안 정책 예외를 적용할 수 있습니다. 다음 기본 설정을 확인하십시오.

- Promiscuous Mode(무차별 모드): **Reject(**거부**)**
- MAC Address Changes(MAC 주소 변경): **Accept(**허용**)**
- Forged Transmits(위조된 전송): **Accept(**허용**)**

다음 ASAv 컨피그레이션에 대해 이러한 설정을 수정해야 할 수도 있습니다. 자세한 내용은 vSphere 설명서를 참조하십시오.

표 ***28:*** 포트 그룹 보안 정책 예외

보안 예외	라우팅 방화벽 모드		투명 방화벽 모드	
	장애 조치 없음	장애 조치	장애 조치 없음	장애 조치
무차별 모드	<any>	<any>	수락	수락
MAC 주소 변경	<any>	수락	<any>	수락
위조된 전송	<any>	수락	수락	수락

ASAv 소프트웨어 다운로드 및 압축 풀기

시작하기 전에

ASAv를 구축하기 전에 vSphere에서 하나 이상의 네트워크(관리용)를 구성해야 합니다.

단계 **1** Cisco.com에서 ZIP 파일을 다운로드하고 로컬 디스크에 저장합니다.

https://www.cisco.com/go/asa-software

참고 Cisco.com 로그인 및 Cisco 서비스 계약이 필요합니다.

단계 **2** 작업 디렉터리에 파일의 압축을 풉니다. 이 디렉터리의 어떤 파일도 삭제하지 마십시오. 다음 파일이 포함됩니다.

- asav-vi.ovf—vCenter 구축용
- boot.vmdk—부팅 디스크 이미지
- disk0.vmdk—ASAv 디스크 이미지
- day0.iso—day0-config 파일과 선택적으로 idtoken 파일을 포함하는 ISO
- asav-vi.mf—vCenter 구축용 매니페스트 파일

Cisco HyperFlex의 ASAv를 vSphere vCenter에 구축

이 절차를 사용하여 HyperFlex의 ASAv를 VMware vSphere vCenter에 구축합니다. VMware 웹 클라이언트 (또는 vSphere 클라이언트)를 사용하여 가상 머신을 구축하고 구성할 수 있습니다.

시작하기 전에

HyperFlex에 ASAv를 구축하기 전에 vSphere에서 하나 이상의 네트워크(관리용)를 구성해야 합니다.

HyperFlex 클러스터에 ASAv를 설치하려면 먼저 HyperFlex 클러스터 및 공유 데이터 저장소를 만들어야 합니다. 자세한 내용은 HyperFlex 구성 가이드를 참조하십시오.

단계 **1** vSphere 웹 클라이언트에 로그인합니다.

단계 **2** vSphere 웹 클라이언트를 사용하여, **ACTIONS(**작업**)** > **Deploy OVF Template(OVF** 템플릿 구축**)**을 클릭하여 이전에 다운로드한 OVF 템플릿 파일을 구축합니다.

Deploy OVF Template(OVF 템플릿 구축) 마법사가 나타납니다.

단계 **3** 파일 시스템에서 OVF 템플릿 소스 위치를 찾은 후 **NEXT(**다음**)**를 클릭합니다.

단계 **4** **OVF** 템플릿 세부 사항 페이지를 검토하고 OVF 템플릿 정보(제품 이름, 버전, 벤더, 다운로드 크기, 디스크 크기 및 설명)를 확인한 뒤 **NEXT(**다음**)**를 클릭합니다.

단계 **5** **EULA(End User License Agreement)** 페이지가 나타납니다. OVF 템플릿(VI 템플릿만 해당)과 함께 패키지된 라이선스 계약을 검토하고 라이선스 약관에 동의하면 **Accept(**수락**)**를 클릭하고 **NEXT(**다음**)**를 클릭합니다.

단계 **6** **Name and Location(**이름 및 위치**)** 페이지에서 이 구축의 이름을 입력하고, 인벤토리(공유 데이터스토어 또는 클러스터)에서 HyperFlex를 구축하려는 위치를 선택한 뒤 **NEXT(**다음**)**를 클릭합니다. 이름은 인벤토리 폴더 내에서 고유해야 하며, 최대 길이는 80자입니다.

vSphere 웹 클라이언트는 인벤토리 보기에서 관리되는 개체의 조직 계층 구조를 보여줍니다. 인벤토리는 vCenter 서버 또는 호스트에서 관리되는 개체를 구성하는 데 사용되는 계층 구조입니다. 이 계층 구조에는 vCenter 서버에서 모니터링되는 모든 개체가 포함됩니다.

단계 **7** ASAv HyperFlex를 실행할 리소스 풀로 이동하여 선택하고 **NEXT(**다음**)**를 클릭합니다.

참고 이 페이지는 클러스터에 리소스 풀이 포함되어 있는 경우에만 나타납니다. 컴퓨팅 리소스 풀의 경우 최상의 성능을 위해 클러스터를 사용하는 것이 좋습니다.

단계 **8** 구축 설정을 선택합니다. **Configuration(**구성**)** 드롭다운 목록에서 3개의 지원되는 vCPU/메모리 값 중 하나를 선택하고 **NEXT(**다음**)**를 클릭합니다.

단계 **9** 가상 머신 파일을 저장할 **Storage(**스토리지**)** 위치를 선택하고 **NEXT(**다음**)**를 클릭합니다.

이 페이지에서 대상 클러스터에 이미 구성된 데이터스토어를 선택합니다(데이터스토어는 HX 연결을 통해 생성된 HX 클러스터 공유 데이터스토어임). 가상 머신 컨피그레이션 파일 및 가상 디스크 파일은 해당 데이터 저장소에 저장되어 있습니다. 가상 머신과 모든 가상 디스크 파일을 수용할 만큼 큰 데이터 저장소를 선택합니다.

단계 **10** **Network Mapping(**네트워크 매핑**)** 페이지에서 OVF 템플릿에 지정된 네트워크를 인벤토리의 네트워크에 매핑한 후 **NEXT(**다음**)**를 선택합니다.

Management0-0 인터페이스가 인터넷에서 연결 가능한 VM 네트워크에 연결되었는지 확인하십시오. 비관리 인터페이스는 관리 모드에 따라 ASAv Mangement Centre 또는 ASAv Device Manager에서 구성할 수 있습니다.

중요 이제 HyperFlex의 ASAv는 가상 디바이스를 생성할 때 vmxnet3 인터페이스를 기본값으로 사용합니다. 이전에는 기본값이 e1000이었습니다. e1000 인터페이스를 사용한다면 전환을 강력하게 권장합니다. vmxnet3 디바이스 드라이버 및 네트워크 처리는 HyperFlex와 통합되므로 리소스가 더 적게 사용되고 네트워크 성능이 향상됩니다.

네트워크는 사전 순이 아닐 수도 있습니다. 네트워크를 찾기 어려운 경우 나중에 **Edit Settings(**설정 수정**)** 대화 상자에서 네트워크를 변경할 수 있습니다. 구축 후 인스턴스를 마우스 오른쪽 버튼으로 클릭하고 **Edit Settings(**설정 수정**)**을 선택합니다. 그러나 ID는 네트워크 매핑 페이지에 표시되지 않습니다(네트워크 어댑터 ID만 표시됨).

다음은 인터페이스용 네트워크 어댑터, 소스 네트워크 및 대상 네트워크 색인에 대한 설명입니다(vmxnet3 인터페이스의 기본값입니다).

표 ***29:*** 소스 **-** 대상 네트워크 매핑 **-** ***VMXNET3***

네트워크 어댑터 **ID**	**ASAv** 인터페이스 **ID**
네트워크 어댑터 1	Management 0/0
네트워크 어댑터 2	GigabitEthernet 0/0
네트워크 어댑터 3	GigabitEthernet 0/1
네트워크 어댑터 4	GigabitEthernet 0/2
네트워크 어댑터 5	GigabitEthernet 0/3
네트워크 어댑터 6	GigabitEthernet 0/4
네트워크 어댑터 7	GigabitEthernet 0/5
네트워크 어댑터 8	GigabitEthernet 0/6
네트워크 어댑터 9	GigabitEthernet 0/7
네트워크 어댑터 10	GigabitEthernet 0/8

ASAv릉 구축할 때 총 10개의 인터페이스를 사용할 수 있습니다. 데이터 인터페이스의 경우 소스 네트워크가 올바른 대상 네트워크에 매핑되는지, 또한 각 데이터 인터페이스가 고유한 서브넷 또는 VLAN에 매핑되는지 확인합니다. 모든 인터페이스를 사용할 필요는 없습니다. 사용하지 않으려는 인터페이스는 컨피그레이션에서 비활성화 상태로 유지해도 됩니다.

단계 **11** **Properties(**속성**)** 페이지에서 OVF 템플릿(VI 템플릿 전용)과 함께 패키지된 사용자 구성 가능 속성을 설정합니다.

- **Password(**비밀번호**)** - 관리자 액세스용 비밀번호를 설정합니다.
- **Network(**네트워크**)** - FQDN(정규화된 도메인 이름), DNS, 검색 도메인 및 네트워크 프로토콜(IPv4 또는 IPv6)을 포함한 네트워크 정보를 설정합니다.

- **Management Interface(**관리 인터페이스**)** - 관리 컨피그레이션을 설정하고 드롭다운을 클릭하여 **DHCP/Manual(DHCP/**수동**)**을 선택하고 관리 인터페이스에 대한 IP 컨피그레이션을 설정합니다.
- **Firewall Mode(**방화벽 모드**)** - 초기 방화벽 모드를 설정합니다. **Firewall Mode(**방화벽 모드**)**의 드롭다운 화살표를 클릭하고 지원되는 두 가지 모드인 **Routed(**라우팅**)** 또는 **Transparent(**투명**)** 중 하나를 선택합니다.

단계 **12** **NEXT(**다음**)**를 클릭합니다. 준비 완료 섹션에서 표시된 정보를 검토하고 확인합니다. 이 설정으로 구축을 시작하려면 **Finish(**마침**)**를 클릭합니다. 변경하려면 **Back(**뒤로**)**을 클릭하여 이전 대화 상자로 이동합니다.

(선택 사항) **Power on after deployment(**구축 후 전원 켜기**)** 옵션을 선택하여 VM의 전원을 켠 다음 **Finish(**마침**)**를 클릭합니다.

마법사를 완료하고 나면 vSphere 웹 클라이언트에서 가상 머신을 처리합니다. **Recent Tasks(**최근 작업**)** 창의 **Global Information(**전체 정보**)** 영역에서 Initialize OVF deployment(OVF 구축 초기화) 상태를 확인할 수 있습니다.

작업이 완료되면 Deploy OVF Template(OVF 템플릿 구축) 완료 상태가 표시됩니다.

인벤토리의 지정된 데이터 센터 아래에 ASAv 인스턴스가 표시됩니다. 새 VM을 시작하는 데는 최대 30분이 소요될 수 있습니다.

참고 ASAv HyperFlex를 Cisco Licensing Authority에 등록하려면 인터넷에 액세스할 수 있어야 합니다. 구축 후 인터넷 액세스 및 성공적인 라이선스 등록을 위해 추가 컨피그레이션이 필요할 수 있습니다.

ASAv 콘솔 액세스

ASDM을 사용할 때 경우에 따라 문제 해결에 CLI를 사용해야 할 수 있습니다. 기본적으로 내장형 VMware vSphere 콘솔에 액세스할 수 있습니다. 또는 복사 및 붙여넣기를 포함하여 더 나은 기능을 갖춘 네트워크 직렬 콘솔을 구성할 수 있습니다.

- VMware vSphere 콘솔 사용
- 네트워크 직렬 콘솔 포트 구성

VMware vSphere 콘솔 사용

초기 컨피그레이션 또는 문제 해결의 경우 VMware vSphere Web Client를 통해 제공된 가상 콘솔에서 CLI에 액세스합니다. 나중에 텔넷(Telnet) 또는 SSH에 대해 CLI 원격 액세스를 구성할 수 있습니다.

시작하기 전에

vSphere Web Client의 경우 ASA 가상 콘솔에 액세스하는 데 필요한 클라이언트 통합 플러그인을 설치합니다.

단계 **1** VMware vSphere Web Client의 인벤토리에서 ASA 가상 인스턴스를 마우스 오른쪽 버튼으로 클릭하고 **Open Console(**콘솔 열기**)**을 선택합니다. 또는 Summary(요약) 탭에서 **Launch Console(**콘솔 실행**)**을 클릭합니다.

단계 **2** 콘솔을 클릭하고 **Enter** 키를 누릅니다. 참고: 커서를 놓으려면 **Ctrl+Alt** 키를 누릅니다.

ASA 가상가 여전히 시작 중인 경우 부팅 메시지가 나타납니다.

ASA 가상는 처음 시작될 때 OVF 파일을 통해 제공된 매개변수를 읽어 ASA 가상 시스템 컨피그레이션에 추가합니다. 그런 다음 가동 및 실행될 때까지 자동으로 부팅을 다시 시작합니다. 이러한 이중 부팅은 ASA 가상를 처음 구축한 경우에만 발생합니다.

참고 라이센스를 설치할 때까지 예비 연결 테스트를 수행할 수 있도록 처리량이 100Kbps로 제한됩니다. 라이센스는 일반적인 운영에 필요합니다. 또한 라이센스를 설치할 때까지 콘솔에 다음 메시지가 반복적으로 표시됩니다.

```
Warning: ASAv platform license state is Unlicensed.
Install ASAv platform license for full functionality.
```

다음 프롬프트가 표시됩니다.

```
ciscoasa>
```

이 프롬프트는 현재 사용자 EXEC 모드에 있음을 의미합니다. 사용자 EXEC 모드에서는 기본 명령만 사용 가능합니다.

단계 **3** 특권 실행 모드에 액세스합니다.

예제**:**

```
ciscoasa> enable
```

다음 프롬프트가 나타납니다.

```
Password:
```

단계 **4** **Enter** 키를 눌러 계속합니다. 기본적으로 비밀번호는 비어 있습니다. 이전에 enable 비밀번호를 설정한 경우 Enter 키를 누르는 대신 enable을 입력합니다.

프롬프트가 다음과 같이 변경됩니다.

```
ciscoasa#
```

모든 비 컨피그레이션 명령은 특권 EXEC 모드에서 사용할 수 있습니다. 또한 특권 EXEC 모드에서 컨피그레이션 모드를 입력할 수도 있습니다.

특권 모드를 종료하려면 **disable**, **exit** 또는 **quit** 명령을 입력합니다.

단계 **5** 전역 컨피그레이션 모드에 액세스합니다.

```
ciscoasa# configure terminal
```

프롬프트가 다음으로 변경됩니다.

```
ciscoasa(config)#
```

전역 컨피그레이션 모드에서 ASA 가상 컨피그레이션을 시작할 수 있습니다. 전역 컨피그레이션 모드를 종료하려면 **exit**, **quit** 또는 **end** 명령을 입력합니다.

네트워크 직렬 콘솔 포트 구성

더 나은 콘솔 경험을 위해 콘솔에 액세스할 수 있는 네트워크 직렬 포트를 단독으로 구성하거나 vSPC(Virtual Serial Port Concentrator)에 연결하여 구성할 수 있습니다. 각 방법에 대한 자세한 내용은

VMware vSphere 설명서를 참조하십시오. ASA 가상에서 가상 콘솔 대신 직렬 포트로 콘솔 출력을 보내야 합니다. 이 절차에서는 직렬 포트 콘솔을 사용하는 방법에 대해 설명합니다.

단계 **1** VMware vSphere에서 네트워크 직렬 포트를 구성합니다. VMware vSphere 설명서를 참조하십시오.

단계 **2** ASA 가상에서 disk0의 루트 디렉토리에 "use_ttyS0"이라는 파일을 만듭니다. 파일 내용은 없어도 됩니다. 이 위치에 파일이 있기만 하면 됩니다.

disk0:/use_ttyS0

- ASDM에서 **Tools(**도구**)** > **File Management(**파일 관리**)** 대화 상자를 사용하여 이 이름으로 빈 텍스트 파일을 업로드할 수 있습니다.
- vSphere 콘솔에서 파일 시스템에 있는 기존 파일(임의의 파일)을 새 이름으로 복사할 수 있습니다. 예를 들면 다음과 같습니다.

```
ciscoasa(config)# cd coredumpinfo
ciscoasa(config)# copy coredump.cfg disk0:/use_ttyS0
```

단계 **3** ASA 가상를 다시 로드합니다.

- ASDM에서 **Tools(**도구**)** > **System Reload(**시스템 다시 로드**)**를 선택합니다.
- vSphere 콘솔에서 **reload(**다시 로드**)**를 입력합니다.

  ASA 가상에서 vSphere 콘솔로 보내는 것을 중지하고 대신 직렬 콘솔로 보냅니다.

단계 **4** 직렬 포트를 추가할 때 지정한 vSphere 호스트 IP 주소와 포트 번호로 텔넷 전송하거나, vSPC IP 주소 및 포트로 텔넷 전송합니다.

vCPU 또는 처리량 라이센스 업그레이드

ASAv에서는 처리량 라이선스를 사용하며, 이 라이선스는 사용 가능한 vCPU 수에 영향을 미칩니다.

ASAv에 대한 vCPU 수를 늘리거나 줄이려면 새 라이센스를 요청하고 이를 적용한 후 VMware에서 VM 속성을 새 값과 일치하도록 변경하면 됩니다.

참고 지정된 vCPU가 ASAv 가상 CPU 라이센스 또는 처리량 라이센스와 일치해야 합니다. RAM도 vCPU에 맞게 크기가 지정되어야 합니다. 업그레이드하거나 다운그레이드할 때 다음 절차에 따라 라이센스 및 vCPU를 즉시 조정하십시오. 일치하지 않은 항목이 있으면 ASAv가 제대로 작동하지 않습니다.

단계 **1** 새 ASAv 가상 CPU 라이선스 또는 처리량 라이선스를 요청합니다.

단계 **2** 새 라이센스를 적용합니다. 장애 조치 쌍의 경우 두 유닛 모두에 새 라이센스를 적용합니다.

단계 3 페일오버 사용 여부에 따라 다음 중 하나를 수행합니다.

- 페일오버 - vSphere Web Client에서 스탠바이 ASAv의 전원을 끕니다. 예를 들어 ASAv를 클릭한 다음 **Power Off the virtual machine(**가상 머신 전원 끄기**)**을 클릭하거나, ASAv를 마우스 오른쪽 버튼으로 클릭하고 **Shut Down Guest OS(**게스트 **OS** 종료**)**를 선택합니다.
- 페일오버를 사용하지 않음 - vSphere Web Client에서 ASAv의 전원을 끕니다. 예를 들어 ASAv를 클릭한 다음 **Power Off the virtual machine(**가상 머신 전원 끄기**)**을 클릭하거나, ASAv를 마우스 오른쪽 버튼으로 클릭하고 **Shut Down Guest OS(**게스트 **OS** 종료**)**를 선택합니다.

단계 4 ASAv를 클릭한 다음 **Edit Virtual machine settings(**가상 머신 설정 수정**)**를 클릭합니다(또는 ASAv를 마우스 오른쪽 버튼으로 클릭하고 **Edit Settings(**설정 수정**)**를 선택합니다.

Edit Settings(설정 수정**)** 대화 상자가 나타납니다.

단계 5 ASA 가상에 대한 라이선싱, 1 페이지의 CPU 및 메모리 요구 사항을 참조하여 새 vCPU 라이센스에 대한 올바른 값을 확인합니다.

단계 6 **Virtual Hardware(**가상 하드웨어**)** 탭의 **CPU** 드롭다운 목록에서 새 값을 선택합니다.

단계 7 **Memory(**메모리**)**에 RAM에 대한 새 값을 입력합니다.

단계 8 **OK(**확인**)**를 클릭합니다.

단계 9 ASAv의 전원을 켭니다. 예를 들어 **Power On the Virtual Machine(**가상 머신 전원 켜기**)**을 클릭합니다.

단계 10 장애 조치 쌍의 경우 다음을 수행합니다.

1. 활성 유닛에 대한 콘솔을 열거나 활성 유닛에서 ASDM을 실행합니다.
2. 대기 유닛의 시작이 완료되면 대기 유닛에 장애 조치를 수행합니다.
   - ASDM: **Monitoring(**모니터링**)** > **Properties(**속성**)** > **Failover(**장애 조치**)** > **Status(**상태**)**를 선택하고 **Make Standby(**대기로 전환**)**을 클릭합니다.
   - CLI: **failover active**
3. 활성 유닛에 대해 3~9단계를 반복합니다.

다음에 수행할 작업

자세한 내용은 ASA 가상에 대한 라이선싱, 1 페이지를 참조하십시오.

Cisco HyperFlex의 ASAv에 대한 성능 조정

ASAv는 고성능 어플라이언스이지만, 최상의 결과를 얻으려면 Cisco HyperFlex를 조정해야 할 수 있습니다.

다음은 HyperFlex 환경에서 ASAv의 성능을 극대화하기 위한 모범 사례 및 권장 사항입니다.

점보 프레임 활성화

대형 MTU를 사용하면 대형 패킷을 전송할 수 있습니다. 큰 패킷은 네트워크에서 더욱 효율적으로 사용할 수 있습니다. 다음 지침을 참조하십시오.

- 트래픽 경로의 MTU 일치 - 트래픽 경로에서는 모든 ASAv 인터페이스 및 기타 디바이스 인터페이스의 MTU를 동일하게 설정하는 것이 좋습니다. MTU를 일치시키면 패킷 분할 시 디바이스가 중간에 끼어드는 현상을 방지할 수 있습니다.
- 점보 프레임 수용 - MTU를 최대 9198바이트로 설정할 수 있습니다. 최대값은 ASAv의 경우 9000입니다.

이 절차에서는 다음 환경에서 점보 프레임을 활성화하는 방법을 설명합니다.

vSphere 7.0.1에서의 **HyperFlex Cluster** > **VMware vSphere vSwitch**> **Cisco UCS FI(Fabric Interconnects).**

단계 **1** ASAv를 구축한 ASAv 호스트의 MTU 설정을 변경합니다.

1. vSphere 웹 클라이언트를 사용하여 vCenter 서버에 연결합니다.
2. HyperFlex 호스트의 **Advanced System Settings(**고급 시스템 설정**)**에서 구성 매개변수 `Net.Vmxnet3NonTsoPacketGtMtuAllowed`의 값을 `1`로 설정합니다.
3. 변경 사항을 저장하고 호스트를 재부팅합니다.

자세한 내용은 https://kb.vmware.com/s/article/1038578를 참고하십시오.

단계 **2** VMware vSphere vSwitch의 MTU 설정을 변경합니다.

1. vSphere 웹 클라이언트를 사용하여 vCenter 서버에 연결합니다.
2. VMware vSphere vSwitch의 속성을 편집하고 **MTU** 값을 `9000`으로 설정합니다.

단계 **3** Cisco UCS FI(Fabric Interconnect)의 MTU 설정을 변경합니다.

1. Cisco UCS 관리 콘솔에 로그인합니다.
2. QoS 시스템 클래스를 편집하려면 **LAN** > **LAN Cloud** > **QoS System Class(QoS** 시스템 클래스**)**를 선택합니다. **General(**일반**)** 탭에서 **MTU** 값을 9216으로 설정합니다.
3. vNIC를 편집하려면 **LAN** > **Policies(**정책**)** > **root(**루트**)** > **Sub-Organizations(**하위 조직**)**를 선택합니다.

   <your-hyperflex-org>**vNIC** 템플릿 <your-vnic>. **General(**일반**)** 탭에서 **MTU** 값을 9000으로 설정합니다.

16 장

Alibaba Cloud에서 ASA Virtual 구축

Cisco Adaptive Security Appliance Virtual은 물리적 Cisco ASA와 동일한 소프트웨어를 실행하여 가상 폼 팩터에서 검증된 보안 기능을 제공합니다. 가상 및 물리적 데이터 센터 워크로드를 보호하기 위해 퍼블릭 Alibaba 클라우드에서 ASA Virtual을 구축하고 구성할 수 있습니다. ASA Virtual은 시간이 지남에 따라 위치를 확장, 축소 또는 옮길 수 있습니다.

중요 9.13(1)부터는 지원되는 모든 ASA 가상 vCPU/메모리 구성에서 모든 ASA 가상 라이센스를 사용할 수 있습니다. ASA Virtual 라이선스가 있으면 ASA Virtual 고객은 다양한 VM 리소스 사용 공간에서 실행할 수 있습니다. 이러한 ASA Virtual 라이선스는 지원되는 Alibaba 인스턴스 유형을 늘리는 역할도 합니다.

Alibaba Cloud에서의 ASA Virtual 구축 정보

ASA Virtual은 다음 Alibaba 인스턴스 유형을 지원합니다.

Alibaba 지원 인스턴스 유형

참고 ASA virtual은 최소 3개의 인터페이스(ENI)와 최대 4개의 인터페이스 가 있어야 인스턴스를 지원할 수 있습니다.

네트워크 요구 사항

- 기본 ASA Virtual을 지원할 수 있도록 최소 1개의 Vswitch(서브넷)가 있는 VPC 1개를 만듭니다.

- Vswitch는 인스턴스가 구축되는 것과 동일한 영역에서 사용할 수 있어야 하며, 그렇지 않다면 생성해야 합니다.

관련 설명서

인스턴스 유형 및 해당 구성에 대한 자세한 내용은 Alibaba Cloud 참조

ASA Virtual 및 Alibaba의 사전 요구 사항

- https://www.alibabacloud.com/에서 계정을 만듭니다.
- Cisco.com에서 ASA Virtual qcow2 파일을 다운로드하고 이를 Linux 호스트에 넣습니다.

  http://www.cisco.com/go/asa-software

참고 Cisco.com 로그인 및 Cisco 서비스 계약이 필요합니다.

- ASA Virtual에 라이선스를 부여합니다. ASA Virtual 라이선스를 등록하기 전에는 저성능 모드에서 실행됩니다. 이 모드에서는 100개의 연결 및 100Kbps의 처리량만 허용됩니다. ASA 가상에 대한 라이선싱, 1 페이지의 내용을 참조하십시오.
- 인터페이스 요건:
  - 관리 인터페이스
  - 내부 및 외부 인터페이스
- 통신 경로:
  - 관리 인터페이스—ASA Virtual을 ASDM에 연결할 때 사용합니다. 통과 트래픽에는 사용할 수 없습니다.
  - 내부 인터페이스(필수)—ASA Virtual을 내부 호스트에 연결하는 데 사용합니다.
  - 외부 인터페이스(필수)—ASA Virtual을 공용 네트워크에 연결하는 데 사용합니다.
- ASA Virtual 시스템 요구 사항은 Cisco ASA 호환성을 참조하십시오.

ASA Virtual 및 Alibaba의 기능 지원 및 제한 사항

지원 기능

Alibaba의 ASA Virtual은 다음 기능을 지원합니다.

- QCOW2 이미지 패키지

- 기본 제품 가져오기
- Day-0 컨피그레이션
- 공개 키 또는 비밀번호를 사용하는 SSH
- 디버깅을 위해 ASA Virtual에 액세스할 수 있는 Alibaba UI 콘솔
- Alibaba UI 중지/다시 시작
- 지원되는 인스턴스 유형: ecs.g5ne.large, ecs.g5ne.xlarge, ecs.g5ne.2xlarge 및 ecs.g5ne.4xlarge
- BYOL 라이선스 지원

지원되지 않는 기능

Alibaba의 ASA Virtual은 버전 7.2에서는 다음을 지원하지 않습니다.

- 고가용성 기능
- 자동 확장
- IPv6
- SR-IOV

제한 사항

- 서브넷 수준 라우팅이 허용되지 않으므로 동일한 VPC에 있는 East-West 트래픽은 Alibaba에서 지원되지 않습니다.
- 투명, 인라인 및 패시브 모드는 현재 지원되지 않습니다.
- ASA Virtual 애플리케이션을 구축하려면 네트워크 고급 인스턴스 사양 제품군인 g5ne를 사용하는 것이 좋습니다.
- Alibaba의 몇 가지 인스턴스 유형에만 제한되므로 점보 프레임은 지원되지 않습니다.

관련 설명서

자세한 내용은 Alibaba Cloud를 참조하십시오.

Alibaba에 ASA Virtual 구축

구축하려는 ASA Virtual의 이미지가 **Image Configuration(**이미지 컨피그레이션**)**에 표시되는지 확인합니다.

단계 **1** https://www.alibabacloud.com/에 로그인하고 지역을 선택합니다.

참고 Alibaba는 여러 지역으로 나뉘며, 각 지역은 상호 격리되어 있습니다. 화면의 우측 상단에 지역이 표시됩니다. 한 지역의 리소스가 다른 지역에는 나타나지 않습니다. 원하는 지역에 있는지 정기적으로 확인합니다.

단계 **2** 맞춤형 가상화 이미지 생성

Alibaba는 QCOW2 이미지만 지원합니다.

a) OSS(Object Storage Service)로 이동하여 버킷을 생성하고 다음을 수행합니다.

버킷 이름은 Alibaba 프로젝트 내에서 전역적으로 고유해야 합니다.

1. 로컬 디렉터리에서 Alibaba 버킷으로 QCOW2 이미지를 업로드합니다.
2. 왼쪽 탐색 창에서 **Buckets(**버킷**)** > **ASA Virtualbucket** > **Upload(**업로드**)**를 클릭합니다.
3. 업로드가 성공적으로 완료되면 **Private(**프라이빗**)**을 ACL로 선택하고 개체 세부 정보에 언급된 OSS 개체 주소를 복사합니다.
4. 버킷에 있는 맞춤형 이미지의 OSS 개체 주소를 붙여넣습니다.
5. **Linux**를 OS로 선택하고 **Others Linux(**기타 **Linux)**를 변형 유형으로 선택합니다.
6. 시스템 아키텍처로 **x86_64**를 선택합니다.
7. Image format(이미지 형식)을 **QCOW2**로 선택합니다.
8. 라이선스 유형을 BYOL로 선택합니다.

b) 이전 단계의 반가상화 이미지에서 인스턴스를 생성합니다.

1. 왼쪽 탐색 창에서 **Images(**이미지**)** > **Custom Image(**맞춤형 이미지**)** > **Actions(**작업**)** > **Create Instance(**인스턴세 생성**)**를 클릭합니다.

단계 **3** 맞춤형 이미지에서 인스턴스 생성

a) **Elastic Compute Service** > **Create Instance(**인스턴스 생성**)**로 이동하고 다음을 선택합니다.

1. **Billing Method(**청구 방법**)**: 종량제
2. **Region(**지역**)**: 요구 사항에 따릅니다.
3. **Instance Type(**인스턴스 유형**)**: ecs.g5ne.large / ecs.g5ne.xlarge /ecs.g5ne.2xlarge /ecs.g5ne.4xlarge
4. **Quantity(**수량**)**: 필요에 따릅니다.
5. **Image(**이미지**)**: 이전 섹션에서 생성한 맞춤형 이미지입니다.
6. **System Disk(**시스템 디스크**)**: 최솟값은 20GB입니다.

b) 계속 진행하려면 다음을 선택합니다.

1. **VPC**: ASA Virtual을 구축할 VPC입니다.
2. **Vswitch**: 기본 인터페이스의 서브넷입니다.

3. **Assign Public IPv4 Address(**퍼블릭 **IPv4** 주소 할당**)**: SSH를 통해 연결하는 데 필요합니다(선택하지 않으면 UI에서의 Alibaba 콘솔 연결을 통해서만 ASA Virtual에 액세스할 수 있습니다).
4. **Security Group(**보안 그룹**)**: 적절한 보안 그룹을 선택합니다.
5. **Interfaces(**인터페이스**)**: 기본 인터페이스는 2단계에서 선택한 서브넷에 속합니다. 인스턴스는 두 개의 인터페이스를 사용하여 구축할 수 있으며, 나머지는 구축 후에 연결할 수 있습니다.

c) 다음 섹션으로 이동하여 다음을 수행합니다.

1. **Key-Pair(**키 쌍**)**: 키 기반 로그인의 경우 (아직 하지 않았다면) 키 쌍을 생성합니다. 비밀번호를 사용하여 인스턴스에 액세스할 수도 있습니다.
2. **Instance-name**: 인스턴스의 적절한 이름입니다.
3. **Day-0(**사용자 데이터**)**: 요구 사항에 맞는 Day0 컨피그레이션을 제공합니다(64 기본 인코딩은 선택하지 마십시오).

   Sample Day 0 Configuration(샘플 **Day 0** 컨피그레이션**)** -

```
! ASA Version 9.x! required config start
interface management0/0
management-only
nameif management
security-level 100
ip address dhcp
no shut
!
crypto key generate rsa modulus 2048 noconfirm
ssh 0 0 management
ssh timeout 60
ssh version 2
username admin nopassword privilege 15
username admin attributes
service-type admin
aaa authentication ssh console LOCAL
! required config end
```

d) 서비스 약관에 동의하고 인스턴스를 생성합니다.

단계 **4** **Launch Instance(**인스턴스 실행**)**를 클릭하여 ASA Virtual을 구축합니다.

Alibaba의 ASAv에 대한 성능 조정

VPN 최적화

Alibaba c5 인스턴스는 이전 c3, c4 및 m4 인스턴스보다 훨씬 뛰어난 성능을 제공합니다. c5 인스턴스 제품군의 대략적인 RA VPN 처리량(AES-CBC 암호화를 통한 450B TCP 트래픽을 사용하는 DTLS)은 다음과 같아야 합니다.

- c5.large에서는 0.5Gbps

- c5.xlarge에서는 1Gbps
- c5.2xlarge에서는 2Gbps
- c5.4xlarge에서는 4Gbps

17 장

구성ASA 가상

ASA 가상 구축에서는 ASDM 액세스를 사전에 구성합니다. 웹 브라우저를 사용하여, 구축 중에 지정한 클라이언트 IP 주소에서 ASA 가상 관리 IP 주소에 연결할 수 있습니다. 이 장에서는 다른 클라이언트에서 ASDM에 액세스하도록 허용하는 방법 및 CLI 액세스(SSH 또는 텔넷)를 허용하는 방법에 대해서도 설명합니다. 이 장에서 다루는 그 밖의 필수 컨피그레이션 작업에는 ASDM에서 마법사를 통해 제공되는 라이센스 설치 및 일반 컨피그레이션 작업이 포함됩니다.

ASDM 시작

단계 **1** ASDM 클라이언트로 지정한 PC에서 다음 URL을 입력합니다.

https://*asa_ip_address*/**admin**

다음 버튼이 있는 ASDM 시작 창이 나타납니다.

- **Install ASDM Launcher and Run ASDM(ASDM Launcher** 설치 및 **ASDM** 실행**)**
- **Run ASDM(ASDM** 실행**)**
- **Run Startup Wizard(**시작 마법사 실행**)**

단계 **2** Launcher를 다운로드하려면

a) **Install ASDM Launcher and Run ASDM(ASDM Launcher** 설치 및 **ASDM** 실행**)**을 클릭합니다.

b) 사용자 이름 및 비밀번호 필드를 비어 있는 상태로 두고(새로 설치하는 경우) **OK(**확인**)**를 클릭합니다. 어떤 HTTPS 인증도 구성되지 않았으므로 사용자 이름 없이, **enable** 비밀번호(기본적으로 비어 있음)를 사용하여 ASDM에 액세스할 수 있습니다. HTTPS 인증을 활성화한 경우 사용자 이름과 관련 비밀번호를 입력합니다.

c) 설치 프로그램을 PC에 저장한 다음 시작합니다. 설치가 완료되면 ASDM-IDM Launcher가 자동으로 열립니다.

d) 관리 IP 주소를 입력한 후 사용자 이름과 비밀번호를 비어 있는 상태로 두고(새로 설치하는 경우) **OK(**확인**)**를 클릭합니다. HTTPS 인증을 활성화한 경우 사용자 이름과 관련 비밀번호를 입력합니다.

단계 3 Java Web Start를 사용하려면

a) **Run ASDM(ASDM 실행)** 또는 **Run Startup Wizard(**시작 마법사 실행**)**를 클릭합니다.
b) 프롬프트에 따라 바로가기를 컴퓨터에 저장합니다. 저장하지 않고 열 수도 있습니다.
c) 바로가기에서 Java Web Start를 시작합니다.
d) 표시되는 대화 상자의 안내에 따라 인증서를 승인합니다. Cisco ASDM-IDM Launcher가 나타납니다.
e) 사용자 이름과 비밀번호를 비어 있는 상태로 두고(새로 설치하는 경우) **OK(**확인**)**를 클릭합니다. HTTPS 인증을 활성화한 경우 사용자 이름과 관련 비밀번호를 입력합니다.

ASDM을 사용하여 초기 컨피그레이션 수행

다음 ASDM 마법사 및 절차를 사용하여 초기 컨피그레이션을 수행할 수 있습니다.

- 시작 마법사 실행
- (선택 사항) ASA 가상 뒤에 있는 공용 서버에 대한 액세스 허용
- (선택 사항) VPN 마법사 실행
- (선택 사항) ASDM에서 다른 마법사 실행

CLI 컨피그레이션에 대해서는 Cisco Secure Firewall ASA Series CLI 컨피그레이션 가이드를 참조하십시오.

시작 마법사 실행

Startup Wizard(시작 마법사**)**를 실행하여 구축에 맞게 보안 정책을 사용자 정의합니다.

단계 1 **Wizards(**마법사**)** > **Startup Wizard(**시작 마법사**)**를 선택합니다.

단계 2 구축에 맞게 보안 정책을 사용자 지정합니다. 다음을 설정할 수 있습니다.

- 호스트 이름
- 도메인 이름
- 관리 비밀번호
- 인터페이스
- IP 주소
- 정적 경로
- DHCP 서버
- NAT(Network Address Translation) 규칙

- 기타 등등...

(선택 사항) ASA 가상 뒤에 있는 공용 서버에 대한 액세스 허용

Configuration(컨피그레이션**)** > **Firewall(**방화벽**)** > **Public Servers(**공용 서버**)** 창에서는 인터넷을 통해 내부 서버에 액세스할 수 있도록 하는 보안 정책을 자동으로 구성합니다. 비즈니스 소유자는 웹 및 FTP 서버 등 외부 사용자가 사용할 수 있도록 해야 하는 내부 네트워크 서비스를 운영할 수 있습니다. 이러한 서비스를 ASA 가상 뒤에 있는 DMZ(Demilitarized Zone)라는 별도의 네트워크에 둘 수 있습니다. 공용 서버를 DMZ에 두면 공용 서버에 대해 실행된 어떤 공격도 내부 네트워크에 영향을 주지 않습니다.

(선택 사항) VPN 마법사 실행

다음 마법사를 사용하여 VPN을 구성할 수 있습니다(**Wizards(**마법사**)** > **VPN Wizards(VPN** 마법사**)**).

- Site-to-Site VPN Wizard(사이트 대 사이트 VPN 마법사) - ASA 가상 및 다른 VPN 지원 디바이스 사이에 IPsec 사이트 대 사이트 터널을 만듭니다.
- AnyConnect VPN Wizard(AnyConnect VPN 마법사) - Cisco AnyConnect VPN 클라이언트를 위한 SSL VPN 원격 액세스를 구성합니다. Secure Client는 기업 리소스에 대한 전체 VPN 터널링을 통해 원격 사용자에게 ASA에 대한 SSL 연결을 제공합니다 원격 사용자가 브라우저를 통해 처음 연결할 때 Secure Client를 다운로드하도록 ASA 정책을 구성할 수 있습니다. Secure Client 3.0 이상을 사용하면 클라이언트에서 SSL 또는 IPsec IKEv2 VPN 프로토콜을 실행할 수 있습니다.
- Clientless SSL VPN Wizard(클라이언트리스 SSL VPN 마법사) - 브라우저에 대한 클라이언트리스 SSL VPN 원격 액세스를 구성합니다. 클라이언트리스 브라우저 기반 SSL VPN을 통해 사용자는 웹 브라우저를 사용하여 ASA에 보안 원격 액세스 VPN 터널을 설정할 수 있습니다. 사용자는 인증 후 포털 페이지에 액세스하여 지원되는 특정 내부 리소스에 액세스할 수 있습니다. 네트워크 관리자는 사용자 그룹별로 리소스에 대한 액세스를 제공합니다. ACL을 적용하여 특정 회사 리소스에 대한 액세스를 제한하거나 허용할 수 있습니다.
- IPsec (IKEv1 or IKEv2) Remote Access VPN Wizard(IPsec(IKEv1 또는 IKEv2) 원격 액세스 VPN 마법사) - Cisco IPsec 클라이언트에 대한 IPsec VPN 원격 액세스를 구성합니다.

Azure에 대한 ASA 가상 IPsec VTI(Virtual Tunnel Interface) 연결을 구성하는 자세한 방법은 Azure에 대한 ASA IPsec VTI 연결 구성을 참고하십시오.

(선택 사항) ASDM에서 다른 마법사 실행

ASDM에서 다른 마법사를 실행하여 고가용성, VPN 클러스터 로드 밸런싱 및 패킷 캡처를 이용해 페일오버를 구성할 수 있습니다.

- High Availability and Scalability Wizard(고가용성 및 확장성 마법사) - 장애 조치 또는 VPN 로드 밸런싱을 구성합니다.

- Packet Capture Wizard(패킷 캡처 마법사) - 패킷 캡처를 구성하고 실행합니다. 이 마법사는 각 인그레스 및 이그레스 인터페이스에서 하나의 패킷 캡처를 실행합니다. 패킷 캡처가 완료되면 패킷 분석기에서 검사하고 재생하기 위해 패킷 캡처를 PC에 저장할 수 있습니다.

고급 컨피그레이션

ASA 가상을 계속 구성하려면 Cisco Secure Firewall ASA Series 설명서 탐색을 참조하십시오.