고급 컨피그레이션

일부 디바이스 기능은 ASA 컨피그레이션 명령을 사용하여 구성됩니다. device manager는 명령 기반 기능을 많이 구성할 수 있지만 이를 모두 지원하지는 않습니다. device manager에서 달리 지원되지 않는 이러한 ASA 기능 중 일부를 사용해야 하는 경우, 스마트 CLI 또는 FlexConfig를 사용하여 기능을 수동으로 구성할 수 있습니다.

다음 주제에서는 이러한 유형의 고급 컨피그레이션에 대해 자세히 설명합니다.

스마트 CLI 및 FlexConfig 정보

Threat Defense ASA 구성 명령을 사용하여 일부 기능(모든 기능이 아님)을 구현합니다. threat defense 구성 명령의 고유 집합은 없습니다.

다음 방법으로 CLI를 사용하여 기능을 구성할 수 있습니다.

- 스마트 **CLI** — (기본 방법) 스마트 CLI 템플릿은 특정 기능에 대해 사전 정의된 템플릿입니다. 이 기능에 필요한 모든 명령은 제공되므로 변수의 값을 선택하기만 하면 됩니다. 시스템에서 선택 항목을 검증해 주기 때문에 기능을 더욱 올바르게 구성할 수 있습니다. 원하는 기능에 해당하는 스마트 CLI 템플릿이 있는 경우, 해당 스마트 CLI를 사용해야 합니다.

- **FlexConfig** — FlexConfig 정책은 FlexConfig 개체의 모음입니다. FlexConfig 개체는 스마트 CLI 템플릿보다 자유 형식으로 이용할 수 있으며, 시스템에서 CLI, 변수 또는 데이터 검증을 수행하지 않습니다. 유효한 명령 시퀀스를 생성하기 위해서는 ASA 컨피그레이션 명령을 알아야 하며 ASA 컨피그레이션 가이드를 준수해야 합니다.

스마트 CLI 및 FlexConfig를 사용하면 device manager 정책 및 설정을 통해 직접 지원되지 않는 기능을 구성할 수 있습니다.

주의 ASA에 대한 강력한 배경 지식을 보유하고 있으며 사용에 대한 전적인 책임을 질 수 있는 고급 사용자인 경우에만 스마트 CLI 및 FlexConfig를 사용하는 것이 좋습니다. 금지되지 않은 모든 명령을 구성할 수 있습니다. 스마트 CLI와 FlexConfig를 통해 기능을 활성화하는 경우, 구성되어 있는 다른 기능과 함께 의도하지 않은 결과를 초래할 수 있습니다.

구성한 스마트 CLI 및 FlexConfig 개체와 관련된 지원을 받기 위해 Cisco TAC(Technical Assistance Center)에 문의할 수 있습니다. Cisco TAC(Technical Assistance Center)에서는 고객을 대신하여 맞춤형 컨피그레이션을 설계하거나 작성하지 않습니다. Cisco에서는 올바른 작동이나 기타 threat defense 기능과의 상호운용성에 대해 어떠한 보증도 명시하지 않습니다. 스마트 CLI 및 FlexConfig 기능은 언제든지 사용이 중지될 수 있습니다. 완벽하게 보장되는 기능을 지원받으려면 device manager의 지원을 기다려야 합니다. 의심스러운 경우에는 스마트 CLI 또는 FlexConfig를 사용하지 마십시오.

다음 주제에서는 이러한 기능에 대해 자세히 설명합니다.

스마트 **CLI** 및 **FlexConfig**에 대한 권장 사용 방법

FlexConfig에는 다음과 같이 권장되는 주요 사용 방법이 두 가지 있습니다.

- ASA에서 threat defense로 마이그레이션하는 중이며 device manager에서 직접 지원하지 않는 호환 가능한 기능을 현재 사용 중이고 계속 사용해야 하는 경우입니다. 이 경우, ASA에서 **show running-config** 명령을 사용하여 해당 기능에 대한 컨피그레이션을 확인하고 FlexConfig 개체를 생성하여 해당 기능을 구현하십시오. 두 디바이스에서 **show running-config** 출력을 비교하여 확인합니다.
- threat defense를 사용 중이지만 구성해야 하는 설정 또는 기능이 있는 경우(예: Cisco TAC(Technical Assistance Center)에서 발생한 특정 문제를 해결하려면 특정 설정이 필요하다고 알려주는 경우), 복잡한 기능에 대해서는 랩 디바이스를 사용하여 FlexConfig를 테스트하고 정상적으로 작동하는지 확인합니다.

ASA 컨피그레이션을 재생성하려면 먼저 표준 정책에서 동일한 기능을 구성할 수 있는지 확인합니다. 예를 들어, 액세스 제어 정책에 침입 탐지 및 방지, HTTP 및 기타 프로토콜 검사 유형, URL 필터링, 애플리케이션 필터링, 액세스 제어(ASA에서는 별도의 기능을 사용하여 구현함)가 포함된 경우, 많은 기능이 CLI 명령을 사용하여 컨피그레이션된 것이 아니므로 **show running-config**의 출력 내에 모든 정책이 표시되지는 않습니다.

참고 ASA와 threat defense는 일대일로 중복되지 않는다는 점을 항상 기억해야 합니다. threat defense 디바이스에서 ASA 컨피그레이션을 완벽하게 재생성하려고 시도하지 마십시오. FlexConfig를 사용하여 구성하는 모든 기능은 신중히 테스트해야 합니다.

스마트 CLI 및 FlexConfig 개체의 CLI 명령

threat defense 는 ASA 구성 명령을 사용하여 일부 기능을 구성합니다. 모든 ASA 기능이 threat defense 에서 호환되는 것은 아니지만, threat defense에서는 작업 가능하나 device manager 정책에서는 구성할 수 없는 기능도 일부 있습니다. 스마트 CLI 및 FlexConfig 개체를 사용하여 이러한 기능을 구성하는 데 필요한 CLI를 지정할 수 있습니다.

스마트 CLI 또는 FlexConfig를 사용하여 기능을 수동으로 구성하려는 경우, 적절한 구문에 따라 명령을 파악하고 구현해야 합니다. FlexConfig는 CLI 명령 구문을 검증하지 않습니다. 적절한 구문 및 CLI 명령 구성에 대한 자세한 내용을 확인하려면 ASA 설명서를 참조하십시오.

- ASA CLI 컨피그레이션 가이드에서는 기능을 구성하는 방법에 대해 설명합니다. 가이드 위치: http://www.cisco.com/c/en/us/support/security/asa-5500-series-next-generation-firewalls/products-installation-and-configuration-guides-list.html
- ASA 명령 참조에서는 명령 이름을 기준으로 정렬된 추가 정보를 제공합니다. 참조 위치: http://www.cisco.com/c/en/us/support/security/asa-5500-series-next-generation-firewalls/products-command-reference-list.html

다음 주제에서는 컨피그레이션 명령에 대해 자세히 설명합니다.

소프트웨어 업그레이드가 FlexConfig 정책에 미치는 영향

threat defense 소프트웨어의 각 새 버전을 설치할 때마다 device manager의 기능 구성을 위한 지원이 추가됩니다. 이러한 새 기능이 이전에 FlexConfig를 사용하여 구성한 기능과 겹치는 경우도 있습니다.

업그레이드 후에는 FlexConfig 정책 및 개체를 검사해야 합니다. device manager 또는 스마트 CLI 내에 추가된 지원으로 인해 금지된 명령을 포함하는 정책이나 개체가 있는 경우, 개체 목록의 아이콘과 메시지에 해당 문제가 표시됩니다. 이러한 경우에는 시간을 할애하여 컨피그레이션을 다시 수행하십시오. 금지된 명령 목록을 사용하면 해당 명령을 구성해야 하는 위치를 확인하는 데 도움이 됩니다.

FlexConfig 정책에 연결된 FlexConfig 개체에 새롭게 금지된 명령이 포함되어 있어도 변경 사항을 구축할 수는 있습니다. 하지만 FlexConfig 정책에 나와 있는 모든 문제를 해결할 때까지 새 스마트 CLI 개체를 생성할 수는 없습니다.

디바이스 컨피그레이션에 능동적으로 구축 중인 개체에만 제한이 적용되므로, 문제가 있는 개체를 FlexConfig 정책에서 제거하기만 하면 됩니다. 따라서 개체를 제거한 다음, 해당하는 스마트 CLI 또는 통합 device manager 컨피그레이션을 생성할 때 참조로 사용할 수 있습니다. 새 컨피그레이션에 만족하는 경우에는 개체만 삭제하면 됩니다. 제거된 개체에 금지되지 않은 일부 요소가 포함된 경우에는 해당 개체를 수정하여 지원되지 않은 명령을 제거한 후 FlexConfig 정책에 개체를 다시 연결할 수 있습니다.

ASA 소프트웨어 버전 및 현재 CLI 컨피그레이션 확인

시스템이 ASA 소프트웨어 명령을 사용하여 일부 기능을 구성하므로 threat defense 디바이스에서 실행 중인 소프트웨어에서 사용되는 현재 ASA 버전을 확인해야 합니다. 이 버전 번호에 따라 기능 구성 시 어떤 ASA CLI 컨피그레이션 가이드를 참조해야 하는지 알 수 있습니다. 또한 현재 CLI 기반 컨피그레이션을 확인하고, 구현하려는 ASA 컨피그레이션과 이를 비교합니다.

모든 ASA 컨피그레이션은 threat defense 컨피그레이션과 매우 다릅니다. threat defense 정책은 CLI 외부에서 구성되는 경우가 많아서 명령을 보고 컨피그레이션을 확인할 수가 없습니다. ASA와 threat defense 컨피그레이션 간에 일대일 대응 관계를 생성하지 마십시오.

이 정보를 확인하려면 device manager에서 CLI 콘솔을 열거나 디바이스 관리 인터페이스에 대한 SSH 연결을 설정하고 다음 명령을 실행합니다.

- **show version system** Cisco Adaptive Security Appliance 소프트웨어 버전 번호를 찾습니다.
- **show running-config** 현재 CLI 컨피그레이션을 확인합니다.
- **show running-config all** 현재 CLI 구성의 모든 기본 명령을 포함합니다.

금지된 **CLI** 명령

스마트 CLI와 FlexConfig의 목적은 device manager를 사용하여 threat defense 디바이스에서는 구성할 수 없으나 ASA 디바이스에서는 사용 가능한 기능을 구성하는 것입니다.

따라서 device manager에서 동일한 역할을 하는 ASA 기능을 구성할 수 없습니다. 다음 표에는 이러한 금지된 명령 영역 중 일부가 나와 있습니다. 이 목록에는 컨피그레이션 모드를 시작하는 상위 명령이 여러 개 포함되어 있습니다. 상위 명령의 금지 사항에는 하위 명령의 금지 사항이 포함됩니다. 여기에는 명령의 **no** 버전과 그와 연관된 **clear** 명령도 포함되어 있습니다.

FlexConfig 개체 편집기를 사용하면 개체에 이러한 명령을 포함할 수 없습니다. 스마트 CLI 템플릿에는 유효하게 구성할 수 있는 명령만 포함되어 있으므로 이 목록이 적용되지 않습니다.

금지된 **CLI** 명령	참고
aaa	**Objects(**개체**)** > **Identity Sources(ID** 소스**)**를 사용합니다.
aaa-server	**Objects(**개체**)** > **Identity Sources(ID** 소스**)**를 사용합니다.
access-group	**Policies(**정책**)** > **Access Control(**액세스 제어**)**을 사용하여 액세스 규칙을 구성합니다.
access-list	부분적으로 차단되었습니다. • **ethertype** 액세스 목록을 생성할 수 있습니다. • **extended** 및 **standard** 액세스 목록은 생성할 수 없습니다. 스마트 CLI 확장 액세스 목록 또는 표준 액세스 목록 개체를 사용하여 이러한 ACL을 생성합니다. 그런 다음, 서비스 정책 트래픽 클래스용 확장 ACL을 사용하는 **match access-list** 등의 개체 이름으로 ACL을 참조하는 FlexConfig 지원 명령에서 이러한 ACL을 사용할 수 있습니다. • 시스템에서 **access-group** 명령과 함께 사용하는 **advanced** 액세스 목록은 생성할 수 없습니다. 대신, **Policies(**정책**)** > **Access Control(**액세스 제어**)**을 사용하여 액세스 규칙을 구성합니다. • **webtype** 액세스 목록은 생성할 수 없습니다.

금지된 **CLI** 명령	참고
anyconnect-custom-data	**Device(**디바이스**)** > **Remote Access VPN(**원격 액세스 **VPN)**을 사용하여 Secure Client를 구성합니다.
asdm	이 기능은 threat defense 시스템에 적용되지 않습니다.
as-path	스마트 CLI AS 경로 개체를 생성한 다음, 스마트 CLI BGP 개체에 이를 사용하여 자동 시스템 경로 필터를 구성할 수 있습니다.
attribute	—
auth-prompt	이 기능은 threat defense 시스템에 적용되지 않습니다.
boot	—
call-home	—
captive-portal	**Policies(**정책**)** > **Identity(ID)**를 사용하여 활성 인증에 사용되는 종속 포털을 구성합니다.
clear	—
client-update	—
clock	**Device(**디바이스**)** > **System Settings(**시스템 설정**)** > **NTP**를 사용하여 시스템 시간을 구성합니다.
cluster	—
command-alias	—
community-list	스마트 CLI 확장 커뮤니티 목록 또는 표준 커뮤니티 목록 개체를 생성한 다음, 스마트 CLI BGP 개체에 이를 사용하여 커뮤니티 목록 필터를 구성할 수 있습니다.
compression	—
configure	—
crypto	**Objects(**개체**)** 페이지에서 **Certificates(**인증서**)**, **IKE Policies(IKE** 정책**)** 및 **IPSec Proposals(IPSec** 제안**)**를 사용합니다.
ddns	**Device(**디바이스**)** > **System Settings(**시스템 설정**)** > **DDNS Service(DDNS** 서비스**)**를 사용하여 동적 DNS를 설정합니다.
dhcp-client	—
dhcpd	**Device(**디바이스**)** > **System Settings(**시스템 설정**)** > **DHCP Server(DHCP** 서버**)**를 사용합니다. 그러나 이 **dhcpd option** 명령은 허용됩니다.

금지된 **CLI** 명령	참고
dhcprelay	대신 위협 방어 API에서 dhcprelayservices 리소스를 사용하십시오.
dns	**Objects(**개체**)** > **DNS Groups(DNS** 그룹**)**를 사용하여 DNS 그룹을 구성한 다음, **Device(**디바이스**)** > **System Settings(**시스템 설정**)** > **DNS Server(DNS** 서버**)**를 사용하여 그룹을 할당합니다.
dns-group	**Objects(**개체**)** > **DNS Groups(DNS** 그룹**)**를 사용하여 DNS 그룹을 구성한 다음, **Device(**디바이스**)** > **System Settings(**시스템 설정**)** > **DNS Server(DNS** 서버**)**를 사용하여 그룹을 할당합니다.
domain-name	**Objects(**개체**)** > **DNS Groups(DNS** 그룹**)**를 사용하여 DNS 그룹을 구성한 다음, **Device(**디바이스**)** > **System Settings(**시스템 설정**)** > **DNS Server(DNS** 서버**)**를 사용하여 그룹을 할당합니다.
dynamic-access-policy-config **dynamic-access-policy-record**	—
enable	—
event	—
failover	—
fips	—
firewall	Device Manager에서는 라우팅 방화벽 모드만 지원합니다.
hostname	**Device(**디바이스**)** > **System Settings(**시스템 설정**)** > **Hostname(**호스트 이름**)**을 사용합니다.
hpm	이 기능은 threat defense 시스템에 적용되지 않습니다.
http	**Device(**디바이스**)** > **System Settings(**시스템 설정**)** > **Management Access(**관리 액세스**)**의 **Data Interfaces(**데이터 인터페이스**)** 탭을 사용합니다.
inline-set	—

금지된 **CLI** 명령	참고
interface (BVI, 관리, 이더넷, GigabitEthernet 및 하위 인터페이스용)	부분적으로 차단되었습니다. **Device(**디바이스**)** > **Interfaces(**인터페이스**)** 페이지에서 물리적 인터페이스, 하위 인터페이스 및 브리지 가상 인터페이스를 구성합니다. 그러면 FlexConfig를 사용하여 추가 옵션을 구성할 수 있습니다. 그러나 다음과 같은 **interface** 모드 명령은 이러한 유형의 인터페이스에 대해 금지되어 있습니다. **cts** **ip address** **ip address dhcp** **ipv6 address** **ipv6 enable** **ipv6 nd dad** **ipv6 nd suppress-ra** **mode** **nameif** **security-level** **shutdown** **zone-member**
vni, **redundant**, **tunnel**용 **interface**	**Device(**디바이스**)** > **Interfaces(**인터페이스**)**페이지에서 인터페이스를 구성합니다. Device Manager에서는 이러한 유형의 인터페이스를 지원하지 않습니다.
ip audit	이 기능은 threat defense 시스템에 적용되지 않습니다. 대신, 액세스 제어 규칙을 사용하여 침입 정책을 적용합니다.
ip-client	데이터 인터페이스를 관리 게이트웨이로 사용하도록 시스템을 구성하려면 **Device(**디바이스**)** > **System Settings(**시스템 설정**)** > **Management Interface(**관리 인터페이스**)**를 사용합니다.
ip local pool	주소 풀을 구성하려면 **Device(**디바이스**)** > **Remote Access VPN(**원격 액세스 **VPN)**을 사용합니다.
ipsec	—
ipv6	스마트 CLI IPv6 접두사 목록 개체를 생성한 다음, 스마트 CLI BGP 개체에 이를 사용하여 IPv6용 접두사 목록 필터링을 구성할 수 있습니다.
ipv6-vpn-addr-assign	주소 풀을 구성하려면 **Device(**디바이스**)** > **Remote Access VPN(**원격 액세스 **VPN)**을 사용합니다.

금지된 **CLI** 명령	참고
isakmp	**Device(**디바이스**)** > **Site-to-Site VPN(**사이트 대 사이트 **VPN)**을 사용합니다.
jumbo-frame	어떤 인터페이스든지 MTU를 기본값인 1500을 넘도록 증가시키는 경우, 시스템에서는 점보 프레임 지원을 자동으로 활성화합니다.
ldap	—
license-server	**Device(**디바이스**)** > **Smart License(**스마트 라이선스**)**를 사용합니다.
logging	**Objects(**개체**)** > **Syslog Servers(Syslog** 서버**)** 및 **Device(**디바이스**)** > **System Settings(**시스템 설정**)** > **Logging Settings(**로깅 설정**)**를 사용합니다. 그러나 **logging history** 명령은 FlexConfig에서 컨피그레이션할 수 있습니다.
management-access	—
migrate	**Device(**디바이스**)** > **Remote Access VPN(**원격 액세스 **VPN)** 및 **Device(**디바이스**)** > **Site-to-Site VPN(**사이트 대 사이트 **VPN)**을 사용하여 IKEv2 지원을 활성화합니다.
mode	Device Manager은 단일 컨텍스트 모드만 지원합니다.
mount	—
mtu	**Device(**디바이스**)** > **Interfaces(**인터페이스**)**에서 인터페이스당 MTU를 구성합니다.
nat	**Policies(**정책**)** > **NAT**를 사용합니다.
ngips	—
ntp	**Device(**디바이스**)** > **System Settings(**시스템 설정**)** > **NTP**를 사용합니다.
object-group network **object network**	**Objects(**개체**)** > **Network(**네트워크**)**를 사용합니다. FlexConfig에서 네트워크 개체 또는 그룹을 생성할 수는 없지만, 템플릿 내부에서 개체 관리자에 정의되어 있는 네트워크 개체 및 그룹을 변수로 사용할 수는 있습니다.

금지된 **CLI** 명령	참고
object service |natorigsvc **object service |natmappedsvc**	**object service** 명령은 일반적으로 허용되지만 이름이 |natorigsvc 또는 |natmappedsvc로 지정된 내부 개체는 편집할 수 없습니다. 이러한 이름에서 세로 막대는 제한된 개체 이름의 첫 번째 문자로, 의도적으로 사용되는 것입니다.
passwd **password**	—
password-policy	—
policy-list	스마트 CLI 정책 목록 개체를 생성한 다음, 스마트 CLI BGP 개체에 이를 사용하여 정책 목록을 구성할 수 있습니다.
policy-map 하위 명령	다음 명령은 정책 맵에서 구성할 수 없습니다. **priority** **police** **match tunnel-group**
prefix-list	스마트 CLI IPv4 접두사 목록 개체를 생성한 다음, 스마트 CLI OSPF 또는 BGP 개체에 이를 사용하여 IPv4용 접두사 목록 필터링을 구성할 수 있습니다.
priority-queue	—
privilege	—
reload	reload 명령은 예약할 수 없습니다. 시스템에서는 **reload** 명령이 아닌 **reboot** 명령을 사용해 재시작합니다.
rest-api	이 기능은 threat defense 시스템에 적용되지 않습니다. REST API는 항상 설치 및 활성화되어 있습니다.
route	**Device(**디바이스**)** > **Routing(**라우팅**)**을 사용하여 정적 경로를 구성합니다.
route-map	스마트 CLI 경로 맵 개체를 생성한 다음, 스마트 CLI OSPF 또는 BGP 개체에 이를 사용하여 경로 맵을 구성할 수 있습니다.
router bgp	BGP용 스마트 CLI 템플릿을 사용합니다.
router eigrp	EIGRP용 스마트 CLI 템플릿을 사용합니다.
router ospf	OSPF용 스마트 CLI 템플릿을 사용합니다.
scansafe	이 기능은 threat defense 시스템에 적용되지 않습니다. 대신, 액세스 제어 규칙에서 URL 필터링을 구성합니다.

금지된 **CLI** 명령	참고
setup	이 기능은 threat defense 시스템에 적용되지 않습니다.
sla	—
snmp-server	FTP API SNMP 리소스를 사용하여 SNMP를 구성합니다.
ssh	**Device(**디바이스**)** > **System Settings(**시스템 설정**)** > **Management Access(**관리 액세스**)**의 **Data Interfaces(**데이터 인터페이스**)** 탭을 사용합니다.
ssl	**Device(**디바이스**)** > **System Settings(**시스템 설정**)** > **SSL Settings(SSL** 설정**)**를 사용합니다.
telnet	Threat Defense는 텔넷 연결을 지원하지 않습니다. 텔넷 대신 SSH를 사용하여 디바이스 CLI에 액세스합니다.
time-range	—
tunnel-group	**Device(**디바이스**)** > **Remote Access VPN(**원격 액세스 **VPN)** 및 **Device(**디바이스**)** > **Site-to-Site VPN(**사이트 대 사이트 **VPN)**을 사용합니다.
tunnel-group-map	**Device(**디바이스**)** > **Remote Access VPN(**원격 액세스 **VPN)** 및 **Device(**디바이스**)** > **Site-to-Site VPN(**사이트 대 사이트 **VPN)**을 사용합니다.
user-identity	**Policies(**정책**)** > **Identity(ID)**를 사용합니다.
username	CLI 사용자를 생성하려면 디바이스에 대한 SSH 또는 콘솔 세션을 열고 **configure user** 명령을 사용합니다.
vpdn	—
vpn	—
vpn-addr-assign	—
vpnclient	—
vpn-sessiondb	—
vpnsetup	—
webvpn	—
zone	—
zonelabs-integrity	이 기능은 threat defense 시스템에 적용되지 않습니다.

스마트 **CLI** 템플릿

다음 표에서는 기능을 기반으로 스마트 CLI 템플릿을 설명합니다.

참고 또한 Smart CLI 템플릿을 사용하여 OSPF 및 BGP를 구성합니다. 그러나 이러한 템플릿은 Advanced Configuration(고급 구성) 페이지가 아닌 **Device(**디바이스**)** > **Routing(**라우팅**)** 페이지를 통해 사용할 수 있습니다.

기능	템플릿	설명
개체: AS 경로	ASPath	라우팅 프로토콜 개체에 사용할 ASPath 개체를 생성합니다.
개체: 액세스 목록	확장 액세스 목록 표준 액세스 목록	라우팅 개체에 사용할 확장 또는 표준 ACL을 생성합니다. ACL을 사용하는 허용된 명령을 구성하는 FlexConfig 개체에서 이러한 개체를 이름으로 참조할 수도 있습니다.
개체: 커뮤니티 목록	확장 커뮤니티 목록 표준 커뮤니티 목록	라우팅 개체에 사용할 확장 또는 표준 커뮤니티 목록을 생성합니다.
개체: 접두사 목록	IPV4 접두사 목록 IPV6 접두사 목록	라우팅 개체에 사용할 IPv4 또는 IPv6 접두사 목록을 생성합니다.
개체: 정책 목록	정책 목록	라우팅 개체에 사용할 정책 목록을 생성합니다.
개체: 경로 맵	경로 맵	라우팅 개체에 사용할 경로 맵을 생성합니다.

스마트 **CLI** 및 **FlexConfig**에 대한 지침 및 제한 사항

스마트 CLI 또는 FlexConfig를 통해 기능을 구성하는 경우 다음 사항에 유의하십시오.

- FlexConfig 개체에 정의된 명령은 스마트 CLI를 포함하여 기능에 대한 모든 명령이 device manager를 통해 정의된 이후에 구축됩니다. 따라서 이러한 명령이 디바이스에 실행되기 전에 구성 중인 개체, 인터페이스 등을 사용할 수 있습니다. 스마트 CLI 템플릿에서 FlexConfig 구축 항목을 사용해야 하는 경우, 스마트 CLI 템플릿을 생성 및 구축하기 전에 FlexConfig를 생성 및 구축하십시오. 예를 들어, OSPF 스마트 CLI 템플릿을 사용하여 EIGRP 경로를 재배포하려면 먼저 FlexConfig를 사용하여 EIGRP를 구성한 다음 OSPF 스마트 CLI 템플릿을 생성하십시오.
- FlexConfig를 통해 구성한 기능 또는 기능 중 일부를 제거하고 싶지만 스마트 CLI 템플릿이 이 기능을 참조하는 경우, 먼저 이 기능을 사용하는 스마트 CLI 템플릿에서 명령을 제거해야 합니다. 그런 다음 스마트 CLI 구성 기능이 이 기능을 더 이상 참조하지 않도록 컨피그레이션을 구축합니다. 그러면 FlexConfig에서 기능을 제거하고 컨피그레이션을 다시 구축하여 이 기능을 완전히 제거할 수 있습니다.

스마트 CLI 개체 구성

스마트 CLI 개체는 device manager의 다른 위치에서는 구성할 수 없는 기능을 정의합니다. 스마트 CLI 개체는 기능 구성에 대한 한 가지 수준의 지침을 제공합니다. 지정된 기능(템플릿)의 경우, 가능한 명령은 모두 사전 로드되며 입력하는 변수는 검증됩니다. 따라서 계속 CLI 명령을 사용하여 기능을 구성하더라도 스마트 CLI 개체는 FlexConfig 개체와 같은 자유 형식으로 사용할 수 없습니다.

스마트 CLI 템플릿에서 한 가지 수준의 지침을 제공하긴 하지만, 네트워크에 대해 올바르게 작동하는 값을 선택하려면 여전히 ASA 컨피그레이션 가이드 및 명령 참조를 읽어 명령 사용 방법을 파악해야 합니다. 작업할 ASA 컨피그레이션이 이미 있고 스마트 CLI 개체에서 동일한 명령 시퀀스를 구축하기만 하면 되는 것이 가장 좋습니다.

스마트 CLI 개체는 기능 영역에 따라 그룹화됩니다.

참고 정의하는 스마트 CLI 개체는 모두 구축됩니다. FlexConfig와 달리 여러 스마트 CLI 개체를 생성한 다음 구축할 개체를 선택할 수 없습니다. 구성하려는 기능에 대해서만 스마트 CLI 개체를 생성하십시오.

프로시저

단계 **1** **Device(**디바이스**)** > **Advanced Configuration(**고급 컨피그레이션**)**에서 **View Configuration(**컨피그레이션 보기**)**을 클릭합니다.

단계 **2** Advanced Configuration(고급 컨피그레이션) 목차의 **Smart CLI(**스마트 **CLI)** 아래에서 적절한 기능 영역을 클릭합니다.

단계 **3** 다음 중 하나를 수행합니다.

- 개체를 생성하려면 **+** 버튼을 클릭합니다.
- 개체를 수정하려면 개체의 수정 아이콘()을 클릭하십시오.

개체를 삭제하려면 해당 개체의 휴지통 아이콘()을 클릭합니다.

단계 **4** 개체의 Name(이름) 및 설명(선택 사항)을 입력합니다.

단계 **5** 구성 중인 기능에 대해 **CLI Template(CLI** 템플릿**)**을 선택합니다.

시스템에서 **Template(**템플릿**)** 창에 명령 템플릿을 로드합니다. 처음에는 필수 명령만 표시됩니다. 이 명령은 템플릿에 필요한 최소 컨피그레이션을 나타냅니다.

단계 **6** 템플릿에서 필요에 따라 변수를 입력하고 명령을 추가합니다.

ASA 또는 threat defense 디바이스(management center에서 관리되는 디바이스)의 기존 컨피그레이션으로 작업하는 것이 가장 좋습니다. 컨피그레이션을 사용할 수 있는 경우, IP 주소 및 인터페이스 이름 등의 변수를 네트워크의 이 특정 디바이스 위치에 적절하게 변경하여 템플릿이 해당 컨피그레이션을 따르도록 설정하기만 하면 됩니다.

템플릿 내용 입력에 대한 몇 가지 팁은 다음과 같습니다.

- 변수의 값을 선택하려면 변수를 클릭하고 적절한 값을 입력하거나 목록에서 선택합니다(값이 열거된 경우). 입력해야 하는 변수 위에 마우스를 올려놓으면 숫자 범위와 같이 해당 옵션에 대해 유효한 값이 표시됩니다. 권장 값이 표시되는 경우도 있습니다.

  예를 들어 OSPF 템플릿에서 필수 명령인 **router ospf** *process-id* 위에 마우스를 올려놓으면 "Process ID (1-65535)(프로세스 ID(1-65535))"가 표시되며, *process-id*를 클릭하면 이 필드가 강조 표시됩니다. 원하는 숫자를 입력하기만 하면 됩니다.

- 변수의 옵션을 선택할 때 옵션 구성에 사용 가능한 추가 명령이 있는 경우, 이러한 명령은 자동으로 표시되며 적절하게 비활성화 또는 활성화됩니다. 이러한 추가 명령을 확인합니다.

- 템플릿 위에 있는 **Show**/**Hide Disabled(**비활성화된 항목 표시/숨기기**)** 링크를 사용하여 보기를 제어합니다. 비활성화된 명령은 구성되지 않으며, 구성하려면 이러한 명령을 표시해야 합니다. 전체 템플릿을 보려면 템플릿 위에 있는 **Show Disabled(**비활성화된 항목 표시**)** 링크를 클릭합니다. 구성되는 명령만 보려면 표 위에 있는 **Hide Disabled(**비활성화된 항목 숨기기**)** 링크를 클릭합니다.

- 개체를 마지막으로 저장한 이후에 편집한 내용을 모두 지우려면 템플릿 위에 있는 **Reset(**재설정**)** 링크를 클릭합니다.

- 선택 사항 명령을 활성화하려면 줄 번호의 왼쪽에서 **+** 버튼을 클릭합니다.

- 선택 사항 명령을 비활성화하려면 줄 번호의 왼쪽에서 **-** 버튼을 클릭합니다. 줄을 편집한 경우 편집한 내용은 삭제되지 않습니다.

- 명령을 중복하려면 Options(옵션)의 **...** 버튼을 클릭하고 **Duplicate(**중복**)**를 선택합니다. 두 번 이상 명령을 입력하는 것이 유효한 경우에만 명령 중복이 허용됩니다.

- 중복된 명령을 삭제하려면 Options(옵션)의 **...** 버튼을 클릭하고 **Delete(**삭제**)**를 선택합니다. 기본 템플릿에 포함되어 있는 명령은 삭제할 수 없습니다.

단계 **7** **OK(**확인**)**를 클릭합니다.

FlexConfig 정책 구성

FlexConfig 정책은 간단히 말해 디바이스 컨피그레이션에 구축하려는 FlexConfig 개체의 목록입니다. 정책에 포함된 개체만 구축되며 다른 개체는 모두 간단히 정의되고 사용되지 않습니다.

FlexConfig 개체에 정의된 명령은 스마트 CLI를 포함하여 기능에 대한 모든 명령이 device manager를 통해 정의된 이후에 구축됩니다. 따라서 이러한 명령이 디바이스에 실행되기 전에 구성 중인 개체, 인터페이스 등을 사용할 수 있습니다. 스마트 CLI 템플릿에서 FlexConfig 구축 항목을 사용해야 하는 경우, 스마트 CLI 템플릿을 생성 및 구축하기 전에 FlexConfig를 생성 및 구축하십시오. 예를 들어, OSPF 스마트 CLI 템플릿을 사용하여 EIGRP 경로를 재배포하려면 먼저 FlexConfig를 사용하여 EIGRP를 구성한 다음 OSPF 스마트 CLI 템플릿을 생성하십시오.

참고 기능에 스마트 CLI 템플릿이 있는 경우, FlexConfig를 사용해서는 이를 구성할 수 없으므로 스마트 CLI 개체를 사용해야 합니다.

시작하기 전에

FlexConfig 개체를 생성합니다. 다음 주제를 참조하십시오.

- FlexConfig 개체 구성, 15 페이지
- FlexConfig 개체에서 변수 생성, 17 페이지
- 비밀 키 개체 구성, 25 페이지

프로시저

단계 **1** **Device(**디바이스**)** > **Advanced Configuration(**고급 컨피그레이션**)**에서 **View Configuration(**컨피그레이션 보기**)**을 클릭합니다.

단계 **2** Advanced Configuration(고급 컨피그레이션) 목차에서 **FlexConfig** > **FlexConfig Policy(FlexConfig** 정책**)**를 클릭합니다.

단계 **3** **Group List(**그룹 목록**)**에서 개체 목록을 관리합니다.

- 개체를 추가하려면 **+** 버튼을 클릭합니다. 개체가 아직 없으면 **Create New FlexConfig Object(**새 **FlexConfig** 개체 생성**)**를 클릭하여 개체를 정의합니다.
- 개체를 삭제하려면 개체 항목의 오른쪽에서 **X** 버튼을 클릭합니다.

참고 각 개체는 완전히 독립적이며 다른 FlexConfig 개체에 정의되어 있는 컨피그레이션의 영향을 받지 않는 것이 좋습니다. 그러면 다른 개체에 영향을 주지 않고 개체를 추가하거나 제거할 수 있습니다.

단계 **4** **Preview(**미리보기**)** 창에서 제안된 명령을 평가합니다.

Expand(펼치기**)** 버튼을 클릭하면 화면을 넓혀 긴 명령을 더 자세히 확인할 수 있습니다(확인 후에는 **Collapse(**접기**)**를 클릭).

미리보기는 변수를 평가하고 실행할 올바른 명령을 생성하므로 이러한 명령이 올바르고 유효한지 확인합니다. 명령으로 인해 디바이스를 사용할 수 없게 만드는 오류 또는 불량한 컨피그레이션이 발생하지 않는지 확인해야 합니다.

주의 시스템에서는 명령을 검증하지 않습니다. 따라서 유효하지 않은 명령이나 디바이스를 파괴하는 명령을 구축하게 될 수도 있습니다. 변경 사항을 구축하기 전에 매우 신중하게 미리보기를 확인하십시오.

단계 **5** **Save(**저장**)**를 클릭합니다.

다음에 수행할 작업

FlexConfig 정책을 편집하고 나서 다음 구축 결과를 주의 깊게 검토합니다. 오류가 있으면 개체에서 CLI를 수정합니다. FlexConfig 정책 트러블슈팅, 26 페이지의 내용을 참조하십시오.

FlexConfig 개체 구성

FlexConfig 개체에는 device manager를 사용하여 달리 구성할 수 없는 특정 기능을 구성하는 데 필요한 ASA 명령이 포함되어 있습니다. 오타 없이 올바른 명령 시퀀스를 입력해야 합니다. 시스템에서는 FlexConfig 개체의 콘텐츠를 검증하지 않습니다.

구성하려는 각 일반 기능에 대해 별도의 개체를 생성하는 것이 좋습니다. 예를 들어, 배너를 정의하고 RIP 라우팅 프로토콜도 구성하려면 별도의 개체 2개를 사용합니다. 별도의 개체에서 기능을 분리하면 구축할 개체를 더 쉽게 선택할 수 있으며 트러블슈팅도 더 간단히 수행할 수 있습니다.

참고 **enable** 및 **configure terminal** 명령은 포함하지 마십시오. 시스템에서 컨피그레이션 명령에 대해 올바른 모드를 자동으로 시작합니다.

프로시저

단계 **1** **Device(**디바이스**)** > **Advanced Configuration(**고급 컨피그레이션**)**에서 **View Configuration(**컨피그레이션 보기**)**을 클릭합니다.

단계 **2** Advanced Configuration(고급 컨피그레이션) 목차에서 **FlexConfig** > **FlexConfig Objects(FlexConfig** 개체**)**를 클릭합니다.

단계 **3** 다음 중 하나를 수행합니다.

- 개체를 생성하려면 + 버튼을 클릭합니다.
- 개체를 수정하려면 개체의 수정 아이콘( )을 클릭하십시오.

참조되지 않는 개체를 삭제하려면 해당 개체의 휴지통 아이콘( )을 클릭합니다.

단계 **4** 개체의 Name(이름) 및 설명(선택 사항)을 입력합니다.

단계 **5** **Variables(**변수**)** 섹션에서 개체 본문에 사용할 변수를 생성합니다.

생성해야 하는 유일한 변수는 device manager 내에 정의되어 있는 개체를 가리키는 변수입니다(특히, 네트워크, 포트 및 암호 키 변수 유형 또는 이름이 지정된 인터페이스를 가리키는 인터페이스 변수). 다른 변수 유형의 경우 간단히 개체 본문에 값을 입력하면 됩니다.

변수 생성 및 사용에 대한 자세한 내용은 FlexConfig 개체에서 변수 생성, 17 페이지를 참조하십시오.

단계 **6** **Template(**템플릿**)** 섹션에서 기능을 구성하는 데 필요한 ASA 명령을 입력합니다.

기능을 구성하기 위해서는 올바른 순서로 명령을 입력해야 합니다. ASA CLI 컨피그레이션 가이드를 사용하여 명령을 입력하는 방법에 대해 알아봅니다. 참조로 사용할 수 있는 ASA 또는 다른 threat defense 디바이스의 사전 테스트된 컨피그레이션 파일이 있는 것이 가장 좋습니다.

Mustache 표기법을 사용하여 변수를 참조하고 처리할 수도 있습니다. 자세한 내용은 FlexConfig 변수 참조 및 값 검색, 18 페이지를 참조하십시오.

개체 본문 생성에 대한 몇 가지 팁은 다음과 같습니다.

- 줄을 추가하려면 줄 끝에 커서를 놓고 Enter 키를 누릅니다.
- 변수를 사용하려면 이중 중괄호 사이에 변수 이름을 입력합니다(예: {{*variable_name*}}). 개체를 참조하는 변수의 경우, 검색 중인 값의 특성을 포함해야 합니다(예: {{*variable_name.attribute*}}). 사용 가능한 특성은 개체 유형에 따라 달라집니다. 자세한 내용은 변수 참조: {{variable}} 또는 {{{variable}}}, 18 페이지를 참조하십시오.
- 스마트 CLI 개체를 사용하려면 개체의 이름을 입력합니다. 스마트 CLI에 구성된 라우팅 프로세스를 참조해야 하는 경우 프로세스 식별자를 입력합니다. FlexConfig 개체의 스마트 CLI 개체 참조, 23 페이지를 참조하십시오.
- 본문의 크기를 더 키우거나 줄이려면 템플릿 본문 위의 **Expand/Collapse(**펼치기/접기**)** 링크를 클릭합니다.
- 개체를 마지막으로 저장한 이후에 변경한 사항을 지우려면 **Reset(**재설정**)** 링크를 클릭합니다.

단계 **7** **Negate Template(**무효화 템플릿**)** 섹션에서 개체 본문에 구성되어 있는 명령을 제거하거나 되돌리는 데 필요한 명령을 입력합니다.

Negate(무효화) 섹션은 매우 중요하며 다음의 두 가지 목적을 위해 사용됩니다.

- 구축을 간소화합니다. 본문에서 명령을 재구축하기 전에 시스템에서는 이러한 명령을 사용하여 먼저 컨피그레이션을 지우거나 실행 취소합니다. 이렇게 하면 구축이 정상적으로 이루어집니다.
- FlexConfig 정책에서 개체를 제거하여 이 기능을 제거하려는 경우, 시스템에서는 이러한 명령을 사용하여 디바이스에서 명령을 제거합니다.

개체 본문에서 CLI를 무효화하거나 되돌리는 데 필요한 명령을 제공하지 않으면 구축 시 개체 내의 명령뿐만 아니라 전체 디바이스의 컨피그레이션을 지우고 모든 정책을 재구축해야 할 수 있습니다. 이 경우, 구축에 더 오랜 시간이 걸리고 트래픽을 방해하게 됩니다. 개체 본문에 정의된 컨피그레이션을 취소하는 데 필요한 명령만 모두 사용했는지 확인합니다. 무효화 명령은 일반적으로 템플릿에 포함된 명령의 **no** 또는 **clear** 형식이지만 이미 활성화된 기능을 실제로 끄는 경우 "negate" 명령은 명령의 긍정 형식(기능을 활성화하는 형식)입니다.

ASA 컨피그레이션 가이드 및 명령 참조를 사용하여 적절한 명령을 파악합니다. 경우에 따라 단일 명령으로 컨피그레이션을 취소할 수 있습니다. 예를 들어 RIP를 컨피그레이션하는 개체에서 간단한 **no router rip** 명령은 하위 명령을 포함한 전체 **router rip** 컨피그레이션을 제거합니다.

마찬가지로 여러 줄로 된 배너를 생성하기 위해 여러 **banner login** 명령을 입력한 경우, 단일 **no banner login** 명령은 전체 로그인 배너를 무효화합니다.

템플릿이 여러 중첩 개체를 생성하는 경우 무효화 템플릿은 개체를 반대 순서로 제거해야 합니다. 즉, 개체를 삭제하기 전에 개체에 대한 참조를 먼저 제거해야 합니다. 예를 들어 ACL을 먼저 생성한 다

음 트래픽 클래스에서 참조하고, 정책 맵에서 해당 트래픽 클래스를 참조하고, 마지막으로 서비스 정책을 사용해 정책 맵을 활성화하는 경우 무효화 템플릿은 서비스 정책, 정책 맵, 트래픽 클래스, ACL을 차례로 제거하여 컨피그레이션을 실행 취소해야 합니다.

단계 **8** **OK(**확인**)**를 클릭합니다.

다음에 수행할 작업

FlexConfig 개체를 생성하기만 하면 이 개체가 구축되지 않으므로 개체를 FlexConfig 정책에 추가해야 합니다. 이때, FlexConfig 정책의 개체만 구축되기 때문에 FlexConfig 개체를 재정의할 수 있으며, 모든 개체를 자동으로 구축하지 않고 일부를 특별한 용도에 맞게 준비할 수 있습니다. FlexConfig 정책 구성, 13 페이지의 내용을 참조하십시오.

FlexConfig 개체에서 변수 생성

FlexConfig 개체 내에서 사용하는 변수는 해당 개체 자체 내에서 정의되며, 별도의 변수 목록은 없습니다. 따라서 변수를 정의한 다음 별도의 FlexConfig 개체에서 이를 사용할 수 없습니다.

변수는 다음과 같은 주요 이점을 제공합니다.

- 변수를 사용하면 device manager를 사용하여 정의된 개체를 가리킬 수 있습니다. 변수에는 네트워크, 포트 및 비밀 키 개체가 포함됩니다.
- 변수는 개체 본문에서 변경될 수 있는 값을 분리합니다. 따라서 값을 변경해야 하는 경우 해당 변수만 편집하면 되고 개체 본문을 편집할 필요는 없습니다. 변수는 특히 여러 명령줄에서 개체를 참조해야 하는 경우에 유용합니다.

이 절차에서는 FlexConfig 개체에 변수를 추가하는 프로세스에 대해 설명합니다.

프로시저

단계 **1** **Device(**디바이스**)** > **Advanced Configuration(**고급 컨피그레이션**)** 페이지에서 FlexConfig 개체를 편집하거나 생성합니다.

FlexConfig 개체 구성, 15 페이지의 내용을 참조하십시오.

단계 **2** **Variables(**변수**)** 섹션에서 다음 작업 중 하나를 수행합니다.

- 변수를 추가하려면 + 버튼을 클릭합니다(아직 정의된 변수가 없는 경우에는 **Add Variable(**변수 추가**)**을 클릭).
- 변수를 편집하려면 해당 변수의 편집 아이콘( )을 클릭합니다.

변수를 삭제하려면 해당 변수의 휴지통 아이콘( )을 클릭합니다. 템플릿 본문의 해당 변수 참조를 모두 제거합니다.

단계 **3** 변수의 이름과 설명(선택 사항)을 입력합니다.

단계 **4** 변수의 데이터 **Type(**유형**)**을 선택한 다음 값을 입력하거나 선택합니다.

다음과 같은 유형의 변수를 생성할 수 있습니다. 변수를 사용할 명령의 데이터 요건에 맞는 유형을 선택합니다.

- **String(**문자열**)** — 텍스트 문자열입니다. 예를 들어, 호스트 이름, 사용자 이름 등이 있습니다.
- **Numeric(**숫자**)** — 정수입니다. 쉼표, 10진수, 기호(예: 음수) 또는 16진수 표기법을 포함하지 마십시오. 정수가 아닌 경우에는 문자열 변수를 사용합니다.
- **Boolean(**부울**)** — 논리적 true/false입니다. True 또는 False 중 하나를 선택합니다.
- **Network(**네트워크**)** — Objects(개체) 페이지에 정의되어 있는 네트워크 개체 또는 그룹입니다. 네트워크 개체 또는 그룹을 선택합니다.
- **Port(**포트**)** — Objects(개체) 페이지에 정의되어 있는 TCP 또는 UDP 포트 개체입니다. 포트 개체를 선택합니다. 그룹을 선택하거나 다른 프로토콜에 대한 개체를 선택할 수는 없습니다.
- **Interface(**인터페이스**)** — Device(디바이스) > Interfaces(인터페이스) 페이지에 정의되어 있는 이름이 지정된 인터페이스입니다. 인터페이스를 선택합니다. 이름이 없는 인터페이스는 선택할 수 없습니다.
- **IP** — 넷마스크 또는 접두사 길이를 포함하지 않는 단일 IPv4 또는 IPv6 IP 주소입니다.
- **Secret(**비밀**)** — FlexConfig에 대해 정의된 비밀 키 개체입니다. 개체를 선택합니다. 비밀 키 개체 생성에 대한 자세한 내용은 비밀 키 개체 구성, 25 페이지를 참조하십시오.

단계 **5** Variable(변수) 대화 상자에서 **Add(**추가**)** 또는 **Save(**저장**)**를 클릭합니다.

이제 FlexConfig 개체의 본문 내에 변수를 사용할 수 있습니다. 변수를 참조하는 방법은 변수 유형에 따라 달라집니다. 이러한 변수를 사용하는 방법에 대한 내용은 다음 주제를 참조하십시오.

- 변수 참조: {{variable}} 또는 {{{variable}}}, 18 페이지
- 섹션 {{#key}} {{/key}} 및 역 섹션 {{^key}} {{/key}}, 22 페이지

단계 **6** FlexConfig Object(FlexConfig 개체) 대화 상자에서 **OK(**확인**)**를 클릭합니다.

FlexConfig 변수 참조 및 값 검색

FlexConfig는 Mustache를 템플릿 언어로 사용하지만 다음 섹션에 설명된 기능만 지원합니다. 이러한 기능을 사용하여 변수를 참조하고 해당 값을 검색 및 처리합니다.

변수 참조: {{variable}} 또는 {{{variable}}}

FlexConfig 개체 내에서 정의하는 변수를 참조하려면 다음 표기법을 사용합니다.

{{*variable_name*}}

또는

{{{*variable_name*}}}

이 표기법은 **Numeric(**숫자**)**, **String(**문자열**)**, **Boolean(**부울**)**, **IP** 유형의 변수를 포함하며 단일 값인 간단한 변수에 사용하기에 충분합니다. 변수에 &와 같은 특수 문자가 포함된 경우, 3중 괄호를 사용하십시오. 또는 모든 변수에 대해 항상 3중 괄호를 사용할 수 있습니다.

그러나 컨피그레이션 데이터베이스에서 개체로 모델링되는 요소를 가리키는 변수의 경우, 점 표기법을 사용하고 검색하려는 개체 특성의 이름을 포함해야 합니다. 이러한 특성 이름은 API Explorer에서 관련 개체 유형에 대해 모델을 검사하여 찾을 수 있습니다. **Secret(**비밀**)**, **Network(**네트워크**)**, **Port(**포트**)**, **Interface(**인터페이스**)** 유형의 변수를 사용하려면 다음과 같은 표기법을 사용해야 합니다.

{{*variable_name.attribute*}}

예를 들어, net-object1(네트워크 그룹이 아니라 네트워크 개체를 가리킴)이라는 이름의 네트워크 변수에서 주소를 검색하려면 다음과 같은 표기법을 사용합니다.

{{net-object1.value}}

개체 내에서 개체의 특성 값을 검색하려는 경우, 일련의 점으로 구분된 특성을 사용하여 원하는 값으로 드릴다운해야 합니다. 예를 들어, 인터페이스의 IP 주소는 인터페이스 개체에 대해 ipv4와 ipv6라는 이름의 하위 개체로 모델링됩니다. 따라서 int-inside(내부 인터페이스를 가리킴)라는 이름의 인터페이스 변수에 대한 IPv4 주소 및 서브넷 마스크를 검색하려면 다음과 같은 표기법을 사용합니다.

{{int-inside.ipv4.ipAddress.ipAddress}} {{int-inside.ipv4.ipAddress.netmask}}

참고 API Explorer를 열려면 More options(추가 옵션) 버튼(⋮)을 클릭하고 **API Explorer**를 선택합니다.

다음 표에는 변수 유형, 변수 유형을 참조하는 방법, 그리고 개체의 경우, API 모델의 이름과 사용할 가능성이 가장 높은 참조가 나와 있습니다.

변수 유형	참조 모델	설명

부울
(간단한 변수)
	변수**:**
	{{*variable_name*}}
	섹션**:**
	`{{#variable_name}}`
	`commands`
	`{{/variable_name}}`
	역 섹션**:**
	`{{^variable_name}}`
	`commands`
	`{{/variable_name}}`
		논리적 true/false입니다. 부울 변수는 주로 섹션 또는 역 섹션에 사용합니다. 예를 들어 기능을 주기적으로 또는 특별한 상황에서만 활성화해야 하는 경우, 부울 변수를 편집하여 명령 섹션을 설정 또는 해제할 수 있습니다.
		일부 개체의 경우 모델에 부울 특성도 있으며, 이를 사용하여 선택적으로 섹션을 처리할 수 있습니다.

변수 유형	참조 모델	설명

인터페이스
(개체 변수: API 모델이 인터페이스임)
	변수:
	{{*variable_name.attribute*}}
	섹션:
	`{{#variable_name.attribute}}`
	`commands`
	`{{/variable_name.attribute}}`
	역 섹션:
	`{{^variable_name.attribute}}`
	`commands`
	`{{/variable_name.attribute}}`
		Device(디바이스) > Interfaces(인터페이스) 페이지에 정의되어 있는 이름이 지정된 인터페이스입니다. 이름이 없는 인터페이스는 가리킬 수 없습니다.
		인터페이스 모델에서 사용할 수 있는 특성은 다양합니다. 또한, 인터페이스 모델은 예를 들어 IP 주소의 하위 개체를 포함합니다.
		유용하게 활용할 수 있는 몇 가지 주요 특성은 다음과 같습니다.
		• *variable_name*.**name**에서는 인터페이스의 논리적 이름을 반환합니다.
		• *variable_name*.**hardwareName**에서는 GigabitEthernet1/8과 같은 인터페이스 포트 이름을 반환합니다.
		• *variable_name*.**managementOnly**은 부울 값입니다. TRUE는 인터페이스가 관리 전용으로 정의되어 있음을 의미합니다. FALSE는 인터페이스가 디바이스를 통과하는 트래픽용임을 의미합니다. 이 옵션은 섹션 키로 사용할 수 있습니다.
		• *variable_name*.**ipv4.ipAddress.ipAddress**에서는 인터페이스의 IPv4 주소를 반환합니다.
		• *variable_name*.**ipv4.ipAddress.netmask**에서는 인터페이스의 IPv4 주소에 대한 서브넷 마스크를 반환합니다.

IP (간단한 변수)	변수: {{*variable_name*}}	넷마스크 또는 접두사 길이를 포함하지 않는 단일 IPv4 또는 IPv6 IP 주소입니다.

변수 유형	참조 모델	설명
네트워크 (개체 변수: API 모델이 네트워크 개체임)	변수**(**네트워크 개체**):** {{*variable_name*.*attribute*}} 섹션**(**그룹 개체**):** {{#*variable_name*.**networkObjects**}} commands referring to one of {{**value**}} {{**name**}} {{/*variable_name*.**networkObjects**}}	Objects(개체) 페이지에 정의되어 있는 네트워크 개체 또는 그룹입니다. 섹션을 사용하여 네트워크 그룹을 처리할 수 있습니다. 유용하게 활용할 수 있는 주요 특성은 다음과 같습니다. • {{*variable_name*.**name**}}에서는 네트워크 개체 또는 그룹의 이름을 반환합니다. • {{*variable_name*.**value**}}에서는 네트워크 개체(네트워크 그룹이 아님)의 IP 주소 콘텐츠를 반환합니다. 지정된 명령에 대해 콘텐츠 유형이 적절한 네트워크 개체를 사용해야 합니다(예: 서브넷 주소 대신 호스트 주소). • {{*variable_name*.**groups**}}에서는 네트워크 그룹에 포함된 네트워크 개체의 목록을 반환합니다. 이 특성은 네트워크 그룹을 가리키는 변수에만 사용하고, 섹션 태그에서 사용하여 그룹의 콘텐츠를 반복적으로 처리하십시오. {{**value**}} 또는 {{**name**}}(을)를 사용하여 각 네트워크 개체의 콘텐츠를 차례대로 검색하십시오.
숫자 (간단한 변수)	변수**:** {{*variable_name*}}	정수 숫자입니다. 쉼표, 10진수, 기호(예: 음수) 또는 16진수 표기법을 포함하지 마십시오. 정수가 아닌 경우에는 문자열 변수를 사용합니다.
Port(포트) (개체 변수: API 모델이 포트 개체, TCP 포트 또는 UDP 포트임)	변수**:** {{*variable_name*.*attribute*}}	Objects(개체) 페이지에 정의되어 있는 TCP 또는 UDP 포트 개체입니다. 이는 포트 그룹이 아니라 포트 개체여야 합니다. 유용하게 활용할 수 있는 주요 특성은 다음과 같습니다. • {{*variable_name*.**port**}}에서는 포트 번호를 반환합니다. 프로토콜은 포함되지 않습니다. • {{*variable_name*.**name**}}에서는 포트 개체의 이름을 반환합니다.
기밀 (개체 변수: API 모델이 비밀임)	변수**:** {{*variable_name*.**password**}} 또는 {{{*variable_name*.**password**}}}	FlexConfig용으로 정의되어 있는 비밀 키 개체입니다. 암호화된 문자열을 반환하는 **password** 속성에 대한 참조만 수행해야 합니다. 암호에 &와 같은 특수 문자가 포함된 경우, 3중 괄호를 사용하십시오.
문자열 (간단한 변수)	변수**:** {{*variable_name*}}	텍스트 문자열입니다. 예를 들어, 호스트 이름, 사용자 이름 등이 있습니다.

섹션 {{#key}}{{/key}} 및 역 섹션 {{^key}}{{/key}}

섹션 또는 역 섹션은 키를 처리 기준으로 사용하는, 섹션 시작과 끝 태그 사이의 명령 블록입니다. 섹션 처리 방법은 일반 섹션 또는 역 섹션 중 어느 것인지에 따라 달라집니다.

- 일반 섹션(또는 간단하게 섹션)은 키가 TRUE이거나 콘텐츠가 비어 있지 않은 경우 처리됩니다. 키가 FALSE이거나 개체에 콘텐츠가 없는 경우 섹션의 명령은 구성되지 않으며, 섹션은 우회됩니다.

  일반 섹션의 구문은 다음과 같습니다.

  ```
  {{#key}}
  one or more commands
  {{/key}}
  ```

- 역 섹션은 섹션과 반대로 키가 FALSE이거나 개체에 콘텐츠가 없는 경우 처리됩니다. 키가 TRUE이거나 개체에 콘텐츠가 있는 경우, 역 섹션은 우회됩니다.

  역 섹션의 구문은 다음과 같습니다. 차이점은 해시 태그가 캐럿 기호로 바뀐 것뿐입니다.

  ```
  {{^key}}
  one or more commands
  {{/key}}
  ```

다음 주제에서는 섹션 및 역 섹션의 주요 활용 사례에 대해 설명합니다.

다중 값 변수를 처리하는 방법

다중 값 변수 처리의 기본 예는 네트워크 그룹을 가리키는 네트워크 변수입니다. 그룹에는 여러 개체(**objects** 속성 아래)가 포함되어 있으므로 네트워크 그룹에서 값을 반복적으로 검토하여 동일한 명령을 다양한 값으로 여러 번 컨피그레이션할 수 있습니다.

개체 그룹은 개체 속성 내에 포함된 네트워크 개체를 정의하지만, 해당 개체는 포함된 개체의 콘텐츠를 포함하지 않습니다. 대신 **networkObjects** 속성을 사용하여 포함된 개체의 콘텐츠를 가져옵니다.

예를 들어 호스트 192.168.10.0, 192.168.20.0 및 192.168.30.0을 사용하는 net-group이라는 이름의 네트워크 그룹이 있는 경우, 다음 기술을 사용하여 RIP 라우팅의 각 주소에 대해 네트워크 명령을 구성할 수 있습니다. 섹션 시작 **value**에서 사용하면 멤버 개체에서 **net-group.networkObjects** 값 속성을 가져옴을 나타내므로 네트워크 개체의 속성만 사용합니다. FlexConfig 개체 내에서 "값" 특성에 대해 별도의 변수를 생성하지 마십시오.

```
router rip
{{#net-group.networkObjects}}
 network {{value}}
{{/net-group.networkObjects}}
```

시스템은 섹션 구조를 다음과 같이 변환합니다.

```
router rip
   network 192.168.10.0
   network 192.168.20.0
```

```
network 192.168.30.0
```

부울 값 또는 빈 개체를 기준으로 선택적 처리를 수행하는 방법

참고 이 항목의 예시는 설명만을 목적으로 합니다. 예를 들어, FlexConfig를 사용하여 버전 6.7 이상의 SNMP를 설정할 수 없습니다. 대신 threat defense API SNMP 리소스를 사용해야 합니다.

섹션이 섹션 시작 태그의 변수 콘텐츠가 TRUE이거나 개체가 비어 있지 않은 경우에는 처리되고, 부울 값이 FALSE이거나 비어 있는 경우(예: 빈 개체)에는 우회됩니다.

이 기능은 주로 부울 값에 사용됩니다. 예를 들어 부울 변수를 생성하고 이 변수가 적용되는 섹션 내에 명령을 입력할 수 있습니다. 그러면 FlexConfig 개체에서 명령의 섹션을 활성화 또는 비활성화해야 하는 경우, 단순히 부울 변수의 값만 변경하면 되며 코드에서 해당 줄을 삭제하지 않아도 됩니다. 따라서 이 기능은 켜거나 끄기가 쉽습니다.

예를 들어 FlexConfig를 사용하여 SNMP를 활성화하는 경우, SNMP 트랩을 해제할 수 있습니다. 즉, enable-traps라는 이름의 부울 변수를 생성하고 처음에 TRUE로 설정하고 나서 트랩을 해제해야 하는 경우 변수를 편집하여 FALSE로 변경하고 개체를 저장한 다음 컨피그레이션을 다시 구축하기만 하면 됩니다. 명령 시퀀스는 다음과 같이 표시됩니다.

```
snmp-server enable
snmp-server host inside 192.168.1.5
snmp-server community clearTextString
{{#enable-traps}}
snmp-server enable traps all
{{/enable-traps}}
```

또한, 개체 내의 부울 값을 기반으로 이러한 유형의 처리를 수행할 수 있습니다. 예를 들어 인터페이스에 일부 특성을 구성하기 전에 인터페이스가 관리 전용인지를 확인할 수 있습니다. 다음 예에서 int-inside는 inside라는 이름의 인터페이스를 가리키는 인터페이스 변수입니다. FlexConfig는 인터페이스가 관리 전용으로 설정되지 않은 경우에만 인터페이스에 EIGRP 관련 인터페이스 옵션을 구성합니다. 부울 값이 FALSE인 경우에만 명령이 구성되도록 역 섹션을 사용할 수 있습니다.

```
router eigrp 2
 network 192.168.1.0 255.255.255.0
{{^int-inside.managementOnly}}
interface {{int-inside.hardwareName}}
 hello interval eigrp 2 60
 delay 200
{{/int-inside.managementOnly}}
```

FlexConfig 개체의 스마트 CLI 개체 참조

FlexConfig 개체를 생성할 때는 변수를 사용해 device manager 내에서 구성할 수 있는 개체를 가리킬 수 있습니다. 예를 들어 인터페이스 요소 또는 네트워크 개체를 가리키는 변수를 생성할 수 있습니다.

그러나 같은 방식으로 스마트 CLI 개체를 가리킬 수는 없습니다.

대신 FlexConfig 정책에서 사용해야 하는 스마트 CLI 개체를 생성하는 경우 적절한 위치에 스마트 CLI 개체의 이름만 입력하면 됩니다.

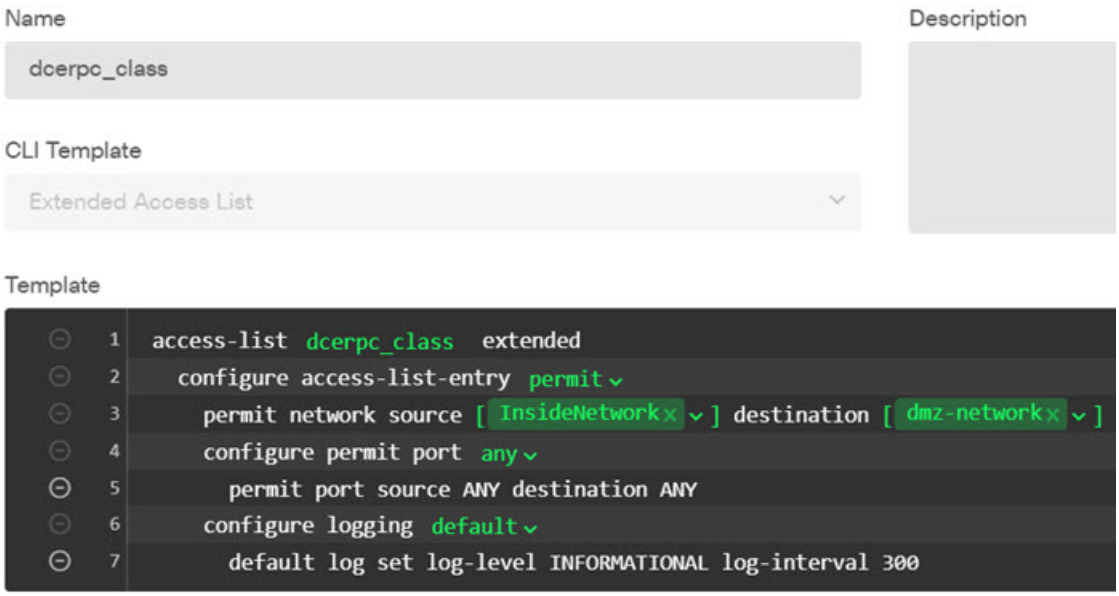
프로토콜 검사를 구성할 때 확장된 액세스 목록을 트래픽 클래스로 사용하려는 경우를 예로 들어 보겠습니다. 확장된 액세스 목록용 스마트 CLI 개체가 있으므로 해당 스마트 CLI 개체를 사용하여 ACL을 생성해야 하며, FlexConfig 개체에서 **access-list** 명령을 사용할 수는 없습니다.

예를 들어 192.168.1.0/24 및 192.168.2.0/24 네트워크 간에 DCERPC 검사를 전역적으로 활성화하려는 경우 다음을 수행합니다.

프로시저

단계 **1** 두 네트워크용으로 각기 별도의 네트워크 개체를 생성합니다. 예를 들어 InsideNetwork 및 dmz-network를 생성합니다.

단계 **2** 스마트 CLI 확장된 액세스 목록 개체에서 이러한 개체를 사용합니다.



단계 **3** 이름을 기준으로 스마트 CLI 개체를 가리키는 FlexConfig 개체를 생성합니다.

예를 들어 개체 이름이 "dcerpc_class"인 경우 FlexConfig 개체는 다음과 같을 수 있습니다. 무효화 템플릿에서는 스마트 CLI 개체를 통해 생성한 액세스 목록을 무효화하지 않습니다. 해당 개체는 실제로 FlexConfig를 통해 생성된 것이 아니기 때문입니다.

Template

```
1  class-map dcerpc_inspection
2   match access-list dcerpc_class
3  policy-map global_policy
4   class dcerpc_inspection
5    inspect dcerpc
```

Negate Template

```
1  policy-map global_policy
2   no class dcerpc_inspection
3  no class-map dcerpc_inspection
```

단계 **4** FlexConfig 정책에 개체를 추가합니다.

비밀 키 개체 구성

비밀 키 개체는 비밀번호 또는 민감한 문자열이 명확하게 표시되지 않게 하기 위해 사용됩니다. FlexConfig 개체 또는 스마트 CLI 템플릿에 사용된 문자열을 누군가가 보지 못하게 하려면 해당 문자열에 대해 비밀 키 개체를 생성합니다.

프로시저

단계 **1** 목차에서 **Objects(**개체**)**와 **Secret Keys(**비밀 키**)**를 차례로 선택합니다.

단계 **2** 다음 중 하나를 수행합니다.

- 개체를 생성하려면 + 버튼을 클릭합니다.
- 개체를 수정하려면 개체의 수정 아이콘( )을 클릭하십시오.

참조되지 않는 개체를 삭제하려면 해당 개체의 휴지통 아이콘( )을 클릭합니다.

단계 **3** 개체의 Name(이름) 및 설명(선택 사항)을 입력합니다.

단계 **4** **Password(**비밀번호**)** 및 **Confirm Password(**비밀번호 확인**)** 필드에 비밀번호 또는 기타 비밀 문자열을 입력합니다.

그러면 입력한 텍스트가 명확하지 않게 표시됩니다.

단계 **5** **OK(**확인**)**를 클릭합니다.

다음에 수행할 작업

- 새 개체의 경우 FlexConfig에서 이를 사용하려면 FlexConfig 개체를 편집하고 비밀 키 유형의 변수를 생성한 다음 해당 개체를 선택합니다. 그런 다음 개체 본문 내의 변수를 참조하십시오. 자세한 내용은 FlexConfig 개체에서 변수 생성, 17 페이지의 내용을 참고하십시오.
- FlexConfig 정책의 일부인 FlexConfig 개체에서 사용된 기존 개체를 편집하는 경우, 컨피그레이션을 구축하여 디바이스를 새로운 문자열로 업데이트해야 합니다.
- 스마트 CLI 템플릿에서 명령에 비밀 키가 필요한 경우 관련 속성을 편집할 때 해당하는 개체의 목록이 표시됩니다. 이에 맞게 적절한 키를 선택합니다.

FlexConfig 정책 트러블슈팅

FlexConfig 정책을 편집하고 나서 다음 구축 결과를 주의 깊게 검토합니다. Pending Changes(보류 중인 변경 사항) 대화 상자에 "Last Deployment Failed(마지막 구축 실패함)" 메시지가 표시되면 **See Details(**세부 사항 참조**)** 링크를 클릭합니다. 이 링크를 클릭하면 감사 로그로 이동하여 실패한 구축 작업을 찾을 수 있습니다. 특정 오류 메시지를 찾으려면 해당 작업을 엽니다.

FlexConfig 문제로 인해 구축에 실패한 경우, 세부 사항에서는 올바르지 않은 명령을 사용하는 FlexConfig 개체에 대해 언급하며 실패한 명령을 표시합니다. 이 정보를 사용하여 개체를 수정하고 다시 구축을 시도합니다. 개체 이름은 링크이므로, 이 링크를 클릭하여 개체에 대한 편집 대화 상자를 엽니다.

예를 들어 TCP MSS(최대 TCP 세그먼트 크기)를 구성하려는 경우, **sysopt connection tcpmss** 명령을 사용하여 이 설정을 제어할 수 있습니다. device manager에서 구성한 경우, 이 옵션에 대한 threat defense 기본값은 0입니다(ASA 기본값은 1380).

ASA 기본값은 기본 MTU(1500)를 사용하는 인터페이스에서 IPv4 VPN을 실행하는 경우 최적의 처리를 수행할 수 있도록 설계되어 있습니다. 시스템에서는 VPN 헤더에 120바이트를 필요로 합니다. IPv6의 경우, 시스템에서는 140바이트를 필요로 합니다. threat defense의 기본값인 0을 사용하면 엔드포인트에서 MSS를 협상할 수 있습니다. 이 설정은 정상적인 트래픽에 대해, 특히 1500을 넘는 MTU를 비롯하여 디바이스의 인터페이스 전체에서 서로 다른 MTU를 사용하는 경우, 이상적입니다. TCP MSS는 인터페이스별 설정이 아니라 글로벌 설정이므로 상당한 비율의 트래픽이 VPN을 통과하며 과도한 프래그멘테이션이 발생하는 경우에만 변경할 수 있습니다. 이 경우, TCP MSS를 MTU에서 120을 뺀 값(IPv4의 경우) 또는 MTU에서 140을 뺀 값(IPv6의 경우)으로 설정하고 모든 인터페이스에 대해 동일한 MTU를 사용할 수 있습니다. MSS를 명시적으로 설정하더라도 TLS/SSL 암호 해독 또는 서버 검색과 같은 구성 요소에 특정 MSS가 필요한 경우, 인터페이스 MTU를 기반으로 해당 MSS를 설정하고 MSS 설정을 무시합니다.

실례를 들기 위해 TCP MSS를 3바이트로 설정했다고 가정해 보겠습니다. 이 경우 명령에서 48바이트를 최솟값으로 사용하므로 다음과 유사한 구축 오류가 발생하게 됩니다.

오류는 다음과 같은 요소로 구성됩니다.

1. 구축 오류 메시지에는 오류를 야기한 FlexConfig 개체의 이름이 포함되어 있습니다. 개체 이름은 편집 대화 상자와 연결되므로 신속하게 개체를 열고 오류를 수정할 수 있습니다. 메시지의 첫 번째 문장에 이 내용이 포함됩니다.
2. "ERROR(오류):"로 시작되는 텍스트는 디바이스에서 반환된 메시지입니다. 이 메시지는 잘못된 명령을 입력한 경우 SSH 클라이언트를 포맷하지 않고 ASA가 대응하는 방식을 정확히 나타냅니다. 이 예에서 오류 메시지는 "ERROR: [3] is smaller than the minimum allowed MSS of 48 by RFC 791.(오류: [3]이 RFC 791에서 허용되는 최소 MSS 값인 48보다 작습니다.)"입니다. "Config Error(컨피그레이션 오류)"로 시작되는 텍스트에 오류 메시지를 생성한 특정 행이 나와 있습니다.
3. 검은색 텍스트는 오류를 야기한 FlexConfig 개체의 실제 줄이며, 이 줄은 수정해야 합니다. 이 예에서 MTU가 1500인 인터페이스(일반적인 상황)에서 IPv4 VPN 트래픽을 수용하려면 3을 1380으로 변경합니다.

   이 예를 수정할 때 CLI 콘솔을 열어 두고 **show running-config all sysopt**(을)를 사용하여 **sysopt** 명령 설정을 모두 확인할 수 있습니다. 대부분의 **sysopt** 명령에는 대부분의 용도에 적절한 기본 설정이 있으므로 실행 중인 컨피그레이션에는 기본 설정이 표시되지 않습니다. **all** 키워드를 사용하면 출력에 이러한 기본 설정이 포함됩니다.

FlexConfig의 예시

다음 주제에서는 FlexConfig를 사용하여 기능을 구성하는 몇 가지 예시를 제공합니다.

전역 기본 검사를 활성화/비활성화하는 방법

동적으로 할당된 포트의 개방형 보조 채널이나 사용자 데이터 패킷에 IP 주소 지정 정보를 포함하는 프로토콜도 있습니다. 이러한 프로토콜의 경우 NAT를 적용하고 보조 채널을 허용할 수 있도록 시스템이 심층 패킷 검사를 수행해야 합니다. 기본적으로 시스템에서는 몇 가지 일반적인 검사 엔진이 활성화되지만 네트워크에 따라 다른 엔진을 활성화하거나 기본 검사를 비활성화해야 할 수도 있습니다.

현재 활성화된 검사의 목록을 확인하려면 CLI 콘솔이나 SSH 세션에서 **show running-config policy-map** 명령을 사용합니다. 아래에는 검사 컨피그레이션을 변경하지 않은 시스템에서 이 명령을 실행하는 경우 표시되는 출력이 나와 있습니다. 이 출력의 끝에 나와 있는 **inspect** 명령의 목록에 활성화된 프로토콜 검사가 표시됩니다. 위의 명령은 inspection_default 트래픽 클래스(일반 프로토콜 및 해당하는 경우 검사되는 프로토콜의 포트 번호)에서 이러한 검사를 활성화합니다. 이 클래스는 service-policy 명령(출력에는 표시되지 않음)을 사용하여 모든 인터페이스에서 이러한 검사를 적용하는 global_policy 정책 맵의 일부분입니다. 예를 들어 ICMP 검사는 디바이스를 통과하는 모든 ICMP 트래픽에 대해 수행됩니다.

```
> show running-config policy-map
!
policy-map type inspect dns preset_dns_map
 parameters
  message-length maximum client auto
  message-length maximum 512
  no tcp-inspection
```

```
policy-map global_policy
 class inspection_default
  inspect dns preset_dns_map
  inspect ftp
  inspect h323 h225
  inspect h323 ras
  inspect rsh
  inspect rtsp
  inspect sqlnet
  inspect skinny
  inspect sunrpc
  inspect xdmcp
  inspect sip
  inspect netbios
  inspect tftp
  inspect ip-options
  inspect icmp
  inspect icmp error
!
```

참고 각 검사의 자세한 설명은 https://www.cisco.com/c/en/us/support/security/asa-firepower-services/products-installation-and-configuration-guides-list.html에서 제공되는 *Cisco ASA Series* 방화벽 컨피그레이션 가이드를 참조하십시오.

다음 절차에서는 전역적으로 적용된 이 기본 검사 클래스에서 검사를 활성화하거나 비활성화하는 방법을 보여줍니다. 이 방법을 설명하기 위해 예시에서는 다음을 수행합니다.

- PPTP(Point-to-Point Tunneling Protocol)를 활성화합니다. 이 프로토콜은 엔드포인트에 간의 지점 간 연결을 터널링하는 데 사용됩니다.
- SIP(Session Initiation Protocol)를 비활성화합니다. 일반적으로는 검사로 인해 네트워크에 문제가 발생하는 경우에만 SIP를 비활성화합니다. 그러나 SIP를 비활성화하는 경우에는 액세스 제어 정책이 SIP 트래픽(UDP/TCP 5060)과 동적으로 할당된 포트를 허용하며 SIP 연결에 대해 NAT 지원이 필요하지 않은지를 확인해야 합니다. 그리고 확인 결과에 따라 FlexConfig가 아닌 표준 페이지를 통해 액세스 제어 및 NAT 정책을 조정합니다.

시작하기 전에

적절한 계획을 세우면 FlexConfig를 효율적으로 사용할 수 있습니다. 이 예시에서는 동일 트래픽 클래스에서 서로 다르며 관련이 없는 두 검사를 변경합니다. 하지만 이러한 정책을 변경해야 하는 경우에는 독립적으로 변경하게 될 가능성이 높습니다.

따라서 이 예시의 각 검사에 대해 각기 별도의 FlexConfig 개체를 생성하는 것이 좋습니다. 이렇게 하면 다른 검사의 설정을 변경하지 않고도 한 검사의 설정을 쉽게 변경할 수 있으며, FlexConfig 개체를 수정할 필요도 없습니다.

프로시저

단계 **1** **Device(**디바이스**)** > **Advanced Configuration(**고급 컨피그레이션**)**에서 **View Configuration(**컨피그레이션 보기**)**을 클릭합니다.

단계 **2** Advanced Configuration(고급 컨피그레이션) 목차에서 **FlexConfig** > **FlexConfig Objects(FlexConfig** 개체**)**를 클릭합니다.

단계 **3** PPTP 검사를 활성화하는 개체를 생성합니다.

a) **+** 버튼을 클릭하여 새 개체를 생성합니다.

b) 개체의 이름을 입력합니다. 예를 들어 **Enable_PPTP_Global_Inspection**을 입력합니다.

c) **Template(**템플릿**)** 편집기에서 들여쓰기를 포함하여 다음 줄을 입력합니다.

```
policy-map global_policy
 class inspection_default
  inspect pptp
```

d) **Negate Template(**무효화 템플릿**)** 편집기에 이 컨피그레이션을 실행 취소하는 데 필요한 행을 입력합니다.

명령을 활성화하려면 상위 명령을 포함해 명령에 대해 정확한 하위 모드를 입력해야 하는 것과 마찬가지로, 무효화 템플릿에도 해당 명령을 포함해야 합니다.

무효화 템플릿은 이 개체를 정상적으로 구축한 후에 FlexConfig 정책에서 제거하는 경우에 적용되며, 실패한 구축 중에도 컨피그레이션을 이전 상태로 재설정하기 위해 적용됩니다.

그러므로 이 예시에서 무효화 템플릿은 다음과 같습니다.

```
policy-map global_policy
 class inspection_default
  no inspect pptp
```

개체는 다음과 같이 표시됩니다.

참고 inspection_default 클래스에는 다른 검사 명령이 활성화되어 있으므로 전체 클래스를 무효화해서는 안 됩니다. 마찬가지로 global_policy 정책 맵도 이러한 기타 검사가 포함되어 있으므로 정책 맵 역시 무효화하면 안 됩니다.

e) **OK(**확인**)**를 클릭하여 개체를 저장합니다.

단계 **4** SIP 검사를 비활성화하는 개체를 생성합니다.

a) + 버튼을 클릭하여 새 개체를 생성합니다.

b) 개체의 이름을 입력합니다. 예를 들어 **Disable_SIP_Global_Inspection**을 입력합니다.

c) **Template(**템플릿**)** 편집기에서 들여쓰기를 포함하여 다음 줄을 입력합니다.

```
policy-map global_policy
 class inspection_default
  no inspect sip
```

d) **Negate Template(**무효화 템플릿**)** 편집기에 이 컨피그레이션을 실행 취소하는 데 필요한 행을 입력합니다.

비활성화를 수행하는 "no" 명령에 대한 "negate" 명령이 기능을 활성화하는 명령입니다. 즉, "negate" 템플릿은 단지 기능을 비활성화하는 명령이 아니라 "positive" 템플릿에서 수행하는 모든 작업을 되돌리는 명령입니다. 이처럼 negate 템플릿의 핵심 기능은 변경 사항을 실행 취소하는 것입니다.

그러므로 이 예시에서 무효화 템플릿은 다음과 같습니다.

```
policy-map global_policy
 class inspection_default
  inspect sip
```

개체는 다음과 같이 표시됩니다.

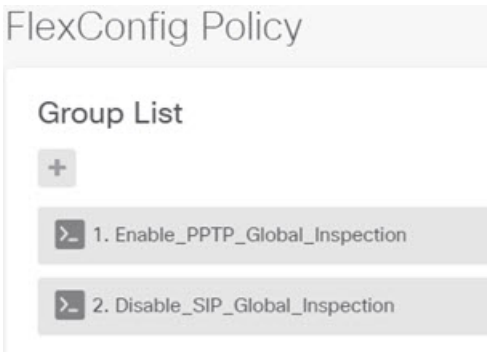
e) **OK(**확인**)**를 클릭하여 개체를 저장합니다.

단계 **5** FlexConfig 정책에 개체를 추가합니다.

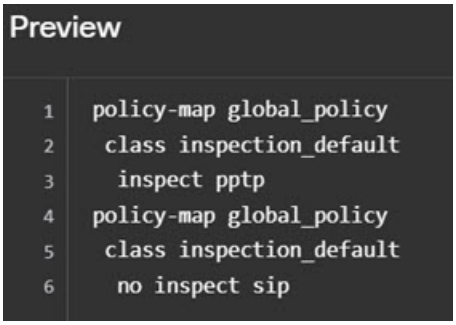
개체를 생성하는 것만으로는 충분하지 않습니다. 개체는 FlexConfig 정책에 추가하고 변경 사항을 저장해야 구축됩니다. 이렇게 하면 완료되지 않은 작업에서 구축 장애 발생 위험 없이 개체를 사용하여 실험을 하고 개체를 부분적으로 완성된 상태로 남겨 둘 수 있습니다. 그런 후에는 개체를 추가 및 제거만 하면 기능을 쉽게 켜거나 끌 수 있으며 매번 개체를 다시 생성할 필요가 없습니다.

a) 목차에서 **FlexConfig Policy(FlexConfig** 정책**)**를 클릭합니다.
b) Group List(그룹 목록)에서 +를 클릭합니다.
c) Enable_PPTP_Global_Inspection 및 Disable_SIP_Global_Inspection 개체를 선택하고 **OK(**확인**)**를 클릭합니다.

그룹 목록은 다음과 같이 표시됩니다.



템플릿의 명령으로 미리보기가 업데이트됩니다. 올바른 명령이 표시되는지 확인합니다.



d) **Save(**저장**)**를 클릭합니다.

이제 정책을 구축할 수 있습니다.

단계 **6** 변경 사항을 커밋합니다.

a) 웹 페이지의 오른쪽 상단에 있는 **Deploy Changes(**변경 사항 구축**)** 아이콘을 클릭합니다.

b) **Deploy Now(**지금 구축**)** 버튼을 클릭합니다.

구축이 완료될 때까지 기다리거나 **OK(**확인**)**를 클릭하고 나중에 작업 목록 또는 구축 기록을 확인할 수 있습니다.

단계 **7** CLI 콘솔 또는 SSH 세션에서 **show running-config policy-map** 명령을 사용하여 실행 중인 컨피그레이션에 정확한 변경 사항이 적용되었는지 확인합니다.

다음 출력에서는 **inspect pptp**(이)가 inspection_default 클래스 맵 아래에 추가되었으며 **inspect sip**(은)는 더 이상 클래스에 포함되어 있지 않다는 점에 유의하십시오. 즉, FlexConfig 개체에 정의된 변경 사항이 정상적으로 구축되었음을 확인할 수 있습니다.

```
> show running-config policy-map
!
policy-map type inspect dns preset_dns_map
 parameters
  message-length maximum client auto
  message-length maximum 512
```

```
  no tcp-inspection
policy-map global_policy
 class inspection_default
  inspect dns preset_dns_map
  inspect ftp
  inspect h323 h225
  inspect h323 ras
  inspect rsh
  inspect rtsp
  inspect sqlnet
  inspect skinny
  inspect sunrpc
  inspect xdmcp
  inspect netbios
  inspect tftp
  inspect ip-options
  inspect icmp
  inspect icmp error
  inspect pptp
!
```

FlexConfig 변경 사항을 실행 취소하는 방법

FlexConfig 개체에 정확한 무효화 템플릿을 입력하는 경우 해당 개체를 사용하여 적용한 변경 사항을 쉽게 제거할 수 있습니다. FlexConfig 정책에서 개체를 삭제하기만 하면 되며, 그러면 다음 구축에서 시스템이 무효화 템플릿을 사용하여 변경을 실행 취소합니다.

그러므로 변경을 실행 취소하기 위해 새 개체를 생성할 필요가 없습니다.

다음 예시에서는 전역 SIP 검사를 다시 활성화하는 방법을 보여 줍니다. 이 예시에서는 전역 기본 검사를 활성화/비활성화하는 방법, 27 페이지에서 설명하는 변경 사항을 되돌려 SIP 검사를 비활성화합니다.

시작하기 전에

FlexConfig 개체에 정확한 무효화 템플릿이 있는지 확인합니다. 정확한 무효화 템플릿이 없으면 무효화 템플릿이 정확해지도록 개체를 수정합니다.

프로시저

단계 **1** **Device(**디바이스**)** > **Advanced Configuration(**고급 컨피그레이션**)**에서 **View Configuration(**컨피그레이션 보기**)**을 클릭합니다.

단계 **2** Advanced Configuration(고급 컨피그레이션) 목차에서 **FlexConfig** > **FlexConfig Policy(FlexConfig** 정책**)**를 클릭합니다.

단계 **3** FlexConfig 정책에서 **Disable_SIP_Global_Inspection** 개체 항목 오른쪽의 **X**를 클릭하여 정책에서 개체를 삭제합니다.

개체의 명령이 미리보기에서 제거됩니다. 무효화 명령은 미리보기에 추가되지 않으며 백그라운드에서 실행됩니다.

단계 **4** **Save(**저장**)**를 클릭합니다.

단계 **5** 변경 사항을 커밋합니다.

a) 웹 페이지의 오른쪽 상단에 있는 **Deploy Changes(**변경 사항 구축**)** 아이콘을 클릭합니다.

b) **Deploy Now(**지금 구축**)** 버튼을 클릭합니다.

구축이 완료될 때까지 기다리거나 **OK(**확인**)**를 클릭하고 나중에 작업 목록 또는 구축 기록을 확인할 수 있습니다.

단계 **6** CLI 콘솔 또는 SSH 세션에서 **show running-config policy-map** 명령을 사용하여 실행 중인 컨피그레이션에 정확한 변경 사항이 적용되었는지 확인합니다.

다음 출력에서는 **inspect sip**(이)가 inspection_default 클래스 맵 아래에 추가되었습니다. 즉, FlexConfig 개체에 정의된 변경 사항이 정상적으로 구축되었음을 확인할 수 있습니다. (이 클래스에서 순서는 중요하지 않으므로 **inspect sip**(이)가 원래 위치가 아닌 끝에 있어도 상관없습니다.)

```
> show running-config policy-map
!
policy-map type inspect dns preset_dns_map
 parameters
  message-length maximum client auto
  message-length maximum 512
  no tcp-inspection
policy-map global_policy
 class inspection_default
  inspect dns preset_dns_map
  inspect ftp
  inspect h323 h225
  inspect h323 ras
  inspect rsh
  inspect rtsp
  inspect sqlnet
  inspect skinny
  inspect sunrpc
  inspect xdmcp
  inspect netbios
  inspect tftp
  inspect ip-options
  inspect icmp
  inspect icmp error
  inspect pptp
  inspect sip
!
```

고유한 트래픽 클래스에 대한 검사를 활성화하는 방법

이 예시에서는 특정 인터페이스의 두 엔드포인트 간 트래픽에 대해 PPTP 검사를 활성화합니다. 이렇게 하면 사이에 포인트 투 포인트 터널이 구성되어 있는 엔드포인트만 검사 대상으로 지정됩니다.

두 엔드포인트 간에 PPTP 검사를 활성화하는 데 필요한 CLI에는 다음 항목이 포함됩니다.

1. 소스와 대상이 엔드포인트 호스트의 IP 주소로 설정된 ACL.
2. 이 ACL을 참조하는 트래픽 클래스.
3. 트래픽 클래스를 포함하며 트래픽 클래스에 대해 PPTP 검사를 활성화하는 정책 맵.
4. 정책 맵을 원하는 인터페이스에 적용하는 서비스 정책. 이 단계에서 정책과 검사가 실제로 활성화됩니다.

참고 검사와 관련된 서비스 정책의 자세한 설명은 https://www.cisco.com/c/en/us/support/security/asa-firepower-services/products-installation-and-configuration-guides-list.html에서 제공되는 *Cisco ASA Series* 방화벽 컨피그레이션 가이드를 참조하십시오.

프로시저

단계 **1** **Device(**디바이스**)** > **Advanced Configuration(**고급 컨피그레이션**)**에서 **View Configuration(**컨피그레이션 보기**)**을 클릭합니다.

단계 **2** Advanced Configuration(고급 컨피그레이션) 목차에서 **FlexConfig** > **FlexConfig Objects(FlexConfig** 개체**)**를 클릭합니다.

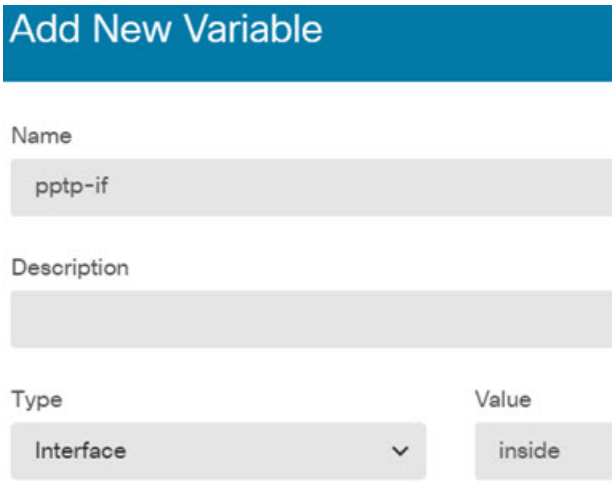
단계 **3** + 버튼을 클릭하여 새 개체를 생성합니다.

단계 **4** 개체의 이름을 입력합니다. 예를 들어 **Enable_PPTP_Inspection_on_Interface**를 입력합니다.

단계 **5** 내부 인터페이스에 대한 변수를 추가합니다.

a) Variables(변수) 목록 위의 +를 클릭합니다.
b) 변수의 이름을 **pptp-if**와 같이 입력합니다.
c) **Type(**유형**)**으로는 **Interface(**인터페이스**)**를 선택합니다.
d) **Value(**값**)**로는 **inside(**내부**)** 인터페이스를 선택합니다.

대화 상자는 다음과 같이 표시됩니다.



e) **Add(**추가**)**를 클릭합니다.

단계 **6** **Template(**템플릿**)** 편집기에서 들여쓰기를 포함하여 다음 줄을 입력합니다.

```
access-list MATCH_ACL permit ip host 192.168.1.55 host 198.51.100.1
class-map MATCH_CMAP
 match access-list MATCH_ACL
policy-map PPTP_POLICY
 class MATCH_CMAP
  inspect pptp
service-policy PPTP_POLICY interface {{pptp-if.name}}
```

변수를 사용하려면 이중 중괄호 사이에 변수 이름을 입력합니다. 또한 점 표기법을 사용하여 검색할 특성을 선택해야 합니다. 인터페이스를 정의하는 개체에는 여러 특성이 포함되어 있기 때문입니다. 인터페이스 이름은 "name" 특성에 포함되어 있으므로 **{{pptp-if.name}}**을 입력하면 변수에 할당된 인터페이스에 대한 name 특성의 값이 검색됩니다. PPTP 검사용 인터페이스를 변경해야 하는 경우 변수 정의에서 다른 인터페이스만 선택하면 됩니다.

단계 **7** **Negate Template(**무효화 템플릿**)** 편집기에 이 컨피그레이션을 실행 취소하는 데 필요한 행을 입력합니다.

이 예시의 경우 PPTP 검사 적용에만 사용되는 클래스 맵, 정책 맵 및 서비스 정책이 있다고 가정합니다. 따라서 무효화 템플릿에서 이러한 항목을 모두 제거할 것입니다.

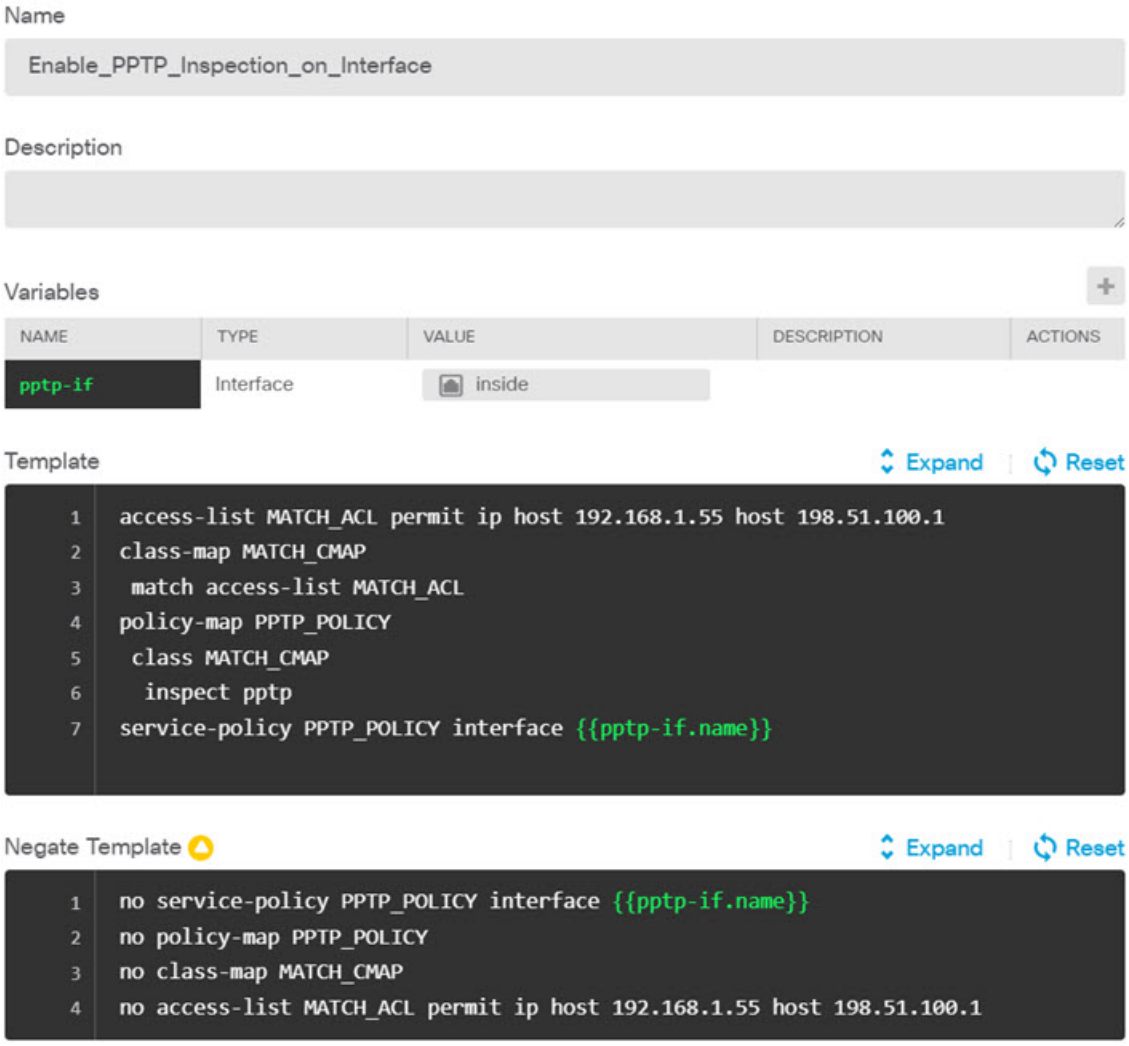
그러나 인터페이스의 기존 서비스 정책에 PPTP 검사를 실제로 추가하는 경우에는 정책 맵이나 서비스 정책을 무효화하지 않습니다. 정책 맵에서 클래스를 무효화하거나, 정책 맵에 포함된 클래스 내에서 검사만 끕니다. 무효화 템플릿 사용 시에 의도하지 않은 결과가 발생하지 않도록 다른 FlexConfig 개체에서 구현하는 내용을 명확하게 파악해야 합니다.

중첩된 항목을 삭제할 때는 해당 항목을 생성할 때와 반대 순서로 삭제해야 합니다. 따라서 먼저 서비스 정책부터 삭제하고 액세스 목록은 맨 끝에 삭제합니다. 이렇게 하지 않으면 사용 중인 개체 삭제를 시도하게 되므로 시스템에서 오류를 반환하며 개체를 삭제할 수 없습니다.

```
no service-policy PPTP_POLICY interface {{pptp-if.name}}
no policy-map PPTP_POLICY
no class-map MATCH_CMAP
```

```
no access-list MATCH_ACL permit ip host 192.168.1.55 host 198.51.100.1
```

개체는 다음과 같이 표시됩니다.



단계 **8** **OK(**확인**)**를 클릭하여 개체를 저장합니다.

단계 **9** FlexConfig 정책에 개체를 추가합니다.

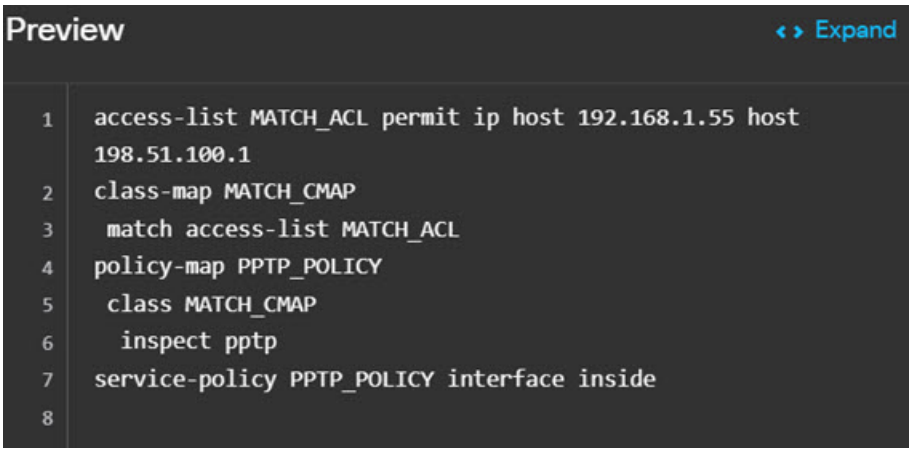
a) 목차에서 **FlexConfig Policy(FlexConfig** 정책**)**를 클릭합니다.

b) Group List(그룹 목록)에서 +를 클릭합니다.

c) **Enable_PPTP_Inspection_on_Interface** 개체를 선택하고 **OK(**확인**)**를 클릭합니다.

그룹 목록은 다음과 같이 표시됩니다.

템플릿의 명령으로 미리보기가 업데이트됩니다. 다음 그림에서와 같이 필요한 명령이 표시되는지 확인합니다. 인터페이스 변수는 미리보기에서 "inside"라는 이름으로 확인됩니다. 미리보기에서 정확하게 확인되지 않는 변수는 정확하게 구축되지 않으므로 변수를 자세히 확인하십시오. 미리보기에서 정확한 변수 변환이 표시될 때까지 FlexConfig 개체를 수정합니다.



d) **Save(**저장**)**를 클릭합니다.

이제 정책을 구축할 수 있습니다.

단계 **10** 변경 사항을 커밋합니다.

a) 웹 페이지의 오른쪽 상단에 있는 **Deploy Changes(**변경 사항 구축**)** 아이콘을 클릭합니다.

b) **Deploy Now(**지금 구축**)** 버튼을 클릭합니다.

구축이 완료될 때까지 기다리거나 **OK(**확인**)**를 클릭하고 나중에 작업 목록 또는 구축 기록을 확인할 수 있습니다.

단계 **11** CLI 콘솔 또는 SSH 세션에서 **show running-config** 명령의 변형을 사용하여 실행 중인 컨피그레이션에 정확한 변경 사항이 적용되었는지 확인합니다.

show running-config(을)를 입력하여 전체 CLI 컨피그레이션을 검사하거나 다음 명령을 사용하여 이 컨피그레이션의 각 부분을 확인할 수 있습니다.

- **show running-config access-list MATCH_ACL** ACL 확인.

- **show running-config class** 클래스 맵 확인. 이 명령을 실행하면 모든 클래스 맵이 표시됩니다.
- **show running-config policy-map PPTP_POLICY** 클래스 및 정책 맵 컨피그레이션 확인.
- **show running-config service-policy** 정책 맵이 인터페이스에 적용되었는지 확인. 이 명령을 실행하면 모든 서비스 정책이 표시됩니다.

이 명령의 시퀀스가 나와 있는 다음 출력을 통해 컨피그레이션이 정확하게 적용되었음을 확인할 수 있습니다.

```
> show running-config access-list MATCH_ACL
access-list MATCH_ACL extended permit ip host 192.168.1.55 host 198.51.100.1

> show running-config class
!
class-map MATCH_CMAP
 match access-list MATCH_ACL
class-map inspection_default
 match default-inspection-traffic
!

> show running-config policy-map PPTP_POLICY
!
policy-map PPTP_POLICY
 class MATCH_CMAP
  inspect pptp
!
> show running-config service-policy
service-policy global_policy global
service-policy PPTP_POLICY interface inside
```

고급 컨피그레이션