



FlexConfig 정책

다음 주제에서는 FlexConfig 정책을 구성하고 구축하는 방법을 설명합니다.

- FlexConfig 정책 개요, 1 페이지
- FlexConfig 정책에 대한 요구 사항 및 사전 요건, 22 페이지
- FlexConfig 가이드라인 및 제한 사항, 23 페이지
- FlexConfig 정책을 사용한 디바이스 구성 맞춤 설정, 23 페이지
- FlexConfig의 예시, 38 페이지

FlexConfig 정책 개요

FlexConfig 정책은 순서가 지정된 FlexConfig 개체 목록의 컨테이너입니다. 각 개체에는 일련의 Apache Velocity 스크립팅 언어 명령, ASA 소프트웨어 구성 명령 및 사용자 정의 변수가 포함됩니다. 각 FlexConfig 개체의 내용은 ASA 명령 시퀀스를 생성하는 프로그램이며, 이는 지정된 디바이스에 구축됩니다. 그런 다음 이 명령 시퀀스에 의해 threat defense 디바이스에서 관련 기능이 구성됩니다.

Threat Defense ASA 구성 명령을 사용하여 일부 기능(모든 기능이 아님)을 구현합니다. threat defense 구성 명령의 고유 집합은 없습니다. 대신, FlexConfig를 사용하면 management center 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 FlexConfig 정책을 사용하는 것이 좋습니다. 금지되지 않은 모든 명령을 구성할 수 있습니다. FlexConfig 정책을 통해 기능을 활성화하는 경우, 구성되어 있는 다른 기능과 함께 의도하지 않은 결과를 초래할 수 있습니다.

구성한 FlexConfig 정책과 관련된 지원을 받기 위해 Cisco TAC(Technical Assistance Center)에 문의할 수 있습니다. Cisco TAC(Technical Assistance Center)에서는 고객을 대신하여 맞춤형 구성을 설계하거나 작성하지 않습니다. Cisco에서는 올바른 작동이나 기타 Firepower System 기능과의 상호운용성에 대해 어떠한 보증도 명시하지 않습니다. FlexConfig 기능은 언제든지 사용이 중지될 수 있습니다. 완벽하게 보장되는 기능을 지원받으려면 management center의 지원을 기다려야 합니다. 의심스러운 경우에는 FlexConfig를 사용하지 마십시오.

FlexConfig 정책에 대한 추천 사용

FlexConfig에는 다음과 같이 권장되는 주요 사용 방법이 두 가지 있습니다.

- ASA에서 threat defense로 변환하는 중이며 management center에서 직접 지원하지 않는 호환 가능한 기능을 현재 사용 중이고 계속 사용해야 하는 경우, 이 경우, ASA에서 **show running-config** 명령을 사용하여 해당 기능에 대한 구성을 확인하고 FlexConfig 개체를 생성하여 해당 기능을 구현하십시오. 적절한 설정을 가져오려면 개체의 구축 설정을 실험합니다(한 번/항상 추가/추가). 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.

시스템에는 테스트된 구성을 나타내는 사전 정의된 FlexConfig 개체 집합이 포함되어 있습니다. 필요한 기능이 이러한 개체로 표시되지 않는 경우 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.

FlexConfig 개체의 CLI 명령

threat defense는 ASA 구성 명령을 사용하여 일부 기능을 구성합니다. 모든 ASA 기능이 threat defense에서 호환되는 것은 아니지만, threat defense에서는 작업 가능하나 management center 정책에서는 구성할 수 없는 기능도 일부 있습니다. FlexConfig 개체를 사용하여 이러한 기능을 구성하는 데 필요한 CLI를 지정할 수 있습니다.

FlexConfig를 사용하여 기능을 수동으로 구성하려는 경우, 적절한 구문에 따라 명령을 파악하고 구현해야 합니다. FlexConfig 정책은 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: <https://www.cisco.com/c/en/us/support/security/adaptive-security-appliance-asa-software/products-installation-and-configuration-guides-list.html>
- ASA 명령 참조에서는 명령 이름을 기준으로 정렬된 추가 정보를 제공합니다. 참조 위치: <https://www.cisco.com/c/en/us/support/security/adaptive-security-appliance-asa-software/products-command-reference-list.html>

다음 주제에서는 컨피그레이션 명령에 대해 자세히 설명합니다.

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 디바이스 관리 인터페이스에 대한 SSH 연결을 설정하고 다음 명령을 실행합니다.

- **show version system** Cisco Adaptive Security Appliance 소프트웨어 버전 번호를 찾습니다. (Secure Firewall Management Center CLI 도구를 통해 명령을 실행하는 경우 **system** 키워드를 생략합니다.)
- **show running-config** 현재 CLI 컨피그레이션을 확인합니다.
- **show running-config all** 현재 CLI 구성의 모든 기본 명령을 포함합니다.

다음 절차를 사용하여 management center 내에서 이 명령을 실행할 수도 있습니다.

프로시저

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

단계 2 FlexConfig 정책이 대상으로 하는 디바이스의 이름을 클릭합니다.

상태 테이블의 **Count**(개수) 열에서 열기/닫기 화살표를 클릭하여 모든 디바이스를 볼 수 있습니다.

단계 3 **Advanced Troubleshooting**(고급 문제 해결)을 선택합니다.

단계 4 **Threat Defense CLI**를 선택합니다.

단계 5 명령을 **show**로 선택하고 **version** 또는 다른 명령 중 하나를 파라미터로 입력합니다.

단계 6 **Execute**(실행)를 클릭합니다.

버전의 경우 Cisco Adaptive Security Appliance 소프트웨어 버전 번호의 출력을 검색합니다.

출력을 선택하고 Ctrl+C를 누른 다음 나중에 분석할 수 있도록 텍스트 파일에 붙여 넣을 수 있습니다.

금지된 CLI 명령

FlexConfig의 목적은 management center를 사용하여 threat defense 디바이스에서는 구성할 수 없으나 ASA 디바이스에서는 사용 가능한 기능을 구성하는 것입니다.

따라서 management center에서 동일한 역할을 하는 ASA 기능을 구성할 수 없습니다. 다음 표에는 이러한 금지된 명령 영역 중 일부가 나와 있습니다.

또한 일부 **clear** 명령은 관리 정책과 중복되기 때문에 금지되며, 관리 정책에 대한 구성의 일부를 삭제할 수 있습니다.

FlexConfig 개체 편집기를 사용하면 개체에 금지된 명령을 포함할 수 없습니다.

금지된 CLI 명령	설명
AAA	차단된 구성.
AAA-Server	차단된 구성.
Access-list	고급 ACL, 확장 ACL 및 표준 ACL이 차단됩니다. 이더 타입 ACL은 허용됩니다. 템플릿 내의 개체 관리자에 정의된 표준 및 확장 ACL 개체를 변수로 사용할 수 있습니다.
ARP 감시	차단된 구성.
As-path Object	차단된 구성.
배너	차단된 구성.
BGP	차단된 구성.
클럭	차단된 구성.
Community-list Object	차단된 구성.
카피	차단된 구성.
삭제	차단된 구성.
DHCP	차단된 구성.
Enable Password(활성화 비밀번호)	차단된 구성.
Erase	차단된 구성.
Fragment Setting	차단됨(fragment reassembly 제외).
Fsck	차단된 구성.
HTTP	차단된 구성.
ICMP	차단된 구성.
인터페이스	nameif, mode, shutdown, ip address 및 mac-address 명령만 차단됩니다.

금지된 CLI 명령	설명
멀티캐스트 라우팅	차단된 구성.
NAT	차단된 구성.
Network Object/Object-group	FlexConfig 개체에서의 네트워크 개체 생성은 차단되어 있지만, 템플릿 내부에서 개체 관리자에 정의되어 있는 네트워크 개체 및 그룹을 변수로 사용할 수는 있습니다.
NTP	차단된 구성.
OSPF/OSPFv3	차단된 구성.
pager	차단된 구성.
비밀번호 암호화	차단된 구성.
Policy-list Object	차단된 구성.
Prefix-list Object	차단된 구성.
재로드	reload 명령은 예약할 수 없습니다. 시스템에서는 reload 명령이 아닌 reboot 명령을 사용해 재시작합니다.
RIP	차단된 구성.
Route-Map Object	FlexConfig 개체에서의 경로 맵 개체 생성은 차단되어 있지만, 템플릿 내부에서 개체 관리자에 정의되어 있는 경로 맵 개체를 변수로 사용할 수는 있습니다.
Service Object/Object-group	FlexConfig 개체에서의 서비스 개체 생성은 차단되어 있지만, 템플릿 내부에서 개체 관리자에 정의되어 있는 포트 개체를 변수로 사용할 수는 있습니다.
SNMP	차단된 구성.
SSH	차단된 구성.
고정 경로	차단된 구성.
시스템 로그	차단된 구성.
시간 동기화	차단된 구성.
시간 초과	차단된 구성.
VPN	차단된 구성.

템플릿 스크립트

스크립팅 언어를 사용하여 FlexConfig 개체 내에서 처리를 제어할 수 있습니다. 스크립팅 언어 명령어는 루프, if/else 문 및 변수를 지원하는 Java 기반 스크립팅 언어인 Apache Velocity 1.3.1 템플릿 엔진에서 지원되는 명령의 하위 집합입니다.

스크립팅 언어 사용 방법을 알아보려면 <http://velocity.apache.org/engine/devel/developer-guide.html>에서 *Velocity Developer Guide*를 참조하십시오.

FlexConfig 변수

명령 또는 처리 지침의 일부가 정적 정보가 아닌 실행 정보에 따라 달라지는 경우 FlexConfig 개체에서 변수를 사용할 수 있습니다. 구축하는 동안 변수는 변수 유형에 따라 디바이스의 다른 구성에서 얻은 문자열로 대체됩니다.

- 정책 개체 변수는 management center에 정의된 개체에서 가져온 문자열로 대체됩니다.
- 시스템 변수는 디바이스 자체 또는 디바이스에 대해 구성된 정책에서 얻은 정보로 대체됩니다.
- 스크립팅 명령이 처리될 때 처리 변수에는 정책 개체 또는 시스템 변수의 내용이 로드됩니다. 예를 들어, 루프에서 정책 개체 또는 시스템 변수에서 하나의 값을 처리 변수로 반복적으로 로드한 다음 해당 처리 변수를 사용하여 명령 문자열을 구성하거나 다른 작업을 수행합니다. 이러한 처리 변수는 FlexConfig 개체의 변수 목록에 표시되지 않습니다 또한 FlexConfig 개체 편집기의 **Insert(삽입)** 메뉴를 사용하여 추가하지 마십시오.
- 비밀 키 변수는 FlexConfig 개체 내의 변수에 대해 정의된 단일 문자열로 대체됩니다.

변수는 @ 문자로 시작하는 비밀 키를 제외하고는 \$ 문자로 시작합니다. 예를 들어 \$ifname은 다음 명령에서 정책 개체 변수인 반면, @keyname은 비밀 키입니다.

```
interface $ifname
key @keyname
```



참고 정책 개체 또는 시스템 변수를 처음 삽입할 때 FlexConfig 개체 편집기의 **Insert(삽입)** 메뉴를 통해 정책 개체 또는 시스템 변수를 삽입해야 합니다. 이 작업은 변수를 FlexConfig 개체 편집기의 하단에 있는 **Variables(변수)** 목록에 추가합니다. 하지만 시스템 변수를 사용하는 경우에도 그 다음 사용 시 변수 문자열을 입력해야 합니다. 개체 또는 시스템 변수 할당이 없는 처리 변수를 추가하는 경우 **Insert(삽입)** 메뉴를 사용하지 마십시오. 비밀 키를 추가하는 경우 항상 **Insert(삽입)** 메뉴를 사용하십시오. 비밀 키 변수는 변수 목록에 표시되지 않습니다.

변수가 단일 문자열, 문자열 목록 또는 값 테이블로 확인되는지 여부는 변수에 할당한 정책 개체 또는 시스템 변수의 유형에 따라 다릅니다. (비밀 키는 항상 단일 문자열로 확인됩니다.) 변수를 올바르게 처리하기 위해 반환할 내용을 이해해야 합니다.

다음 주제는 다양한 유형의 변수와 이러한 변수를 처리하는 방법을 설명합니다.

변수 처리 방법

실행 시간에서 변수는 단일 문자열, 동일한 유형의 문자열 목록, 다른 유형의 문자열 목록 또는 이름이 지정된 값의 테이블로 확인될 수 있습니다. 또한 여러 값으로 확인되는 변수는 확실하거나 불확실한 길이일 수 있습니다. 값을 올바르게 처리하기 위해 반환할 내용을 이해해야 합니다.

다음은 기본 가능성입니다.

단일 값 변수

변수가 항상 단일 문자열로 확인되는 경우 FlexConfig 스크립트에서 변수의 수정 없이 직접 사용하십시오.

예를 들어 사전 정의된 텍스트 변수 `tcpMssBytes`는 항상 단일 값(숫자여야 함)으로 확인됩니다. 그런 다음 **Sysopt_basic** FlexConfig는 if/then/else 구조를 사용하여 또 다른 단일 값 텍스트 변수 `tcpMssMinimum`의 값을 기반으로 최대 세그먼트 크기를 설정합니다.

```
#if($tcpMssMinimum == "true")
  sysopt connection tcpmss minimum $tcpMssBytes
#else
  sysopt connection tcpmss $tcpMssBytes
#end
```

이 예에서는 FlexConfig 개체 편집기의 **Insert**(삽입) 메뉴를 사용하여 `$tcpMssBytes`의 첫 번째 사용을 추가하지만 변수를 `#else` 행에 직접 입력하기도 합니다.

비밀 키 변수는 단일 값 변수의 특수 유형입니다. 비밀 키의 경우 **Insert**(삽입) 메뉴를 사용하여 두 번째 상^o 및 후속 사용을 위해 변수를 추가합니다. 이러한 변수는 FlexConfig 개체의 변수 목록에 표시되지 않습니다



참고 네트워크 개체의 정책 개체 변수는 호스트 주소, 네트워크 주소 또는 주소 범위와 같은 단일 IP 주소 사양과 동일합니다. 하지만 이 경우 ASA 명령에 특정 주소 유형이 필요하기 때문에 예상 주소 유형을 알아야 합니다. 예를 들어 명령에 호스트 주소가 필요한 경우 네트워크 주소가 포함된 개체를 가리키는 네트워크 개체 변수를 사용하면 구축 중에 오류가 발생합니다.

다중 값 변수, 모든 값이 동일한 유형

여러 정책 개체 및 시스템 변수가 동일한 유형의 여러 값으로 확인됩니다. 예를 들어 네트워크 개체 그룹을 가리키는 개체 변수는 그룹 내의 IP 주소 목록으로 확인됩니다. 마찬가지로 시스템 변수 `$$SYS_FW_INTERFACE_NAME_LIST`는 인터페이스 이름 목록으로 확인됩니다.

동일한 유형의 여러 값에 대한 텍스트 개체를 생성할 수도 있습니다. 예를 들어 사전 정의된 텍스트 개체인 `enableInspectProtocolList`는 둘 이상의 프로토콜 이름을 포함할 수 있습니다.

동일한 유형의 항목 목록으로 확인되는 여러 값 변수는 종종 불확실한 길이입니다. 예를 들어 사용자는 언제든지 인터페이스를 구성 또는 구성 해제할 수 있으므로 이름이 지정된 디바이스의 인터페이스 수를 미리 알 수 없습니다.

따라서 일반적으로 루프를 사용하여 동일한 유형의 여러 값 변수를 처리합니다. 예를 들어 사전 정의된 FlexConfig **Default_Inspection_Protocol_Enable**은 #foreach 루프를 사용하여 enableInspectProtocolList 개체를 탐색하고 각 값을 처리합니다.

```
policy-map global_policy
  class inspection_default
    #foreach ( $protocol in $enableInspectProtocolList)
      inspect $protocol
    #end
```

이 예에서 스크립트는 각 값을 차례대로 \$protocol 변수에 할당합니다. 이 변수는 ASA **inspect** 명령에서 해당 프로토콜에 대한 검사 엔진을 활성화하는 데 사용됩니다. 이 경우 변수 이름으로 \$protocol을 입력하기만 하면 됩니다. 개체 또는 시스템 값을 변수에 할당하지 않으므로 **Insert**(삽입) 메뉴를 사용하여 추가하지 마십시오. 하지만 \$enableInspectProtocolList를 추가하려면 **Insert**(삽입) 메뉴를 사용해야 합니다.

시스템은 \$enableInspectProtocolList에 값이 남아 있지 않을 때까지 #foreach와 #end 사이의 코드를 반복합니다.

다중 값 변수, 값이 다른 유형

여러 개의 값 텍스트 개체를 만들 수 있지만 각 값은 다른 용도로 사용됩니다. 예를 들어 사전 정의된 **netflow_Destination** 텍스트 개체는 인터페이스 이름, 대상 IP 주소 및 UDP 포트 번호 순으로 3개의 값을 가져야 합니다.

이 방법으로 정의된 개체는 확실한 수의 값을 가져야 합니다. 그렇지 않으면 처리가 어려울 수 있습니다.

이러한 개체를 처리하려면 **get** 메서드를 사용합니다. 개체 이름 끝에 **.get(n)**을 입력하여 *n*을 개체의 인덱스를 대체합니다. 텍스트 개체가 1에서 시작하는 값을 나열하더라도 0에서 계산이 시작됩니다.

예를 들어 Netflow_Add_Destination 개체는 다음 줄을 사용하여 netflow_Destination의 3가지 값을 ASA **flow-export** 명령에 추가합니다.

```
flow-export destination $netflow_Destination.get(0) $netflow_Destination.get(1)
$netflow_Destination.get(2)
```

이 예에서는 FlexConfig 개체 편집기의 **Insert**(삽입) 메뉴를 사용하여 \$netflow_Destination의 첫 번째 사용을 추가한 다음 **.get(0)**을 추가합니다. 하지만 **\$netflow_Destination.get(1)** 및 **\$netflow_Destination.get(2)** 사양에 대한 변수를 직접 입력해야 합니다.

값 테이블을 확인하는 다중 값 변수

일부 시스템 변수는 값의 테이블을 반환합니다. 이러한 변수에는 이름에 MAP이 포함됩니다(예: \$SYS_FTD_ROUTED_INTF_MAP_LIST). 라우팅된 인터페이스 맵은 다음과 같은 데이터를 반환합니다(명확성을 위해 줄바꿈이 추가됨).

```
{[intf_hardwarare_id=GigabitEthernet0/0, intf_ipv6_eui64_addresses=[],
intf_ipv6_prefix_addresses=[], intf_subnet_mask_v4=255.255.255.0,
intf_ip_addr_v4=10.100.10.1, intf_ipv6_link_local_address=,
intf_logical_name=outside},
```

```
{intf_hardwarare_id=GigabitEthernet0/1, intf_ipv6_eui64_addresses=[],
intf_ipv6_prefix_addresses=[], intf_subnet_mask_v4=255.255.255.0,
intf_ip_addr_v4=10.100.11.1, intf_ipv6_link_local_address=,
intf_logical_name=inside},

{intf_hardwarare_id=GigabitEthernet0/2, intf_ipv6_eui64_addresses=[],
intf_ipv6_prefix_addresses=[], intf_subnet_mask_v4=, intf_ip_addr_v4=,
intf_ipv6_link_local_address=, intf_logical_name=},

{intf_hardwarare_id=Management0/0, intf_ipv6_eui64_addresses=[],
intf_ipv6_prefix_addresses=[], intf_subnet_mask_v4=, intf_ip_addr_v4=,
intf_ipv6_link_local_address=, intf_logical_name=diagnostic}}
```

위의 예에서는 4개의 인터페이스에 대한 정보가 반환됩니다. 각 인터페이스는 명명된 값의 테이블을 포함합니다. 예를 들어 `intf_hardwarare_id`는 인터페이스 하드웨어 이름 속성의 이름이고 `GigabitEthernet0/0`과 같은 문자열을 반환합니다.

이 유형의 변수는 일반적으로 길이가 일정하지 않으므로 루핑을 사용하여 값을 처리해야 합니다. 하지만 검색할 값을 나타내기 위해 변수 이름에 속성 이름을 추가해야 합니다.

예를 들어 IS-IS 구성에서는 인터페이스 구성 모드에서 논리적 이름이 있는 인터페이스에 `ASA isis` 명령을 추가해야 합니다. 하지만 인터페이스의 하드웨어 이름을 사용하여 해당 모드를 입력합니다. 따라서 논리적 이름이 있는 인터페이스를 확인한 다음 해당 하드웨어 이름을 사용하여 해당 인터페이스만 구성해야 합니다. 사전 정의된 `ISIS_Interface_Configuration FlexConfig`는 루프에 중첩된 `if/then` 구조를 사용하여 이 작업을 수행합니다. 다음 코드에서는 `#foreach` 스크립팅 명령이 각 인터페이스 맵을 `$intf` 변수로 로드한 다음 `#if` 문이 맵(`$intf.intf_logical_name`)의 `intf_logical_name` 값을 해제한다는 것을 알 수 있으며, 값이 `isisIntfList` 사전 정의의 텍스트 변수에 정의된 목록에 있으면 `intf_hardwarare_id` 값(`$intf.intf_hardwarare_id`)을 사용하여 인터페이스 명령을 입력합니다. ISIS를 구성할 인터페이스의 이름을 추가하려면 `isisIntfList` 변수를 편집해야 합니다.

```
#foreach ($intf in $SYS_FTD_ROUTED_INTF_MAP_LIST)
  #if ($isisIntfList.contains($intf.intf_logical_name))
    interface $intf.intf_hardwarare_id
      isis
      #if ($isisAddressFamily.contains("ipv6"))
        ipv6 router isis
      #end
    #end
  #end
#end
```

값이 디바이스에 대해 반환하는 사항을 확인하는 방법

변수가 반환할 값을 평가할 수 있는 간단한 방법은 주석이 있는 변수 목록을 처리하는 작업만 수행하는 간단한 FlexConfig 개체를 생성하는 것입니다. 그런 다음 FlexConfig 정책에 할당하고, 디바이스에 정책을 할당하고, 정책을 저장한 다음 해당 디바이스의 구성을 미리 볼 수 있습니다. 미리보기에 확인된 값이 표시됩니다. 미리보기 텍스트를 선택하고 Ctrl+C를 누른 다음 분석할 수 있도록 텍스트 파일에 출력을 붙여 넣을 수 있습니다.



참고 하지만 유효한 구성 명령이 포함되어 있지 않으므로 이 FlexConfig를 디바이스에 구축하지 마십시오. 구축 오류가 발생할 수 있습니다. 미리보기를 얻은 후 FlexConfig 정책에서 FlexConfig 개체를 삭제하고 정책을 저장합니다.

예를 들어 다음 FlexConfig 개체를 구성할 수 있습니다.

```
Following is a network object group variable for the
IPv4-Private-All-RFC1918 object:
```

```
$IPv4_Private_addresses
```

```
Following is the system variable SYS_FW_MANAGEMENT_IP:
```

```
$$SYS_FW_MANAGEMENT_IP
```

```
Following is the system variable SYS_FW_ENABLED_INSPECT_PROTOCOL_LIST:
```

```
$$SYS_FW_ENABLED_INSPECT_PROTOCOL_LIST
```

```
Following is the system variable SYS_FTD_ROUTED_INTF_MAP_LIST:
```

```
$$SYS_FTD_ROUTED_INTF_MAP_LIST
```

```
Following is the system variable SYS_FW_INTERFACE_NAME_LIST:
```

```
$$SYS_FW_INTERFACE_NAME_LIST
```

이 개체의 미리보기는 다음과 유사할 수 있습니다(명확성을 위해 줄바꿈이 추가됨).

```
###Flex-config Prepended CLI ###
```

```
###CLI generated from managed features ###
```

```
###Flex-config Appended CLI ###
```

```
Following is an network object group variable for the
IPv4-Private-All-RFC1918 object:
```

```
[10.0.0.0, 172.16.0.0, 192.168.0.0]
```

```
Following is the system variable SYS_FW_MANAGEMENT_IP:
```

```
192.168.0.171
```

```
Following is the system variable SYS_FW_ENABLED_INSPECT_PROTOCOL_LIST:
```

```
[dns, ftp, h323 h225, h323 ras, rsh, rtsp, sqlnet, skinny, sunrpc,
xdmcp, sip, netbios, tftp, icmp, icmp error, ip-options]
```

```
Following is the system variable SYS_FTD_ROUTED_INTF_MAP_LIST:
```

```
[{intf_hardwarare_id=GigabitEthernet0/0, intf_ipv6_eui64_addresses=[],
intf_ipv6_prefix_addresses=[], intf_subnet_mask_v4=255.255.255.0,
intf_ip_addr_v4=10.100.10.1, intf_ipv6_link_local_address=,
intf_logical_name=outside},
```

```
{intf_hardwarare_id=GigabitEthernet0/1, intf_ipv6_eui64_addresses=[],
intf_ipv6_prefix_addresses=[], intf_subnet_mask_v4=255.255.255.0,
```

```

intf_ip_addr_v4=10.100.11.1, intf_ipv6_link_local_address=,
intf_logical_name=inside},

{intf_hardwarare_id=GigabitEthernet0/2, intf_ipv6_eui64_addresses=[],
intf_ipv6_prefix_addresses=[], intf_subnet_mask_v4=, intf_ip_addr_v4=,
intf_ipv6_link_local_address=, intf_logical_name=},

{intf_hardwarare_id=Management0/0, intf_ipv6_eui64_addresses=[],
intf_ipv6_prefix_addresses=[], intf_subnet_mask_v4=, intf_ip_addr_v4=,
intf_ipv6_link_local_address=, intf_logical_name=diagnostic}}

Following is the system variable SYS_FW_INTERFACE_NAME_LIST:

[outside, inside, diagnostic]

```

FlexConfig 정책 개체 변수

정책 개체 변수는 개체 관리자에 구성된 특정 정책 개체와 연결됩니다. FlexConfig 개체에 정책 개체 변수를 삽입하면 변수에 이름을 지정하고 연결된 개체를 선택합니다.

변수에 연결된 개체와 정확히 동일한 이름을 지정할 수 있지만 변수 자체는 연결된 개체와 동일한 것이 아닙니다. FlexConfig 개체 편집기의 **Insert(삽입) > Insert Policy Object(정책 개체 삽입) > Object Type(개체 유형)** 메뉴를 사용하여 FlexConfig의 스크립트에 처음으로 변수를 추가하여 개체와의 연결을 설정해야 합니다. \$ 기호 앞에 개체의 이름을 입력하지만 하면 정책 개체 변수가 생성되지 않습니다.

다음 유형의 개체를 가리키는 변수를 생성할 수 있습니다. 각 변수에 대해 올바른 유형의 개체를 생성했는지 확인하십시오. 개체를 생성하려면 **Objects(개체) > Object Management(개체 관리)** 페이지로 이동합니다.

- **Text Objects(텍스트 개체)** - 텍스트 문자열의 경우 IP 주소, 숫자 및 인터페이스 또는 영역 이름과 같은 기타 자유 형식 텍스트를 포함할 수 있습니다. 목차에서 **FlexConfig > Text Object(텍스트 개체)**를 선택한 다음 **Add Text Object(텍스트 개체 추가)**를 클릭합니다. 단일 값 또는 다중 값을 포함하도록 이 개체를 구성할 수 있습니다. 이러한 개체는 매우 유연하며 FlexConfig 개체 내에서 사용하도록 특수하게 생성됩니다. 자세한 내용은 [FlexConfig 텍스트 개체 설정, 29 페이지](#)의 내용을 참조하십시오.
- **Network(네트워크)** - IP 주소 목적입니다. 네트워크 개체 또는 그룹을 사용할 수 있습니다. 목차에서 **Network(네트워크)**를 선택한 다음 **Add Network(네트워크 추가) > Add Object(개체 추가)** 또는 **Add Group(그룹 추가)**를 선택합니다. 그룹 개체를 사용하는 경우 변수는 그룹 내의 각 IP 주소 사양 목록을 반환합니다. 주소는 개체 내용에 따라 호스트, 네트워크 또는 주소 범위가 될 수 있습니다. [네트워크](#)의 내용을 참조하십시오.
- **Security Zones(보안 영역)** - 보안 영역 또는 인터페이스 그룹 내의 인터페이스 목적입니다. 목차에서 **Interface(인터페이스)**를 선택한 다음 **Add(추가) > Security Zone(보안 영역)** 또는 **Interface Group(인터페이스 그룹)**을 선택합니다. 보안 영역 변수는 구성 중인 디바이스에 대한 해당 영역 또는 그룹 내의 인터페이스 목록을 반환합니다. [Interface\(인터페이스\)](#)의 내용을 참조하십시오.
- **Standard ACL Object(표준 ACL 개체)** - 표준 액세스 제어 목록 목적입니다. 표준 ACL 변수는 표준 ACL 개체의 이름을 반환합니다. 목차에서 **Access List(액세스 목록) > Standard(표준)**를 선택

택한 다음 **Add Standard Access List Object**(표준 액세스 목록 개체 추가)를 클릭합니다. [액세스 목록](#)의 내용을 참조하십시오.

- **Extended ACL Object**(확장된 ACL 개체) - 확장된 액세스 제어 목록 목적입니다. 확장된 ACL 변수는 확장된 ACL 개체의 이름을 반환합니다. 목차에서 **Access List**(액세스 목록) > **Extended**(확장됨)를 선택한 다음 **Add Extended Access List Object**(확장된 액세스 목록 개체 추가)를 클릭합니다. [액세스 목록](#)의 내용을 참조하십시오.
- **Route Map**(경로 맵) - 경로 맵 개체 목적입니다. 경로 맵 변수는 경로 맵 개체의 이름을 반환합니다. 목차에서 **Route Map**(경로 맵)을 선택한 다음 **Add Route Map**(경로 맵 추가)를 클릭합니다. [경로 맵](#)의 내용을 참조하십시오.

FlexConfig 시스템 변수

시스템 변수는 이 대 한 구성 된 정책 또는 디바이스 자체에서 얻은 정보개로 바뀝니다.

FlexConfig 개체 편집기의 **Insert**(삽입) > **Insert System Variable**(시스템 변수 삽입) > **Variable Name**(변수 이름) 메뉴를 사용하여 FlexConfig의 스크립트에 처음으로 변수를 추가하여 시스템 변수와의 연결을 설정해야 합니다. \$ 기호가 앞에 오는 시스템 변수의 이름을 입력하는 것만으로는 FlexConfig 개체의 컨텍스트 내에서 시스템 변수가 생성되지 않습니다.

다음 테이블에서는 사용 가능한 시스템 변수를 설명합니다. 변수를 사용하기 전에 일반적으로 변수에 대해 반환되는 항목을 검사하십시오. [값이 디바이스에 대해 반환하는 사항을 확인하는 방법, 9 페이지](#) 섹션을 참조하십시오.

이름	설명
SYS_FW_OS_MODE	디바이스의 운영 체제 모드입니다. 가능한 값은 ROUTED 또는 TRANSPARENT입니다.
SYS_FW_OS_MULTIPLICITY	디바이스가 단일 또는 다중 컨텍스트 모드로 실행 중인지 여부입니다. 가능한 값은 SINGLE, MULTI 또는 NOT_APPLICABLE입니다.
SYS_FW_MANAGEMENT_IP	디바이스의 관리되는 IP 주소
SYS_FW_HOST_NAME	디바이스 호스트네임
SYS_FTD_INTF_POLICY_MAP	인터페이스 이름을 키로, 정책 맵을 값으로 하는 맵입니다. 이 변수는 디바이스에 정의된 인터페이스 기반 서비스 정책이 없는 경우 아무 것도 반환하지 않습니다.
SYS_FW_ENABLED_INSPECT_PROTOCOL_LIST	검사가 활성화되는 프로토콜 목록입니다.
SYS_FTD_ROUTED_INTF_MAP_LIST	디바이스에서 라우팅된 인터페이스 맵 목록입니다. 각 맵에는 라우팅된 인터페이스 구성과 관련된 명명된 값 집합이 포함되어 있습니다.
SYS_FTD_SWITCHED_INTF_MAP_LIST	디바이스에서 전환된 인터페이스 맵 목록입니다. 각 맵에는 전환된 인터페이스 구성과 관련된 명명된 값 집합이 포함되어 있습니다.

이름	설명
SYS_FTD_INLINE_INTF_MAP_LIST	디바이스의 인라인 인터페이스 맵 목록입니다. 각 맵에는 인라인 설정 인터페이스 구성과 관련된 명명된 값 집합이 포함되어 있습니다.
SYS_FTD_PASSIVE_INTF_MAP_LIST	디바이스의 수동 인터페이스 맵 목록입니다. 각 맵에는 수동 인터페이스 구성과 관련된 명명된 값 집합이 포함되어 있습니다.
SYS_FTD_INTF_BVI_MAP_LIST	디바이스의 Bridge Virtual Interface 맵 목록입니다. 각 맵에는 BVI 구성과 관련된 명명된 값 집합이 포함되어 있습니다.
SYS_FW_INTERFACE_HARDWARE_ID_LIST	디바이스의 인터페이스에 대한 하드웨어 이름 목록(예: GigabitEthernet0/0)입니다.
SYS_FW_INTERFACE_NAME_LIST	디바이스의 인터페이스에 대한 논리적 이름 목록입니다(예: inside).
SYS_FW_INLINE_INTERFACE_NAME_LIST	수동 또는 ERSPAN 수동으로 구성된 인터페이스의 논리적 이름 목록입니다.
SYS_FW_NON_INLINE_INTERFACE_NAME_LIST	모든 라우팅된 인터페이스와 같이 인라인 집합에 속하지 않은 인터페이스의 논리적 이름 목록입니다.

사전 정의된 FlexConfig 개체

사전 정의된 FlexConfig 개체는 선택된 기능에 대해 테스트된 구성을 제공합니다. 이러한 기능을 구성해야 하는 경우 이 개체를 사용하십시오. 그렇지 않은 경우 management center를 사용하여 구성할 수 없습니다.

다음 테이블에는 사용 가능한 개체가 나와 있습니다. 연결된 텍스트 개체를 적어 둡니다. 사전 정의된 FlexConfig 개체의 동작을 사용자 정의하려면 이러한 텍스트 개체를 편집해야 합니다. 텍스트 개체를 사용하면 네트워크 및 디바이스에 필요한 IP 주소 및 기타 속성을 사용하여 구성을 사용자 정의할 수 있습니다.

사전 정의된 FlexConfig 개체를 수정해야 할 경우 개체를 복사하고 복사본을 변경한 다음 새 이름으로 저장합니다. 사전 정의된 FlexConfig 개체를 직접 편집할 수 없습니다.

FlexConfig를 사용하여 다른 ASA 기반 기능을 구성할 수도 있지만 이러한 기능의 구성은 테스트되지 않았습니다. ASA 기능이 management center 정책에서 구성할 수 있는 기능과 중복되는 경우 FlexConfig를 통해 구성하지 마십시오.

예를 들어 Snort 검사에는 HTTP 프로토콜이 포함되어 있으므로 ASA 스타일 HTTP 검사를 활성화하지 마십시오. (실제로 http를 enableInspectProtocolList 개체에 추가할 수 없습니다. 이 경우 디바이스를 잘못 구성하는 것이 방지됩니다.) 대신, HTTP 검사 요구 사항을 구현하기 위해 필요에 따라 애플리케이션 또는 URL 필터링을 수행하도록 액세스 제어 정책을 구성합니다.

표 1: 사전 정의된 FlexConfig 개체

FlexConfig 개체 이름	설명	연결된 텍스트 개체
Default_Inspection_Protocol_Disable	global_policy 기본 정책 맵에서 프로토콜을 비활성화합니다.	disableInspectProtocolList
Default_Inspection_Protocol_Enable	global_policy 기본 정책 맵에서 프로토콜을 활성화합니다.	enableInspectProtocolList
DHCPv6_Prefix_Delegation_Configure	IPv6 접두사 위임을 위해 외부 인터페이스(Prefix Delegation 클라이언트) 및 내부 인터페이스(위임된 접두사의 수신자)를 하나씩 구성할 수 있습니다. 이 템플릿을 사용하려면 복사하여 변수를 수정합니다.	pdoutside, pdinside 또한 시스템 변수 SYS_FTD_ROUTED_INTF_MAP_LIST 를 사용합니다.
DHCPv6_Prefix_Delegation_UnConfigure	DHCPv6 접두사 위임 구성을 제거합니다.	pdoutside, pdinside 또한 시스템 변수 SYS_FTD_ROUTED_INTF_MAP_LIST 를 사용합니다.
Inspect_IPv6_Configure	global_policy 정책 맵에서 IPv6 검사를 구성하고 IPv6 헤더 내용을 기반으로 트래픽을 기록 및 삭제합니다.	IPv6RoutingHeaderDropLogList, IPv6RoutingHeaderLogList, IPv6RoutingHeaderDropList.
Inspect_IPv6_UnConfigure	IPv6 검사를 지우고 비활성화합니다.	—
ISIS_Configure	IS-IS 라우팅에 대한 전역 파라미터를 구성합니다.	isIsNet, isIsAddressFamily, isIsType
ISIS_Interface_Configuration	인터페이스 레벨 IS-IS 구성.	isIsAddressFamily, IsIsIntfList 또한 시스템 변수 SYS_FTD_ROUTED_INTF_MAP_LIST 를 사용합니다.
ISIS_Unconfigure	디바이스의 IS-IS 라우터 구성을 지웁니다.	—
ISIS_Unconfigure_All	디바이스 인터페이스의 라우터 할당을 포함하여 디바이스에서 IS-IS 라우터 구성을 지웁니다.	—
Netflow_Add_Destination	Netflow 내보내기 대상을 만들고 구성합니다.	Netflow_Destinations, netflow_Event_Types
Netflow_Clear_Parameters	Netflow 내보내기 전역 기본 설정을 복원합니다.	—

FlexConfig 개체 이름	설명	연결된 텍스트 개체
Netflow_Delete_Destination	Netflow 내보내기 대상을 삭제 합니다.	Netflow_Destinations, netflow_Event_Types
Netflow_Set_Parameters	Netflow 내보내기 전역 파라미터를 설정 합니다.	netflow_Parameters
NGFW_TCP_NORMALIZATION	기본 TCP 정규화 구성을 수정합니다.	—
Policy_Based_Routing	이 예제 구성을 사용하려면 복사하고, 인터페이스 이름을 수정하고, r-map-object 텍스트 개체를 사용하여 개체 관리자에서 경로 맵 개체를 식별합니다.	—
Policy_Based_Routing_Clear	디바이스에서 정책 기반 라우팅 구성을 지웁니다.	—
Sysopt_AAA_radius	RADIUS 계정 응답에서 인증 키를 무시 합니다.	—
Sysopt_AAA_radius_negate	Sysopt_AAA_radius 구성을 무효화합니다.	—
Sysopt_basic	sysopt 대기 시간, TCP 패킷의 최대 세그먼트 크기 및 자세한 트래픽 통계를 구성합니다.	tcpMssMinimum, tcpMssBytes
Sysopt_basic_negate	sysopt_basic 세부 트래픽 통계, 대기 시간 및 TCP 최대 세그먼트 크기를 지웁니다.	—
Sysopt_clear_all	디바이스에서 모든 sysopt 구성을 지웁니다.	—
Sysopt_noproxyarp	noproxy-arp CLI를 구성합니다.	시스템 변수 SYS_FW_NON_INLINE_INTF_NAME_LIST를 사용합니다.
Sysopt_noproxyarp_negate	Sysopt_noproxyarp 구성을 지웁니다.	시스템 변수 SYS_FW_NON_INLINE_INTF_NAME_LIST를 사용합니다.
Sysopt_Preserve_Vpn_Flow	sysopt preserve VPN 흐름을 구성합니다.	—
Sysopt_Preserve_Vpn_Flow_negate	Sysopt_Preserve_Vpn_Flow 구성을 지웁니다.	—
Sysopt_Reclassify_Vpn	sysopt reclassify vpn을 구성합니다.	—

FlexConfig 개체 이름	설명	연결된 텍스트 개체
Sysopt_Reclassify_Vpn_Negate	sysopt reclassify vpn을 무효화합니다.	—
Threat_Detection_Clear	위협 탐지 TCP 가로채기 구성을 지웁니다.	—
Threat_Detection_Configure	TCP 가로채기에 의해 가로채기된 공격에 대한 위협 탐지 통계를 구성합니다.	threat_detection_statistics
Wccp_Configure	이 템플릿은 WCCP를 구성하는 예제를 제공합니다.	isServiceIdentifier, serviceIdentifier, wccpPassword
Wccp_Configure_Clear	WCCP 구성을 지웁니다.	—

지원 중단된 FlexConfig 개체

다음 표에는 이제 GUI에서 기본적으로 구성할 수 있는 기능을 구성하는 개체가 나와 있습니다. 가능한 한 빨리 이러한 개체의 사용을 중단하십시오.

표 2: 지원 중단된 사전 정의된 FlexConfig 개체

사용 중단 버전	FlexConfig 개체	설명	지금 구성
6.3	Default_DNS_Configure	데이터 인터페이스에서 정규화된 도메인 이름을 확인할 때 사용할 수 있는 DNS 서버를 정의하는 기본 DNS 그룹을 구성합니다. 연결된 텍스트 개체: defaultDNSNameServerList, defaultDNSParameters	플랫폼 설정.
6.3	DNS_Configure	기본값이 아닌 DNS 서버 그룹에 DNS 서버를 구성합니다. 그룹의 이름을 변경하려면 개체를 복사합니다.	개체 관리자의 DNS 서버 그룹.
6.3	DNS_UnConfigure	Default_DNS_Configure 및 DNS_Configure가 수행하는 DNS 서버 구성을 제거합니다. DNS_Configure를 변경한 경우 개체를 복사하여 DNS 서버 그룹 이름을 변경합니다.	개체 관리자의 DNS 서버 그룹.

사용 중단 버전	FlexConfig 개체	설명	지금 구성
7.2	Eigrp_Configure	EIGRP 라우팅 next-hop, auto-summary, router-id, eigrp-stub 을 구성합니다. 연결된 텍스트 개체: eigrpAS, eigrpNetworks, eigrpDisableAutoSummary, eigrpRouterId, eigrpStubReceiveOnly, eigrpStubRedistributed, eigrpStubConnected, eigrpStubStatic, eigrpStubSummary	모든 EIGRP 개체에 대해서는 를 참조하십시오. EIGRP 시스템은 업그레이드 후 구축을 허용하지만 EIGRP 구성을 다시 실행하라는 경고도 합니다. 이 프로세스를 지원하기 위해 Cisco에서는 명령줄 마이그레이션 툴을 제공합니다.
7.2	Eigrp_Interface_Configure	EIGRP 인터페이스 인증 모드, 인증 키, hello 간격, 보류 시간, split horizon 을 구성합니다. 연결된 텍스트 개체: eigrpIntfList, eigrpAS, eigrpAuthKey, eigrpAuthKeyId, eigrpHelloInterval, eigrpHoldTime, eigrpDisableSplitHorizon 또한 시스템 변수 SYS_FTD_ROUTED_INTF_MAP_LIST 를 사용합니다.	
7.2	Eigrp_Unconfigure	- 5) 디바이스의 자율 시스템에 대한 EIGRP 구성을 지웁니다.	
7.2	Eigrp_Unconfigure_all	모든 EIGRP 구성을 지웁니다.	
6.3	TCP_Embryonic_Conn_Limit	SYN 플러드 서비스 거부(DoS) 공격으로부터 보호하기 위해 원시 연결 제한을 구성합니다. 연결된 텍스트 개체: tcp_conn_misc, tcp_conn_limit	서비스 정책.
6.3	TCP_Embryonic_Conn_Timeout	SYN 플러드 서비스 거부(DoS) 공격으로부터 보호하기 위해 원시 연결 시간 초과를 구성합니다. 연결된 텍스트 개체: tcp_conn_misc, tcp_conn_timeout	서비스 정책.

사용 중단 버전	FlexConfig 개체	설명	지금 구성
7.2	VxLAN_Clear_Nve	VxLAN_Configure_Port_And_Nve 디바이스에서 사용 하는 경우 구성 된 NVE 1을 제거 합니다.	모든 VxLAN 개체에 대해서는 VXLAN 인터페이스 구성 의 내용을 참조하십시오. 이전 버전에서 FlexConfig를 사용하여 VXLAN 인터페이스를 구성한 경우 계속 작동합니다. 실제로 이 경우 FlexConfig가 우선적으로 적용됩니다. 웹 인터페이스에서 VXLAN 구성을 다시 실행하는 경우 FlexConfig 설정을 제거합니다.
7.2	VxLAN_Clear_Nve_Only	구축될 때 인터페이스에 구성된 NVE를 지웁니다.	
7.2	VxLAN_Configure_Port_And_Nve	VLAN 포트 및 NVE 1을 구성합니다. 연결된 텍스트 개체: vxlan_Port_And_Nve	
7.2	VxLAN_Make_Nve_Only	NVE 전용 인터페이스를 설정합니다. 연결된 텍스트 개체: vxlan_Nve_Only 또한 시스템 변수 SYS_FTD_ROUTED_MAP_LIST 및 SYS_FTD_SWITCHED_INTF_MAP_LIST 를 사용합니다.	
7.2	VxLAN_Make_Vni	VNI 인터페이스를 만듭니다. 구축 후에는 VNI 인터페이스를 제대로 검색할 수 있도록 디바이스를 등록 취소하고 다시 등록해야 합니다. 연결된 텍스트 개체: vxlan_Vni	

사전 정의된 텍스트 개체

여러 개의 사전 정의된 텍스트 개체가 있습니다. 이 개체는 사전 정의된 FlexConfig 개체에 쓰이는 변수와 관련 있습니다. 대부분의 경우 관련 FlexConfig 개체를 사용하는 경우 이 개체를 편집하여 값을

추가해야 합니다. 그렇지 않으면 구축 중에 오류가 표시됩니다. 이러한 옵션 중 일부는 기본값을 포함하고 있으나 어떤 옵션은 비어 있습니다.

텍스트 개체 편집에 대한 내용은 [FlexConfig 텍스트 개체 설정, 29 페이지](#) 섹션을 참조하십시오.

이름	설명	관련 FlexConfig 개체
defaultDNSNameServerList (사용되지 않음)	기본 DNS 그룹에서 구성할 DNS 서버 IP 주소입니다. 버전 6.3부터 Firepower Threat Defense 플랫폼 설정 정책에서 데이터 인터페이스에 대한 DNS를 구성합니다.	Default_DNS_Configure
defaultDNSParameters (사용되지 않음)	기본 DNS 서버 그룹에 대한 DNS 동작을 제어하는 파라미터입니다. 개체에는 재시도, 시간 초과, <code>expire-entry-timer</code> , <code>poll-timer</code> , <code>domain-name</code> 에 대한 개별 항목이 순서대로 포함됩니다. 버전 6.3부터 Firepower Threat Defense 플랫폼 설정 정책에서 데이터 인터페이스에 대한 DNS를 구성합니다.	Default_DNS_Configure
disableInspectProtocolList	기본 정책 맵에서 프로토콜을 비활성화합니다(<code>global_policy</code>).	Disable_Default_Inspection_Protocol
dnsNameServerList	사용자 정의 DNS 그룹에서 구성할 DNS 서버 IP 주소입니다.	DNS_Configure
dnsParameters	기본값 이외의 DNS 서버 그룹에 대한 DNS 동작을 제어하는 파라미터입니다. 개체에는 재시도, 시간 초과, <code>domain-name</code> , <code>name-server-interface</code> 에 대한 개별 항목이 순서대로 포함됩니다.	DNS_Configure
enableInspectProtocolList	기본 정책 맵에서 프로토콜을 활성화합니다(<code>global_policy</code>). 해당 프로토콜 검사가 Snort 검사와 충돌하는 프로토콜을 추가할 수 없습니다.	Enable_Default_Inspection_Protocol
IPv6RoutingHeaderDropList	허용하지 않으려는 IPv6 라우팅 헤더 유형의 목록입니다. IPv6 검사는 중단을 기록하지 않고 이러한 헤더가 포함된 패킷을 삭제합니다.	Inspect_IPv6_Configure

이름	설명	관련 FlexConfig 개체
IPv6RoutingHeaderDropLogList	허용 및 기록하지 않으려는 IPv6 라우팅 헤더 유형의 목록입니다. IPv6 검사는 이러한 헤더가 포함된 패킷을 삭제하고 중단에 대한 시스템 로그 메시지를 보냅니다.	Inspect_IPv6_Configure
IPv6RoutingHeaderLogList	허용하지만 기록하지 않으려는 IPv6 라우팅 헤더 유형의 목록입니다. IPv6 검사는 이러한 헤더를 포함하는 패킷을 허용하지만 헤더의 존재 여부에 대한 시스템 로그 메시지를 보냅니다.	Inspect_IPv6_Configure
isIsAddressFamily	IPv4 또는 IPv6 주소군입니다.	ISIS_Configure ISIS_Interface_Configuration
IsIsIntfList	논리적 인터페이스 이름 목록입니다.	ISIS_Interface_Configuration
isIsSType	IS 유형(level-1, level-2-only 또는 level-1-2)입니다.	ISIS_Configure
isIsNet	네트워크 엔티티입니다.	ISIS_Configure
isServiceIdentifier	false이면 표준 web-cache 서비스 식별자를 사용합니다.	Wccp_Configure
netflow_Destination	단일 Netflow 내보내기 대상의 인터페이스, 대상, UDP 포트 번호를 정의합니다.	Netflow_Add_Destination
netflow_Event_Types	대상에 대해 내보낼 이벤트 유형을 all , flow-create , flow-defined , flow-teardown , flow-update 의 하위 집합으로 정의합니다.	Netflow_Add_Destination
netflow_Parameters	Netflow 내보내기 전역 설정(활성 새로고침 간격(흐름 업데이트 이벤트 사이의 시간(분)), 지연(초 단위의 흐름 생성 지연, 기본값 0 = 명령이 나타나지 않음) 및 템플릿 시간 초과 비율(분)을 제공합니다.	Netflow_Set_Parameters
PrefixDelegationInside	DHCPv6 접두사 위임을 위한 내부 인터페이스를 구성합니다. 개체에는 인터페이스 이름, 프리픽스 길이가 포함된 IPv6 접미사 및 접두사 풀 이름 순서대로 여러 항목이 포함됩니다.	없음. 하지만 DHCPv6_Prefix_Delegation_Configure 사본과 함께 사용할 수 있습니다.

이름	설명	관련 FlexConfig 개체
PrefixDelegationOutside	외부 DHCPv6 접두사 위임 클라이언트를 구성합니다. 개체에는 인터페이스 이름, 프리픽스 길이 순서대로 여러 항목이 포함됩니다.	없음. 그러나 DHCPv6_Prefix_Delegation_Configure 사본과 함께 사용할 수 있습니다.
serviceIdentifier	동적 WCCP 서비스 식별자 번호입니다.	Wccp_Configure
tcp_conn_limit (사용되지 않음)	TCP 원시 연결 제한을 구성하는 데 사용되는 파라미터입니다. 버전 6.3부터 Firepower Threat Defense Service 정책에서 이러한 기능을 구성합니다. 해당 정책은 디바이스에 할당된 액세스 제어 정책의 Advanced(고급) 탭에서 찾을 수 있습니다.	TCP_Embryonic_Conn_Limit
tcp_conn_misc (사용되지 않음)	TCP 원시 연결 설정을 구성하는 데 사용되는 파라미터입니다. 버전 6.3부터 Firepower Threat Defense Service 정책에서 이러한 기능을 구성합니다. 해당 정책은 디바이스에 할당된 액세스 제어 정책의 Advanced(고급) 탭에서 찾을 수 있습니다.	TCP_Embryonic_Conn_Limit, TCP_Embryonic_Conn_Timeout
tcp_conn_timeout (사용되지 않음)	TCP 원시 연결 시간 초과를 구성하는 데 사용되는 파라미터입니다. 버전 6.3부터 Firepower Threat Defense Service 정책에서 이러한 기능을 구성합니다. 해당 정책은 디바이스에 할당된 액세스 제어 정책의 Advanced(고급) 탭에서 찾을 수 있습니다.	TCP_Embryonic_Conn_Timeout
tcpMssBytes	최대 세그먼트 크기(바이트)입니다.	Sysopt_basic
tcpMssMinimum	이 플래그가 true인 경우에만 설정되는 최대 세그먼트 크기(MSS)를 설정할지 여부를 확인합니다.	Sysopt_basic
threat_detection_statistics	TCP 가로채기에 대한 위협 탐지 통계에 사용되는 파라미터입니다.	Threat_Detection_Configure

이름	설명	관련 FlexConfig 개체
vxlan_Nve_Only	<p>인터페이스에서 NVE 전용 구성을 위한 파라미터:</p> <ul style="list-style-type: none"> • 인터페이스의 논리적 이름 • IPv4 주소(라우팅된 인터페이스에 대한 선택 사항) • IPv4 넷마스크(라우팅된 인터페이스에 대한 선택 사항) 	VxLAN_Make_Nve_Only
vxlan_Port_And_Nve	<p>포트 및 VXLAN용 NVE 구성에 사용되는 파라미터:</p> <ul style="list-style-type: none"> • vxlan port • 소스 인터페이스(논리적 이름) • 유형(피어 또는 mcast) • 피어 IP 주소 또는 default-mcast-group 	VxLAN_Configure_Port_And_Nve
vxlan_Vni	<p>VNI 생성에 사용되는 파라미터:</p> <ul style="list-style-type: none"> • 인터페이스 번호(1-10000) • segment-id(1-16777215) • nameif(논리적 인터페이스 이름) • 유형(라우팅 또는 투명) • IP 주소(라우팅된 모드 디바이스의 경우 사용) 또는 브리지 그룹 번호(투명 모드 디바이스의 경우 사용) • 넷마스크(디바이스가 라우팅 모드에 있는 경우) 또는 사용되지 않음 	VxLAN_Make_Vni
wccpPassword	WCCP 비밀번호.	Wccp_Configure

FlexConfig 정책에 대한 요구 사항 및 사전 요건

모델 지원

Threat Defense

지원되는 도메인

모든

사용자 역할

관리자

FlexConfig 가이드라인 및 제한 사항

- FlexConfig 정책을 잘못 수행하면 실패한 FlexConfig가 포함된 구축 시도의 모든 변경 사항이 롤백됩니다. 구축 실패로 인한 롤백에는 구성 지우기가 포함되므로 네트워크에 지장을 줄 수 있습니다. 업무 시간 외 FlexConfig 변경 사항을 포함하는 구축을 고려하십시오. 또한 FlexConfig 변경 사항만 포함되도록 구축을 격리하고 다른 정책 업데이트는 고려하지 마십시오.
- VxLAN_Make_VNI 개체를 사용하는 경우 클러스터 또는 고가용성 쌍을 구성하기 전에 동일한 FlexConfig를 클러스터 또는 고가용성 쌍의 모든 유닛에 구축해야 합니다. Management Center에서는 클러스터 또는 고가용성 쌍을 만들기 전에 모든 디바이스에서 VxLAN 인터페이스를 일치시켜야 합니다.

FlexConfig 정책을 사용한 디바이스 구성 맞춤 설정

FlexConfig 정책을 사용하여 threat defense 디바이스 구성을 사용자 정의합니다.

FlexConfig를 사용하기 전에 management center의 다른 기능을 사용하여 필요한 모든 정책과 설정을 구성합니다. FlexConfig는 threat defense와 호환되지만 management center에서는 구성할 수 없는 ASA 기반 기능을 구성하기 위한 마지막 수단입니다.

다음은 FlexConfig 정책을 구성하고 구축하기 위한 엔드 투 엔드 절차입니다.

프로시저

단계 1 구성하려는 CLI 명령 시퀀스를 결정합니다.

ASA 디바이스에서 작동하는 구성이 있는 경우 **show running-config**를 사용하여 필요한 명령 시퀀스를 가져옵니다. 필요에 따라 인터페이스 이름 및 IP 주소와 같은 항목을 조정합니다.

이 기능이 새로운 기능인 경우 올바른 명령 시퀀스인지 확인하기 위해 실험실 설정에서 ASA 디바이스에 구현하는 것이 가장 좋습니다.

자세한 내용은 다음 주제를 참고하십시오.

- [FlexConfig 정책에 대한 추천 사용, 2 페이지](#)
- [FlexConfig 개체의 CLI 명령, 2 페이지](#)

단계 2 목차에서 **Objects(개체) > Object Management(개체 관리)**를 선택한 다음 **FlexConfig > FlexConfig Objects(FlexConfig 개체)**를 선택합니다.

사전 정의된 FlexConfig 개체를 검사하여 필요한 명령을 생성할 수 있는지 확인합니다. **View(보기)** (👁)을 클릭해서 개체 내용을 볼 수 있습니다. 기존 개체가 원하는 개체에 가까울 경우 해당 개체의 사본을 만든 다음 편집합니다. [사전 정의된 FlexConfig 개체, 13 페이지](#)의 내용을 참조하십시오.

개체를 검사하면 FlexConfig 개체의 구조, 명령 구분 및 예상되는 시퀀스에 대한 아이디어를 얻을 수 있습니다.

참고 직접 또는 사본으로 사용할 개체를 찾는 경우 개체의 맨 아래에 있는 변수 목록을 검사합니다. 시스템 변수인 **SYS**로 시작하는 모든 대문자를 제외한 변수 이름을 기록합니다. 이러한 변수는 특히 기본값 열에서 해당 개체에 값이 없다는 것이 표시되는 경우 편집하고 재정의해야 하는 텍스트 개체입니다.

단계 3 자체 FlexConfig 개체를 생성해야 하는 경우 필요한 변수를 결정하고 관련 개체를 생성합니다.

배포해야 하는 CLI에는 IP 주소, 인터페이스 이름, 포트 번호 및 시간 경과에 따라 조정할 수 있는 기타 파라미터가 포함될 수 있습니다. 이러한 변수는 필요한 값을 포함하는 개체를 가리키는 변수로 가장 잘 격리되어 있습니다. 구성의 일부이지만 시간 경과에 따라 변경될 수 있는 문자열에 대한 변수가 필요할 수도 있습니다.

또한 정책을 할당할 각 디바이스마다 다른 값이 필요한지 여부도 결정합니다. 예를 들어 세 가지 디바이스에 기능을 구성하려고 할 수 있지만 각 디바이스에 대해 지정된 명령에 다른 인터페이스 이름이나 IP 주소를 지정해야 할 수 있습니다. 각 디바이스에 대해 개체를 사용자 정의해야 하는 경우 개체를 생성할 때 재정의할 활성화하도록 설정한 다음 디바이스별로 재정의의 값을 정의합니다.

다양한 유형의 변수에 대한 설명과 필요한 경우 관련 개체를 구성하는 방법에 대해서는 다음 주제를 참조하십시오.

- [FlexConfig 변수, 6 페이지](#)
- [FlexConfig 정책 개체 변수, 11 페이지](#)
- [FlexConfig 시스템 변수, 12 페이지](#)
- [FlexConfig 텍스트 개체 설정, 29 페이지](#)

단계 4 사전 정의된 FlexConfig 개체를 사용하는 경우 변수로 사용되는 텍스트 개체를 편집하십시오.

[FlexConfig 텍스트 개체 설정, 29 페이지](#)의 내용을 참조하십시오.

단계 5 (필요한 경우) FlexConfig 개체 구성, [25 페이지](#).

사전 정의된 개체가 작업을 수행할 수 없는 경우에만 개체를 생성해야 합니다.

단계 6 [FlexConfig 정책 설정, 31 페이지](#).

단계 7 [FlexConfig 정책에 대한 대상 디바이스 설정, 32 페이지](#).

정책을 생성할 때 디바이스에 해당 정책을 할당할 수도 있습니다. 미리보기 전에 정책에 할당된 디바이스가 하나 이상 있어야 합니다.

단계 8 [FlexConfig 정책 미리보기, 33 페이지](#).

정책을 미리보기 전에 변경 사항을 저장해야 합니다.

생성된 명령이 의도된 것인지, 모든 변수가 올바르게 확인되는지 확인하십시오.

단계 9 메뉴 모음에서 **Deploy(구축) > Deployment(구축)**를 선택합니다.

단계 10 정책에 할당된 디바이스를 선택하고 **Deploy(구축)**를 클릭합니다.

구축이 완료될 때까지 기다립니다.

단계 11 [구축된 설정 확인, 34 페이지](#).

단계 12 (필요한 경우) [FlexConfig를 사용하여 설정된 기능 제거, 36 페이지](#).

다른 유형의 정책과 달리 디바이스에서 FlexConfig를 할당 해제하면 관련 구성이 제거되지 않을 수 있습니다. FlexConfig 생성 구성을 제거하려면 언급된 절차를 따릅니다.

이제 제품에서 직접 지원되기 때문에 기능을 제거하는 경우 [FlexConfig에서 관리되는 기능으로 변환, 37 페이지](#)의 내용도 참조하십시오.

FlexConfig 개체 구성

FlexConfig 개체를 사용하여 디바이스에 구축할 구성을 정의합니다. 각 FlexConfig 정책은 FlexConfig 개체 목록으로 구성되므로 개체는 본질적으로 Apache Velocity 스크립팅 명령, ASA 소프트웨어 구성 명령 및 변수로 구성된 코드 모듈입니다.

직접 사용할 수 있는 사전 정의된 FlexConfig 개체가 있으며, 편집해야 할 경우 사본을 만들 수 있습니다. 처음부터 자체 개체를 생성할 수도 있습니다. FlexConfig 개체의 콘텐츠는 단순한 단일 명령 문자 열에서부터 CLI 명령어 구조를 정의하는 범위까지 다양합니다. 이때 이 구조는 변수 및 스크립팅 명령을 사용하여 콘텐츠가 디바이스 간 또는 구축 간 다를 수 있는 명령을 구축합니다.

FlexConfig 정책을 정의할 때 FlexConfig 정책 개체를 생성할 수도 있습니다.

시작하기 전에

다음에 유의해야 합니다.

- FlexConfig 개체는 명령으로 변환된 다음 디바이스에 구축됩니다. 이러한 명령은 이미 전역 구성 모드에서 실행됩니다. 따라서 **enable** 및 **configure terminal** 명령을 FlexConfig 개체의 일부로 포함하지 마십시오.
- 필요한 변수 유형을 결정하고 필요한 모든 정책 개체를 생성합니다. FlexConfig 개체를 편집하는 동안 변수에 대한 개체를 생성할 수 없습니다.
- 명령이 어떤 방식으로든 VPN 또는 디바이스의 액세스 제어 구성과 충돌하지 않는지 확인합니다.
- 인터페이스에 대해 두 개 이상의 명령 집합이 있는 경우 마지막 명령 집합만 구축됩니다. 따라서 시작 및 끝 명령을 사용하여 인터페이스를 구성하지 않는 것이 좋습니다. 인터페이스 구성 예제는 사전 정의된 `ISIS_Interface_Configuration` FlexConfig 개체를 참조하십시오.

프로시저

단계 1 **Objects(개체) > Object Management(개체 관리)**을(를) 선택합니다.

단계 2 개체 유형 목록에서 **FlexConfig > FlexConfig Object(FlexConfig 개체)**를 선택합니다.

단계 3 다음 중 하나를 수행합니다.

- 새 개체를 생성하려면 **Add FlexConfig Object(FlexConfig 개체 추가)**를 클릭합니다.
- **Edit(수정)** (✎)을 클릭하여 기존 개체를 편집합니다.
- **View(보기)** (👁)을 클릭하여 사전 정의된 개체 콘텐츠를 볼 수 있습니다.
- 사전 정의된 개체를 편집하려면 **Copy(복사)** (📄)을 클릭하여 동일한 콘텐츠로 새 개체를 생성합니다.

단계 4 개체의 이름과 설명(선택 사항)을 입력합니다.

단계 5 개체 본문 영역에 명령 및 지침을 입력하여 필요한 구성을 생성합니다.

개체 콘텐츠는 올바른 ASA 소프트웨어 명령 시퀀스를 생성하는 일련의 스크립팅 명령 및 구성 명령입니다. **threat defense** 디바이스는 ASA 소프트웨어 명령을 사용하여 일부 기능을 구성합니다. 스크립팅 및 구성 명령에 대한 자세한 내용은 다음을 참조하십시오.

- [템플릿 스크립트, 6 페이지](#)
- [FlexConfig 개체의 CLI 명령, 2 페이지](#)

변수를 사용하여 실행 시간에만 알 수 있거나 디바이스 간에 다를 수 있는 정보를 제공할 수 있습니다. 처리 변수를 입력하면 되지만 **Insert(삽입)** 메뉴를 사용하여 정책 개체 또는 시스템 변수와 연관된 변수 또는 비밀 키인 변수를 추가해야 합니다. 변수에 대한 자세한 내용은 [FlexConfig 변수, 6 페이지](#) 섹션을 참조하십시오.

- 시스템 변수를 삽입하려면 **Insert(삽입) > Insert System Variable(시스템 변수 삽입) > Variable Name(변수 이름)**을 선택합니다. 이 변수에 대한 자세한 설명은 [FlexConfig 시스템 변수, 12 페이지](#) 섹션을 참조하십시오.
- 정책 개체 변수를 삽입하려면 **Insert(삽입) > Insert Policy Object(정책 개체 삽입) > Object Type(개체 유형)**을 선택하여 적절한 유형의 개체를 선택합니다. 그런 다음 변수에 이름(관련 정책 개체와 동일한 이름일 수 있음)을 제공하고 변수와 연결할 개체를 선택한 후 **Save(저장)**를 클릭합니다. 이 유형에 대한 자세한 설명은 [FlexConfig 정책 개체 변수, 11 페이지](#) 섹션을 참조하십시오. 자세한 절차는 [FlexConfig 개체에 정책 개체 변수 추가, 28 페이지](#) 섹션을 참조하십시오.
- 비밀 키 변수를 삽입하려면 **Insert(삽입) > Secret Key(비밀 키)**를 선택하고 변수 이름 및 값을 정의합니다. 자세한 절차는 [비밀 키 구성, 29 페이지](#) 섹션을 참조하십시오.

참고 새 정책 개체 또는 시스템 변수를 생성하려면 **Insert**(삽입) 메뉴를 사용해야 합니다. 하지만 해당 변수의 후속 사용을 위해 \$를 포함하여 입력해야 합니다. 이는 시스템 변수에서도 마찬가지입니다. 처음 사용할 때 **Insert**(삽입) 메뉴에서 추가하십시오. 그런 다음 후속 사용을 위해 입력합니다. 시스템 변수에 **Insert**(삽입) 메뉴를 두 번 이상 사용하면 시스템 변수가 변수 목록에 여러 번 추가되고 FlexConfig가 확인하지 않으므로 변경 사항을 저장할 수 없습니다. 변수(정책 개체 또는 시스템 변수와 연관되지 않은 변수)를 처리하려면 변수를 입력합니다. 비밀 키를 추가하는 경우 항상 **Insert**(삽입) 메뉴를 사용하십시오. 비밀 키 변수는 변수 목록에 표시되지 않습니다.

단계 6 구축 빈도 및 유형을 선택합니다.

- **Deployment**(구축) - 개체에 명령을 한 번 또는 항상 구축할지 여부입니다. 올바른 옵션을 선택하는 유일한 방법은 구축 결과를 테스트하는 것입니다.

Everytime(항상)을 선택하여 시작합니다. 그런 다음 FlexConfig 정책에 개체를 연결한 후 구성을 구축합니다. 구축이 성공적으로 완료되면 FlexConfig 정책으로 돌아가서 [FlexConfig 정책 미리보기, 33 페이지](#)에 설명된 대로 할당된 디바이스 중 하나의 구성을 미리 봅니다. `###CLI generated from managed features ###` 섹션에 개체의 명령을 지우거나 무효화하는 명령이 포함되어 있고 `###Flex-config Appended CLI ###` 섹션에 기능을 재구성하는 명령이 포함되어 있는 경우

Everytime(항상)이 적절한 옵션입니다.

명령 무효화가 표시되지 않더라도 디바이스 구성을 약간 변경한 다음 다른 구축을 실행합니다. 구축이 성공적으로 완료되면 구축 내역을 확인할 수 있습니다([구축된 설정 확인, 34 페이지](#) 참조). 오류가 발생하지 않은 상태에서 명령이 다시 실행된 것을 확인하면 **Everytime**(항상)을 계속 유지할 수 있습니다.

시스템에서 개체를 다시 발급하기 전에 먼저 개체의 명령을 무효화하지 않거나 구축으로 인해 명령과 관련된 오류가 발생하는 경우에만 **Once**(한 번)로 변경합니다. 경우에 따라 시스템에서 이미 구성된 명령을 실행할 수 없지만 이는 예외의 경우입니다.

몇 가지 추가 팁:

- FlexConfig 개체가 네트워크 또는 ACL 개체와 같은 시스템 관리 개체를 가리키는 경우 **Everytime**(항상)을 선택합니다. 그렇지 않으면 개체에 대한 업데이트가 구축되지 않을 수 있습니다.
- 개체에서 구성을 지우는 작업만 수행하는 경우 **Once**(한 번)를 사용합니다. 그리고 다음 구축 이후에 FlexConfig 정책에서 개체를 제거합니다.

- **Type**(유형) - 다음 중 하나를 선택합니다.

- **Append**(뒤에 추가) - (기본값) 개체의 명령은 management center 정책에서 생성된 구성의 마지막에 배치됩니다. 관리 개체에서 생성된 객체를 가리키는 정책 객체 변수를 사용하는 경우 Append(뒤에 추가)를 사용해야 합니다. 다른 정책에 대해 생성된 명령이 개체에 지정된 명령과 중복되면 이 옵션을 선택하여 명령을 덮어쓰지 않아야 합니다. 이는 가장 안전한 옵션입니다.
- **Prepend**(앞에 추가) - 개체의 명령은 management center 정책에서 생성된 구성의 앞에 배치됩니다. 일반적으로 구성을 지우거나 무효화하는 명령에는 Prepend(앞에 추가)를 사용합니다.

단계 7 (선택 사항). 개체 본문 위에 있는 **Validate**(확인)를 클릭하여 스크립트의 무결성을 확인합니다.

Save(저장)를 클릭하면 항상 개체가 확인됩니다. 잘못된 개체는 저장할 수 없습니다.

단계 8 **Save**(저장)를 클릭합니다.

다음에 수행할 작업

- 활성 정책이 개체를 참조하는 경우 구성 변경 사항을 구축합니다. 참조.

FlexConfig 개체에 정책 개체 변수 추가

다른 유형의 정책 개체와 연결된 FlexConfig 정책 개체에 변수를 삽입할 수 있습니다. FlexConfig가 디바이스에 구축되면 이 변수는 연결된 개체의 이름이나 콘텐츠로 확인됩니다.

FlexConfig 개체에서 정책 개체 변수를 처음 사용할 때는 다음 절차를 사용합니다. 개체를 다시 참조해야 하는 경우 변수(\$ 기호 포함)를 입력합니다. 이러한 변수를 사용하는 방법을 알아보려면 [변수 처리 방법, 7 페이지](#) 섹션을 참조하십시오.

프로시저

단계 1 **Insert**(삽입) > **Insert Policy Object**(정책 개체 삽입) > **Object Type**(개체 유형)을 선택하여 적절한 유형의 개체를 선택합니다.

단계 2 변수의 이름과 설명(선택 사항)을 입력합니다.

이름은 FlexConfig 개체의 컨텍스트 내에서 고유해야 합니다. 공백을 포함할 수 없습니다. 변수와 연관된 개체와 정확히 동일한 이름을 사용할 수 있습니다.

단계 3 변수와 연결할 개체를 선택하고 **Add**(추가)를 클릭하여 **Selected Object**(선택한 개체) 목록으로 이동합니다.

변수를 단일 개체에만 연결할 수 있습니다.

참고 텍스트 개체의 경우 필요에 따라 사전 정의된 개체를 선택할 수 있습니다. 하지만 이러한 개체의 대부분에는 기본값이 없습니다. 필요한 값을 직접 추가하거나 FlexConfig 개체를 구축할 디바이스의 개체를 재정의로 업데이트해야 합니다. 이러한 개체를 업데이트하지 않고 FlexConfig를 구축하려고 하면 일반적으로 구축 오류가 발생합니다.

단계 4 **Save**(저장)를 클릭합니다.

변수는 FlexConfig 개체 편집기의 하단에 있는 변수 목록에 나타납니다.

비밀 키 구성

비밀 키는 비밀번호와 같이 콘텐츠를 마스크 처리하려는 단일 문자열 변수입니다. 이 시스템은 민감한 정보의 전달을 방지하기 위해 이러한 변수에 대한 특수 취급 방식을 제공합니다.

비밀 키는 FlexConfig 개체의 변수 목록에 표시되지 않습니다

다음 절차에 따라 FlexConfig 개체에서 비밀 키 변수를 생성, 삽입 및 관리하십시오. 다른 유형의 변수와 달리 지정된 비밀 키 변수를 삽입해야 할 때마다 **Insert(삽입)** 명령을 사용할 수 있습니다. 처리와 관련하여 이러한 변수는 단일 값 텍스트 개체 변수처럼 동작합니다. [단일 값 변수, 7 페이지](#) 섹션을 참조하십시오.



참고 비밀 키 변수에 정의된 모든 데이터는 FlexConfig 정책을 미리 볼 때를 제외하고 사용자로부터 마스크 처리됩니다. 또한 FlexConfig 정책을 내 보내면 모든 비밀 키 변수의 콘텐츠가 지워집니다. 정책을 가져올 때 각 비밀 키 변수를 수동으로 편집하여 데이터를 입력해야 합니다.

프로시저

단계 1 FlexConfig 정책 개체를 편집하는 동안 **Insert(삽입) > Secret Key(비밀 키)**를 선택합니다.

단계 2 Insert Secret Key(비밀 키 삽입) 대화 상자에서 다음 중 하나를 수행합니다.

- 새 키를 생성하려면 **Add Secret Key(비밀 키 추가)**를 클릭한 후 다음 정보를 입력하고 **Add(추가)**를 클릭합니다.
 - **Secret Key Name(비밀 키 이름)** - 변수 이름입니다. 이 이름은 접두사가 @인 FlexConfig 개체에 나타납니다.
 - **Password(비밀번호), Confirm Password(비밀번호 확인)** - 입력할 때 별표로 마스크 처리되는 비밀 문자열입니다.
- FlexConfig 개체에 비밀 키 변수를 삽입하려면 변수의 확인란을 선택합니다.
- 비밀 키 변수 값을 편집하려면 해당 변수의 **Edit(수정)** (✎)을 클릭합니다. 필요에 맞게 변경하고 **Add(추가)**를 클릭합니다.
- 비밀 키를 삭제하려면 해당 변수의 **Delete(삭제)** (🗑️)을 클릭합니다.

단계 3 **Save(저장)**를 클릭합니다.

FlexConfig 텍스트 개체 설정

FlexConfig 개체의 텍스트 개체를 정책 개체 변수의 대상으로 사용합니다. 변수를 사용하여 실행 시간에만 알 수 있거나 디바이스 간에 다를 수 있는 정보를 제공할 수 있습니다. 구축 중에 텍스트 개체를 가리키는 변수는 텍스트 개체의 콘텐츠로 대체됩니다.

텍스트 개체에는 키워드, 인터페이스 이름, 숫자, IP 주소 등 자유 형식 문자열이 포함됩니다. 콘텐츠는 FlexConfig 스크립트 내에서 정보를 사용하는 방법에 따라 다릅니다.

텍스트 개체를 생성하거나 편집하기 전에 필요한 콘텐츠를 정확히 결정하십시오. 여기에는 개체를 처리하는 방법이 포함되어 있어 단일 문자열 또는 다중 문자열 개체를 생성할지 결정하는 데 도움이 됩니다. 다음 주제를 읽습니다.

- [FlexConfig 변수, 6 페이지](#)
- [변수 처리 방법, 7 페이지](#)

프로시저

단계 1 **Objects(개체) > Object Management(개체 관리)**을(를) 선택합니다.

단계 2 개체 유형 목록에서 **FlexConfig > Text Object(텍스트 개체)**를 선택합니다.

단계 3 다음 중 하나를 수행합니다.

- 새 개체를 생성하려면 **Add Text Object(텍스트 개체 추가)**를 클릭합니다.
- **Edit(수정)** (✎)을 클릭하여 기존 개체를 편집합니다. 사전 정의된 텍스트 개체를 편집할 수 있으며, 이는 사전 정의된 FlexConfig 개체를 사용하려는 경우 필요합니다.

단계 4 개체의 이름과 설명(선택 사항)을 입력합니다.

단계 5 (새 개체에만 해당) 드롭다운 목록에서 **Variable Type(변수 유형)**을 선택합니다.

- **Single(단일)** - 개체가 단일 텍스트 문자열을 포함해야 하는 경우
- **Multiple(다중)** - 개체가 텍스트 문자열 목록을 포함해야 하는 경우

개체를 저장 한 후에는 변수 유형을 변경할 수 없습니다.

단계 6 변수 유형이 **Multiple(다중)**인 경우 위쪽 및 아래쪽 화살표를 사용하여 **Count(개수)**를 지정합니다.

숫자를 변경하면 개체에서 행이 추가되거나 제거됩니다.

단계 7 개체에 콘텐츠를 추가합니다.

변수 번호 옆에 있는 텍스트 상자를 클릭하고 값을 입력하거나 텍스트 개체를 사용하는 FlexConfig 개체가 할당될 각 디바이스에 대한 디바이스 재정의를 설정할 수 있습니다. 두 가지 작업을 모두 수행할 수도 있습니다. 이 경우 기본 객체에 구성된 값은 지정된 디바이스에 대한 재정의가 없는 경우 기본값으로 사용됩니다.

사전 정의된 개체를 편집할 때는 디바이스 재정의를 사용하는 것이 좋으므로 다른 FlexConfig 정책에서 개체를 사용해야 할 수 있는 다른 사용자가 시스템 기본값을 그대로 유지해야 합니다. 접근 방식은 조직의 요구 사항에 따라 다릅니다.

팁 일부 사전 정의된 개체는 각 값이 특정 용도로 사용되는 여러 값을 필요로 합니다. 설명 텍스트를 주의깊게 읽고 개체의 예상 값을 확인하십시오. 일부 경우에 지침에는 기준 값을 변경하는 대신 재정의를 사용해야 한다고 명시되어 있습니다. `enableInspectProtocolList`의 경우 해당 프로토콜 검사가 Snort 검사와 호환되지 않는 프로토콜을 입력할 수 없습니다.

디바이스 재정의를 사용하려면 다음을 수행합니다.

- Allow Overrides**(재정의 허용)를 선택합니다.
- Overrides**(재정의) 영역을 확장하고(필요한 경우) **Add**(추가)를 클릭합니다.
해당 디바이스에 대한 재정의가 이미 있는 경우 해당 재정의를 클릭하고 편집하여 변경합니다.
- 개체 재정의 추가 대화 상자의 **Targets**(대상)에서 값을 정의할 디바이스를 선택하고 **Add**(추가)를 클릭하여 **Selected Devices**(선택한 디바이스) 목록으로 이동합니다.
- Override**(재정의)를 클릭하고 필요에 따라 **Count**(개수)를 조정한 다음 변수 필드를 클릭하고 디바이스에 대한 값을 입력합니다.
- Add**(추가)를 클릭합니다.

단계 8 **Save**(저장)를 클릭합니다.

다음에 수행할 작업

- 활성 정책이 개체를 참조하는 경우 구성 변경 사항을 구축합니다. 참조.

FlexConfig 정책 설정

FlexConfig 정책에는 FlexConfig 개체의 순서가 지정된 목록이 두 개 포함되어 있습니다. 하나는 앞에 추가된 목록이고 하나는 뒤에 추가된 목록입니다. 앞에 추가/뒤에 추가에 대한 설명은 [FlexConfig 개체 구성, 25 페이지](#) 섹션을 참조하십시오.

FlexConfig 정책은 여러 디바이스에 할당할 수 있는 공유 정책입니다.

프로시저

단계 1 **Devices**(디바이스) > **FlexConfig**을(를) 선택합니다.

단계 2 다음 중 하나를 수행합니다.

- **New Policy**(새 정책)를 클릭하여 새 FlexConfig 정책을 생성합니다. 이름을 입력하라는 메시지가 표시됩니다. 선택적으로 **Available Devices**(사용 가능한 디바이스) 목록에서 디바이스를 선택하고 **Add to Policy**(정책에 추가)를 클릭하여 디바이스를 할당합니다. **Save**(저장)를 클릭합니다.
- **Edit**(수정) (✎)을 클릭하여 기존 규칙을 편집합니다. 편집 모드에서 이름이나 설명을 클릭하여 변경할 수 있습니다.

- 동일한 콘텐츠의 새 정책을 만들려면 **Copy(복사)** ()를 클릭합니다. 이름에 대한 메시지가 표시됩니다. 디바이스 할당은 사본에 대해 유지되지 않습니다.
- 더 이상 필요하지 않은 정책을 제거하려면 **Delete(삭제)**를 클릭합니다.

단계 3 Available FlexConfig(사용 가능한 **FlexConfig**) 목록에서 정책에 필요한 FlexConfig 개체를 선택하고 >를 클릭하여 정책에 추가합니다.

개체는 FlexConfig 개체에 지정된 구축 유형에 따라 자동으로 목록의 앞이나 뒤에 추가됩니다.

선택한 개체를 제거하려면 개체 옆에 있는 **Delete(삭제)** ()를 클릭합니다.

단계 4 선택한 각 개체에 대해 해당 개체 옆에 있는 **View(보기)** ()를 클릭하여 개체에 사용된 변수를 식별합니다.

SYS로 시작하는 시스템 변수를 제외하고 변수와 연관된 개체가 비어 있지 않도록 해야 합니다. 빈칸이나 괄호 사이에 아무것도 없는 경우 []는 빈 개체를 나타냅니다. 정책을 구축하기 전에 이러한 개체를 편집해야 합니다.

참고 개체 재정의를 사용하면 해당 값이 이 보기에 표시되지 않습니다. 따라서 기본값이 비어 있어도 필요한 값이 있는 개체를 업데이트하지 않았다는 의미는 아닙니다. 구성을 미리 보면 지정된 디바이스에 대한 변수가 올바르게 확인되는지 여부가 표시됩니다. [FlexConfig 정책 미리보기, 33 페이지](#)를 참조하십시오.

단계 5 Save(저장)를 클릭합니다.

다음에 수행할 작업

- 정책의 대상 디바이스를 설정합니다. [FlexConfig 정책에 대한 대상 디바이스 설정, 32 페이지](#) 섹션을 참조하십시오.
- **Deploy configuration changes(구성 변경 사항 구축)** 참조.

FlexConfig 정책에 대한 대상 디바이스 설정

FlexConfig 정책을 생성할 때 정책을 사용하는 디바이스를 선택할 수 있습니다. 이후에 아래 설명된 대로 정책에 대한 디바이스 할당을 변경할 수 있습니다.



참고 일반적으로 디바이스에서 정책 할당을 취소하면 시스템은 다음 구축 시 관련 구성을 자동으로 제거합니다. 하지만 FlexConfig 개체는 사용자 정의 명령을 구축하기 위한 스크립트이기 때문에 디바이스에서 FlexConfig 정책 할당을 해제하지 않아도 FlexConfig 개체가 구성한 명령은 제거되지 않습니다. 디바이스 구성에서 FlexConfig 생성 명령을 제거하려는 경우 [FlexConfig를 사용하여 설정된 기능 제거, 36 페이지](#) 섹션을 참조하십시오.

프로시저

단계 1 **Devices > FlexConfig**를 선택하고 FlexConfig 정책을 편집합니다.

단계 2 **Policy Assignments**(정책 할당)를 클릭합니다.

단계 3 **Targeted Devices**(대상 디바이스)에 대상 목록을 만듭니다.

- Add(추가) - 하나 이상의 **Available Devices**(사용 가능한 디바이스)를 선택한 다음 **Add to Policy**(정책에 추가)를 클릭하거나 **Selected Devices**(선택한 디바이스) 목록으로 드래그 앤 드롭합니다. 정책을 디바이스, 고가용성 쌍 및 클러스터된 디바이스에 할당할 수 있습니다.
- Delete(삭제) - 단일 디바이스 옆에 있는 **Delete**(삭제) ()을 클릭하거나 여러 디바이스를 선택하고 오른쪽 클릭한 다음 **Delete Selection**(선택 사항 삭제)을 선택합니다.

단계 4 선택 사항을 저장하려면 **OK**(확인)를 클릭합니다.

단계 5 **Save**(저장)를 클릭하여 FlexConfig 정책을 저장합니다.

다음에 수행할 작업

- Deploy configuration changes(구성 변경 사항 구축)참조.

FlexConfig 정책 미리보기

FlexConfig 정책을 미리 보고 FlexConfig 개체를 CLI 명령으로 변환하는 방법을 확인합니다. 미리보기는 FlexConfig 개체에 사용되는 스크립트 및 변수에서 선택한 디바이스에 대해 생성될 명령을 표시합니다. 변수는 디바이스 구성에 따라 확인되므로 구축할 대상에 대해 명확히 알 수 있습니다.

미리보기를 사용하여 FlexConfig 개체의 잠재적 문제를 찾습니다. 미리보기에서 예상 결과가 표시될 때까지 개체를 편집합니다.

변수는 디바이스 구성에 따라 다르게 확인될 수 있으므로 각 디바이스에 대한 구성을 별도로 미리 봐야 합니다.

프로시저

단계 1 **Devices > FlexConfig**를 선택하고 FlexConfig 정책을 편집합니다.

단계 2 보류 중인 변경 내용이 있을 경우 **Save**(저장)를 클릭합니다.

미리보기에는 가장 최근에 저장된 정책 버전에 있던 FlexConfig 개체에 대한 결과만 표시됩니다. 새로 추가된 객체를 미리 보려면 정책을 저장해야 합니다.

단계 3 **Preview Config**(구성 미리보기)를 클릭합니다.

단계 4 **Select Device**(디바이스 선택) 드롭다운 목록에서 디바이스를 선택합니다.

시스템은 디바이스 및 구성된 정책에서 정보를 검색하고 다음 구축 시 디바이스에 대해 생성될 CLI 명령을 결정합니다. 출력을 선택하고 Ctrl+C를 사용하여 클립 보드에 복사할 수 있으며, 추가 분석을 위해 텍스트 파일에 붙여 넣을 수 있습니다.

미리보기에는 다음 섹션이 포함됩니다.

- Flex-config Prepended CLI - 구성에 대해 앞에 추가되는 FlexConfigs 생성 명령입니다.
- 관리되는 기능에서 생성된 CLI - management center에서 구성된 정책에 대해 생성된 명령입니다. 마지막으로 디바이스에 성공적으로 구축된 이후의 새 정책 또는 변경된 정책에 대해 명령이 생성됩니다. 이 명령은 할당된 정책을 구현하는 데 필요한 모든 명령을 나타내지는 않습니다. 이 섹션의 명령은 FlexConfig 개체에서 생성되지 않습니다.
- Flex-config Appended CLI - 구성에 대해 뒤에 추가되는 FlexConfigs 생성 명령입니다.

단계 5 **Close**(닫기)를 클릭하여 미리보기 대화 상자를 종료합니다.

구축된 설정 확인

FlexConfig 정책을 디바이스에 구축한 후 구축이 성공적이었으며 결과 구성이 예상한 그대로인지 확인합니다. 또한 디바이스가 예상대로 작동하는지 확인합니다.

프로시저

단계 1 구축에 성공했는지 확인하려면 다음을 수행합니다.

- a) 메뉴 모음에서 **System Status**(시스템 상태)를 클릭합니다. 이 메뉴는 **Deploy**(구축) 및 **System**(시스템) 사이에 이름이 지정되지 않은 메뉴입니다.

아이콘은 다음 중 하나와 같으며 오류가 있는 경우 숫자가 포함될 수 있습니다.

 - **Indicates No Warnings**(경고 없음 표시) — 시스템에 발생한 오류 및 경고가 없음을 나타냅니다.
 - **Indicates One or More Warnings**(경고가 하나 이상임을 표시) — 시스템에 오류 없이 하나 이상의 경고가 발생했음을 나타냅니다.
 - **Indicates One or More Errors**(오류가 하나 이상임을 표시) — 시스템에 하나 이상의 오류 및 경고가 발생했음을 나타냅니다.
- b) **Deployments**(구축)에서 구축에 성공했는지 확인합니다.
- c) 특히 실패한 구축에 대한 세부 정보를 보려면 **Show History**(기록 표시)를 클릭합니다.
- d) 왼쪽 열에 있는 작업 목록에서 구축 작업을 선택합니다.

작업 정보는 역순으로 나열되며 가장 최근의 작업이 목록 상단에 표시됩니다.
- e) 오른쪽 열의 디바이스에 대한 **Transcript**(기록) 열에서 다운로드를 클릭합니다.

구축 기록에는 디바이스로 전송된 명령과 디바이스에서 반환된 응답이 포함되어 있습니다. 이러한 응답은 정보 메시지 또는 오류 메시지일 수 있습니다. 장애가 발생한 구축의 경우 FlexConfig를 통해 전송한 명령 오류를 나타내는 메시지를 확인합니다. 이 오류를 확인하여 명령을 구성하려는 FlexConfig 개체의 스크립트를 수정할 수 있습니다.

참고 관리 기능에 대해 전송된 명령과 FlexConfig 정책에서 생성된 명령이 기록에서 구분되지 않습니다.

예를 들어 다음 시퀀스에서는 management center가 외부에서 논리적 이름으로 GigabitEthernet0/0을 구성하기 위해 명령을 전송했음을 확인할 수 있습니다. 디바이스에서 보안 수준을 0으로 자동 설정했다고 응답했습니다. Threat Defense에서는 어떠한 경우에도 보안 수준을 사용하지 않습니다. FlexConfig와 관련된 메시지는 기록의 CLI Apply(CLI 적용) 섹션에 있습니다.

```
===== CLI APPLY =====
FMC >> interface GigabitEthernet0/0
FMC >> nameif outside
FTDv 192.168.0.152 >> [info] : INFO: Security level for "outside" set to 0 by default.
```

단계 2 구축된 구성에 예상되는 명령이 포함되어 있는지 확인합니다.

디바이스의 관리 IP 주소에 SSH 연결을 설정하여 이 작업을 수행할 수 있습니다. **show running-config** 명령을 사용하여 구성을 봅니다.

또는 Secure Firewall Management Center 내의 CLI 툴을 사용합니다.

- a) **System(시스템) > Health(상태) > Monitor(모니터링)**을 선택하고 디바이스의 이름을 클릭합니다. 상태 테이블의 **Count(개수)** 열에서 열기/닫기 화살표를 클릭하여 모든 디바이스를 볼 수 있습니다.
- b) **Advanced Troubleshooting(고급 문제 해결)**을 클릭합니다.
- c) **Threat Defense CLI(위협 방어 CLI)**를 클릭합니다.
- d) **show**를 명령으로 선택하고 **running-config**를 파라미터로 입력합니다.
- e) **Execute(실행)**를 클릭합니다.

실행 중인 구성이 텍스트 상자에 나타납니다. 구성을 선택하고 Ctrl+C를 누른 다음 나중에 분석할 수 있도록 텍스트 파일에 붙여 넣을 수 있습니다.

단계 3 디바이스가 예상대로 작동하는지 확인합니다.

이 기능과 관련된 **show** 명령을 사용하여 자세한 정보와 통계를 확인합니다. 예를 들어 추가 프로토콜 검사를 활성화한 경우 **show service-policy** 명령은 이 정보를 제공합니다. 사용할 정확한 명령은 기능에 따라 다르며, ASA 설정 가이드 및 기능 구성 방법에 대해 알아보는 데 사용한 명령 참조에 나와 있습니다.

통계를 표시하는 명령에서 숫자가 변경되지 않음을 나타내면(예: 적중 횟수, 연결 수 등) 구성은 유효하지만 의미가 없습니다. 통계에 표시되어야 하는 디바이스를 통해 트래픽이 진행되고 있음을 알고 있는 경우 구성에서 누락된 부분을 찾습니다. 예를 들어 기능이 작동하기 전에 NAT 또는 액세스 규칙이 트래픽을 삭제하거나 변경했을 수 있습니다.

SSH 세션 또는 management center CLI 톨을 통해 **show** 명령을 사용할 수 있습니다.

하지만 사용해야 하는 **show** 명령을 threat defense CLI에서 직접 사용할 수 없는 경우 디바이스에 SSH 연결을 설정해서 명령을 사용해야 합니다. CLI에서 다음 명령 시퀀스를 입력하여 진단 CLI 내의 Privileged EXEC 모드로 진입합니다. 여기에서만 지원되는 **show** 명령을 입력할 수 있어야 합니다.

```
> system support diagnostic-cli
Attaching to Diagnostic CLI ... Press 'Ctrl+a then d' to detach.
Type help or '?' for a list of available commands.
firepower> enable
Password: <press enter, do not enter a password>
firepower#
```

FlexConfig를 사용하여 설정된 기능 제거

FlexConfig를 사용하여 구성한 일련의 구성 명령을 제거해야 한다고 결정한 경우 해당 구성을 수동으로 제거해야 할 수 있습니다. FlexConfig 정책을 디바이스에서 할당 해제하면 모든 구성이 제거되지 않을 수 있습니다.

구성을 수동으로 제거하려면 새 FlexConfig 개체를 생성하여 구성 명령을 지우거나 무효화합니다.

시작하기 전에

개체에 의해 생성된 구성의 일부 또는 전부를 수동으로 제거해야 하는지 확인하려면 다음을 수행합니다.

1. **FlexConfig 정책 미리보기, 33 페이지**에 설명된 대로 구성 미리보기를 검사합니다. `###CLI generated from managed features ###` 섹션에 FlexConfig 개체의 모든 명령을 제거하는 **clear** 또는 **negate** 명령이 포함되어 있으면 FlexConfig 정책, 저장 및 재구축 단계에서 개체를 간단히 제거할 수 있습니다.
2. FlexConfig 정책에서 개체를 제거하고 변경 사항을 저장한 다음 구성을 다시 미리 봅니다. `###CLI generated from managed features ###` 섹션에 필요한 **clear** 또는 **negate** 명령이 여전히 포함되어 있지 않으면 이 절차에 따라 수동으로 구성을 제거해야 합니다.

프로시저

단계 1 Objects(개체) > Object Management(개체 관리)를 선택하고 FlexConfig 개체를 생성하여 구성 명령을 지우거나 무효화합니다.

기능에 모든 구성 설정을 제거할 수 있는 **clear** 명령이 있는 경우 해당 명령을 사용합니다. 예를 들어 사전 정의된 `ISIS_Unconfigure_All` 개체에는 모든 ISIS 관련 구성 명령을 제거하는 단일 명령이 포함되어 있습니다.

```
clear configure router isis
```

해당 기능에 대한 **clear** 명령이 없으면 제거할 각 명령의 **no** 형식을 사용해야 합니다. 예를 들어 사전 정의된 `Sysopt_basic_negate` 개체는 사전 정의된 `Sysopt_basic` 개체를 통해 구성된 명령을 제거합니다.

```
no sysopt traffic detailed-statistics
```

```
no sysopt connection timewait
```

일반적으로 구성을 prepended, deploy once 개체로 제거하는 FlexConfig 개체를 구성합니다.

단계 2 Devices(디바이스) > FlexConfig를 선택하고 새 FlexConfig 정책을 생성하거나 기존 정책을 편집합니다.

구성 명령을 구축하는 FlexConfig 정책을 유지하려면 명령을 무효화하기 위한 새 정책을 생성하고 디바이스를 정책에 할당합니다. 그런 다음 새 FlexConfig 개체를 정책에 추가합니다.

FlexConfig 구성 개체를 모든 디바이스에서 완전히 제거하려면 기존 FlexConfig 정책에서 해당 명령을 삭제하고 구성을 무효화하는 개체로 간단히 대체할 수 있습니다.

단계 3 Save(저장)를 클릭하여 FlexConfig 정책을 저장합니다.

단계 4 Preview Config(구성 미리보기)를 클릭하고 clear 및 negation 명령이 올바르게 생성되는지 확인합니다.

단계 5 메뉴 모음에서 **Deploy(구축) > Deployment(구축)**를 클릭하고 디바이스를 선택하고 **Deploy(구축)**를 클릭합니다.

구축이 완료될 때까지 기다립니다.

단계 6 명령이 제거되었는지 확인합니다.

디바이스에서 실행 중인 구성을 보고 명령이 제거되었는지 확인합니다. 자세한 내용은 [구축된 설정 확인, 34 페이지](#) 섹션을 참조하십시오.

단계 7 FlexConfig 정책을 편집하는 동안 **Policy Assignments(정책 할당)**를 클릭하고 디바이스를 제거합니다. 선택적으로 정책에서 FlexConfig 개체를 제거합니다.

FlexConfig 정책이 원치 않는 구성 명령을 간단히 제거한다고 가정하면 제거가 완료된 후 정책을 디바이스에 할당하지 않아도 됩니다.

하지만 FlexConfig 정책이 디바이스에 구성하려는 옵션을 유지하는 경우 정책에서 무효화 개체를 제거합니다. 해당 개체는 더 이상 필요하지 않습니다.

FlexConfig에서 관리되는 기능으로 변환

각 소프트웨어 릴리스는 제품에 관리되는 기능, 즉 FlexConfig 외부에서 제어되는 정책을 통해 직접 구성하는 기능을 추가합니다. 이로 인해 현재 사용 중인 FlexConfig 명령의 지원이 중단될 수 있습니다. 구성은 자동으로 변환되지 않습니다. 업그레이드 후에는 새로 사용이 중단된 명령을 사용하여 FlexConfig 개체를 할당하거나 생성할 수 없습니다. 소프트웨어 업그레이드 후에는 FlexConfig 정책 및 개체를 검토하십시오.

FlexConfig를 사용하여 구성된 기능이 관리되는 기능으로 지원되기 시작하면 FlexConfig 사용에서 관리 기능 사용으로 변환해야 합니다. 대부분의 경우 기존 FlexConfig 구성은 업그레이드 후에도 계속 작동하며 계속 구축할 수 있습니다. 그러나 경우에 따라 사용되지 않는 명령을 사용하면 구축 문제가 발생할 수 있습니다. GUI와 FlexConfig 모두에서 기능을 구성하는 것은 지원되지 않습니다.

프로시저

단계 1 FlexConfig를 사용하여 설정된 기능 제거, 36 페이지에 설명된 대로 FlexConfig를 제거합니다.

단계 2 새로 지원되는 관리되는 기능에서 설정을 구성합니다.

릴리스 노트에는 릴리스의 새로운 기능 목록이 있습니다.

FlexConfig의 예시

다음은 FlexConfig 사용의 몇 가지 예입니다.

Precision Time Protocol을 구성하는 방법(ISA 3000)

PTP(Precision Time Protocol)는 패킷 기반 네트워크에서 다양한 디바이스의 클록을 동기화하기 위해 개발된 시간 동기화 프로토콜입니다. 이러한 디바이스 클록은 일반적으로 정밀도와 안정성이 다양합니다. 이 프로토콜은 산업, 네트워크에 연결된 측정 및 제어 시스템을 위해 특별히 설계되었으며 최소한의 대역폭 및 적은 처리 오버헤드를 필요로 하기 때문에 분산 시스템에서 사용하기에 가장 적합합니다.

PTP 시스템은 PTP 및 비 PTP 디바이스의 조합으로 구성된 분산형, 네트워크에 연결된 시스템입니다. PTP 디바이스에는 일반 클록, 경계 클록 및 투명 클록이 있습니다. 비 PTP 디바이스에는 네트워크 스위치, 라우터 및 기타 인프라 디바이스가 있습니다.

FTD 디바이스를 투명 클록이 되도록 구성할 수 있습니다. FTD 디바이스에서는 클록을 PTP 클록과 동기화하지 않습니다. FTD 디바이스에서는 PTP 클록에 정의된 대로 PTP 기본 프로필을 사용합니다.

PTP 디바이스를 구성할 때 함께 작동할 디바이스의 도메인 번호를 정의합니다. 따라서 여러 PTP 도메인을 구성한 다음, 하나의 특정 도메인에 대해 PTP 클록을 사용하도록 비 PTP 디바이스를 각각 구성할 수 있습니다.

시작하기 전에

디바이스에서 사용해야 하는 PTP 클록에 구성된 도메인 번호를 결정합니다. 이 예에서는 PTP 도메인 번호를 10으로 가정합니다. 또한 시스템에서 도메인의 PTP 클록에 연결하기 위해 통과하는 인터페이스를 결정합니다.

다음은 PTP 구성에 대한 지침입니다.

- 이 기능은 Cisco ISA 3000 어플라이언스에서만 사용할 수 있습니다.

- Cisco PTP는 멀티캐스트 PTP 메시지만 지원합니다.
- PTP는 IPv4 네트워크용으로만 사용할 수 있으며 IPv6 네트워크용으로는 사용할 수 없습니다.
- PTP 구성은 독립형 또는 브리지 그룹 멤버에 관계없이 물리적 이더넷 데이터 인터페이스에서 지원됩니다. 이는 관리 인터페이스, 하위 인터페이스, EtherChannel, BVI(Bridge Virtual Interfaces) 또는 기타 가상 인터페이스에서 지원되지 않습니다.
- VLAN 하위 인터페이스에서 이동하는 PTP가 지원되며 이때 적절한 PTP 구성이 현재 상위 인터페이스에 있다고 가정합니다.
- PTP 패킷이 디바이스를 통해 이동할 수 있는지 확인해야 합니다. PTP 트래픽은 UDP 대상 포트 319 및 320과 대상 IP 주소 224.0.1.129로 식별되므로 이 트래픽을 허용하는 액세스 제어 규칙이 작동해야 합니다.
- 라우팅 방화벽 모드에서는 PTP 멀티캐스트 그룹에 대해 멀티캐스트 라우팅을 활성화해야 합니다. 또한 PTP를 활성화하는 인터페이스가 브리지 그룹에 없는 경우에는 IGMP 멀티캐스트 그룹인 224.0.1.129에 조인하도록 인터페이스를 구성해야 합니다. 물리적 인터페이스가 브리지 그룹 멤버인 경우에는 IGMP 멀티캐스트 그룹에 조인하도록 구성하지 마십시오.

프로시저

단계 1 (라우팅된 모드 전용) 멀티캐스트 라우팅을 활성화하고, 인터페이스에 대한 IGMP 그룹을 구성합니다.

라우티드 모드에서는 멀티캐스트 라우팅을 활성화해야 합니다. 또한 독립형 물리적 인터페이스, 즉 브리지 그룹 멤버가 아닌 인터페이스의 경우에는 인터페이스가 224.0.1.129 IGMP 그룹에 가입하도록 구성해야 합니다. 브리지 그룹 멤버가 IGMP 그룹에 가입하도록 구성할 수는 없지만, 브리지 그룹 멤버의 PTP 구성은 IGMP에 가입하지 않아도 정상적으로 적용됩니다.

PTP를 구성할 디바이스별로 이 절차를 수행합니다.

참고 각 디바이스의 PTP 지향 인터페이스의 하드웨어 이름(예: GigabitEthernet1/1)을 기록합니다.

- Devices**(디바이스) > **Device Management**(디바이스 관리)를 선택하고 디바이스를 수정합니다.
- Routing**(라우팅)을 클릭합니다.
- Multicast Routing**(멀티캐스트 라우팅) > **IGMP**를 선택합니다.
- Enable Multicast Routing**(멀티캐스트 라우팅 활성화) 확인란을 선택합니다.
- Join Group**(조인 그룹)을 클릭합니다.
- Add**(추가)를 클릭하고, Add IGMP Join Group Parameters(IGMP 가입 그룹 매개변수 추가) 대화상자에서 다음 옵션을 구성한 다음 **OK**(확인)를 클릭합니다.
 - **Interface**(인터페이스) - PTP 지향 독립형 인터페이스를 선택합니다.
 - **Join Group**(조인 그룹) - 새 네트워크 개체를 추가하려면 +를 클릭합니다. 주소 224.0.1.129를 사용하여 호스트 개체를 생성합니다. 추가 인터페이스를 구성할 때는 이 개체를 선택하기만 하면 됩니다.

디바이스의 PTP 지향 독립형 인터페이스마다 이 단계를 반복합니다.

g) Routing(라우팅) 페이지에서 **Save(저장)**를 클릭합니다.

단계 2 FlexConfig 개체를 생성해 PTP를 전역 및 인터페이스에서 활성화합니다.

다음 절차에서는 구성 중인 모든 디바이스에서 PTP 지향 인터페이스가 동일하다고 가정합니다. 다른 디바이스에서 다른 인터페이스를 사용한다면, 각 고유 조합에 대해 별도의 개체를 생성해야 합니다. 예를 들어 디바이스 A와 B에서 GigabitEthernet1/1을 사용하고 디바이스 C와 D에서는 GigabitEthernet1/2를 사용하며, 디바이스 E와 F에서는 GigabitEthernet1/1 및 1/2를 모두 사용한다면, 별도의 FlexConfig 개체 3개와 (다음 단계에서 설명하는) 별도의 FlexConfig 정책 3개가 필요합니다.

- Objects(개체) > Object Management(개체 관리)**를 선택합니다.
- 목록에서 **FlexConfig > FlexConfig Object(FlexConfig 개체)**를 선택합니다.
- Add FlexConfig Object(FlexConfig 개체 추가)**를 클릭하고, 다음 속성을 구성한 다음 **Save(저장)**를 클릭합니다.

- **Name(이름)** - 개체 이름입니다. 예를 들어, Enable_PTP입니다.
- **Deployment(구축) - Everytime(항상)**을 선택합니다. 모든 구축에 구성을 전송해 설정 상태를 유지합니다.
- **Type(유형)** - 기본값인 **Append(추가)**를 그대로 유지합니다. 명령은 직접 지원 기능용 명령이 전송된 후에 디바이스에 전송됩니다. 이렇게 하면 인터페이스 구성에 대한 다른 변경 사항은 이러한 명령을 실행하기 전에 설정됩니다.
- **Object body(개체 본문)** - 개체 본문에는 PTP를 전역적으로 구성하고 각 PTP 지향 인터페이스에서 구성하는 데 필요한 명령을 입력합니다. 예를 들어 PTP 도메인 10의 전역 구성 및 GigabitEthernet1/1에서의 인터페이스 구성에 필요한 명령은 다음과 같습니다.

```
ptp mode e2transparent
ptp domain 10
interface gigabitethernet1/1
  ptp enable
```

개체 본문은 다음과 비슷해야 합니다.

Insert ▾		Deployment: Everytime ▾	Type: Append ▾
----------	--	--------------------------------------	-----------------------------

```
ptp mode e2transparent
ptp domain 10
interface gigabitethernet1/1
  ptp enable
```

단계 3 FlexConfig 정책을 생성하고 디바이스에 할당합니다.

다양한 PTP 지향 인터페이스 조합에 대한 여러 FlexConfig 개체를 생성했다면, 개체별로 FlexConfig 정책을 만들고 구성해야 하는 인터페이스에 맞는 올바른 디바이스에 정책을 할당해야 합니다. 디바이스 그룹별로 다음 절차를 반복합니다.

- Devices(디바이스) > FlexConfig**를 선택합니다.

- b) **New Policy**(새 정책)를 클릭하거나, 기존 FlexConfig 정책을 대상 디바이스에 할당해야 한다면(또는 이미 할당되어 있다면) 해당 정책을 수정합니다.

새 정책을 생성할 때는 정책 이름을 지정하는 대화 상자의 정책에 대상 디바이스를 할당합니다.

- c) 목차의 **User Defined**(사용자 정의) 폴더에서 PTP FlexConfig 개체를 선택하고 >을 클릭해 정책에 추가합니다.

개체는 **Selected Appended FlexConfigs** 목록에 추가해야 합니다.

Selected Append FlexConfigs		
#	Name	Description
1	Enable_PTP	

- d) **Save**(저장)를 클릭합니다.
- e) 아직 모든 대상 디바이스를 정책에 할당하지 않았다면 **Save**(저장) 아래에 있는 **Policy Assignments**(정책 할당) 링크를 클릭하여 할당합니다.
- f) **Preview Config**(구성 미리보기)를 클릭하고, **Preview**(미리보기) 대화 상자에서 할당된 디바이스 중 하나를 선택합니다.

시스템은 디바이스에 전송될 구성 CLI의 미리보기를 생성합니다. PTP FlexConfig 개체에서 생성된 명령이 올바르게 표시되는지 확인합니다. 미리보기 끝부분에 표시됩니다. 관리 대상 기능에 적용한 다른 변경 사항에서 생성된 명령도 함께 표시됩니다. PTP 명령의 경우 다음과 유사한 내용이 표시되어야 합니다.

```
###Flex-config Appended CLI ###
ptp mode e2transparent
ptp domain 10
interface gigabitethernet1/1
  ptp enable
```

단계 4 변경 사항을 배포합니다.

FlexConfig 정책을 디바이스에 할당했기 때문에 항상 구축 경고가 표시됩니다. FlexConfig는 주의해서 사용해야 한다는 뜻입니다. **Proceed**(계속하기)를 클릭하여 구축을 계속 진행합니다.

구축이 끝나면 구축 내역과 구축 기록을 확인할 수 있습니다. 이 기능은 구축이 실패했을 때 특히 유용합니다. [구축된 설정 확인, 34 페이지](#)의 내용을 참조하십시오.

단계 5 각 디바이스의 PTP 구성을 확인합니다.

각 디바이스의 SSH 또는 콘솔 세션에서 PTP 설정을 확인합니다.

```
> show ptp clock
PTP CLOCK INFO
  PTP Device Type: End to End Transparent Clock
  Operation mode: One Step
  Clock Identity: 34:62:88:FF:FE:1:73:81
  Clock Domain: 10
  Number of PTP ports: 4
> show ptp port
```

```

PTP PORT DATASET: GigabitEthernet1/1
Port identity: Clock Identity: 34:62:88:FF:FE:1:73:81
Port identity: Port Number: 1
PTP version: 2
Port state: Enabled

PTP PORT DATASET: GigabitEthernet1/2
Port identity: Clock Identity: 34:62:88:FF:FE:1:73:81
Port identity: Port Number: 2
PTP version: 2
Port state: Disabled

PTP PORT DATASET: GigabitEthernet1/3
Port identity: Clock Identity: 34:62:88:FF:FE:1:73:81
Port identity: Port Number: 3
PTP version: 2
Port state: Disabled

PTP PORT DATASET: GigabitEthernet1/4
Port identity: Clock Identity: 34:62:88:FF:FE:1:73:81
Port identity: Port Number: 4
PTP version: 2
Port state: Disabled

```

정전(ISA 3000)에 대한 자동 하드웨어 우회를 구성하는 방법

정전 상태에서도 인터페이스 쌍 간에 트래픽 플로우가 계속되도록 하드웨어 바이패스를 활성화할 수 있습니다. 지원되는 인터페이스 쌍은 구리 인터페이스 GigabitEthernet 1/1과 1/2 및 GigabitEthernet 1/3과 1/4입니다. 파이버 이더넷 모델을 사용하는 경우에는 구리 이더넷 쌍(GigabitEthernet 1/1 및 1/2)만 하드웨어 바이패스를 지원합니다.

하드웨어 바이패스가 활성화 상태이면 트래픽이 계층 1에서 이러한 인터페이스 쌍 간을 통과합니다. FTD CLI는 인터페이스가 중단되는 것으로 간주합니다. 방화벽 기능은 없으므로 트래픽의 디바이스 통과를 허용하는 경우의 위험을 파악해야 합니다.

CLI 콘솔 또는 SSH 세션에서 **show hardware-bypass** 명령을 사용하여 운영 상태를 모니터링합니다.

시작하기 전에

다음 조건을 충족해야 하드웨어 바이패스가 작동합니다.

- 같은 브리지 그룹에 인터페이스 쌍을 배치해야 합니다.
- 스위치의 액세스 포트에 인터페이스를 연결해야 합니다. 트렁크 포트에는 인터페이스를 연결하지 마십시오.

디바이스에 할당된 액세스 제어 정책에 연결된 Threat Defense Service 정책을 사용하여 TCP 시퀀스 번호 임의 설정을 전역적으로 비활성화하는 것이 좋습니다. 기본적으로 ISA 3000을 통과하는 TCP 연결의 ISN(초기 시퀀스 번호)는 임의의 숫자로 재작성됩니다. 하드웨어 바이패스를 활성화하면 ISA 3000은 더 이상 데이터 경로에 없으며 시퀀스 번호를 변환하지 않습니다. 수신 클라이언트는 예상치 않은 시퀀스 번호를 수신하므로 연결을 삭제합니다. 따라서 TCP 세션을 다시 설정해야 합니다. TCP 시퀀스 번호 임의 설정을 비활성화하더라도 전환 중에 일시적으로 중단되는 링크 때문에 일부 TCP 연결은 다시 설정해야 합니다.

프로시저

단계 1 FlexConfig 개체를 생성하여 자동 우회를 활성화합니다.

- a) **Objects(개체) > Object Management(개체 관리)**를 선택합니다.
- b) 목차에서 **FlexConfig > FlexConfig Object(FlexConfig 개체)**를 선택합니다.
- c) **Add FlexConfig Object(FlexConfig 개체 추가)**를 클릭하고, 다음 속성을 구성한 다음 **Save(저장)**를 클릭합니다.

- **Name(이름)** - 개체 이름입니다. 예를 들어, Enable_HW-Bypass를 입력합니다.

- **Deployment(구축) - Everytime(항상)**을 선택합니다. 모든 구축에 구성을 전송해 설정 상태를 유지합니다.

- **Type(유형)** - 기본값인 **Append(추가)**를 그대로 유지합니다. 명령은 직접 지원 기능용 명령이 전송된 후에 디바이스에 전송됩니다.

- **Object body(개체 본문)** - 개체 본문에서 자동 하드웨어 우회를 활성화하는 데 필요한 명령을 입력합니다. 예를 들어, 가능한 두 인터페이스 쌍에 필요한 명령은 다음과 같습니다.

```
hardware-bypass GigabitEthernet 1/1-1/2
hardware-bypass GigabitEthernet 1/3-1/4
```

개체 본문은 다음과 비슷해야 합니다.

단계 2 FlexConfig 정책을 생성하고 디바이스에 할당합니다.

- a) **Devices(디바이스) > FlexConfig**를 선택합니다.
- b) **New Policy(새 정책)**를 클릭하거나, 기존 FlexConfig 정책을 대상 디바이스에 할당해야 한다면(또는 이미 할당되어 있다면) 해당 정책을 수정합니다.

새 정책을 생성할 때는 정책 이름을 지정하는 대화 상자의 정책에 대상 디바이스를 할당합니다.

- c) 목차의 **User Defined(사용자 정의)** 폴더에서 하드웨어 우회 FlexConfig 개체를 선택하고 >을 클릭하여 정책에 추가합니다.

개체는 **Selected Appended FlexConfigs** 목록에 추가해야 합니다.

Selected Appended FlexConfigs	
#	Name
1	Enable_HW-Bypass

- d) **Save(저장)**를 클릭합니다.

- e) 아직 모든 대상 디바이스를 정책에 할당하지 않았다면 **Save(저장)** 아래에 있는 **Policy Assignments(정책 할당)** 링크를 클릭하여 할당합니다.
- f) **Preview Config(구성 미리보기)**를 클릭하고, **Preview(미리보기)** 대화 상자에서 할당된 디바이스 중 하나를 선택합니다.

시스템은 디바이스에 전송될 구성 CLI의 미리보기를 생성합니다. 하드웨어 우회 FlexConfig 개체에서 생성된 명령이 올바르게 표시되는지 확인합니다. 미리보기 끝부분에 표시됩니다. 관리 대상 기능에 적용한 다른 변경 사항에서 생성된 명령도 함께 표시됩니다. 하드웨어 우회 명령의 경우, 다음과 유사한 내용이 표시됩니다.

```
###Flex-config Appended CLI ###
hardware-bypass GigabitEthernet 1/1-1/2
hardware-bypass GigabitEthernet 1/3-1/4
```

단계 3 변경 사항을 배포합니다.

FlexConfig 정책을 디바이스에 할당했기 때문에 항상 구축 경고가 표시됩니다. FlexConfig는 주의해서 사용해야 한다는 뜻입니다. **Proceed(계속하기)**를 클릭하여 구축을 계속 진행합니다.

구축이 끝나면 구축 내역과 구축 기록을 확인할 수 있습니다. 이 기능은 구축이 실패했을 때 특히 유용합니다. [구축된 설정 확인, 34 페이지](#)의 내용을 참조하십시오.

다음에 수행할 작업

수동으로 하드웨어 우회를 호출하거나 직접 해제하려면 FlexConfig 개체를 두 개 생성해야 합니다.

- 하나는 수동으로 우회를 시작하는 명령인데, 두 쌍에 대해 우회를 호출할지에 따라 다음 명령 중 하나 또는 둘 다를 포함합니다.

```
hardware-bypass manual GigabitEthernet 1/1-1/2
hardware-bypass manual GigabitEthernet 1/3-1/4
```

- 다른 하나는 다음 명령 중 하나 또는 둘 모두를 포함해서 우회 기능을 수동으로 해제하는 명령입니다.

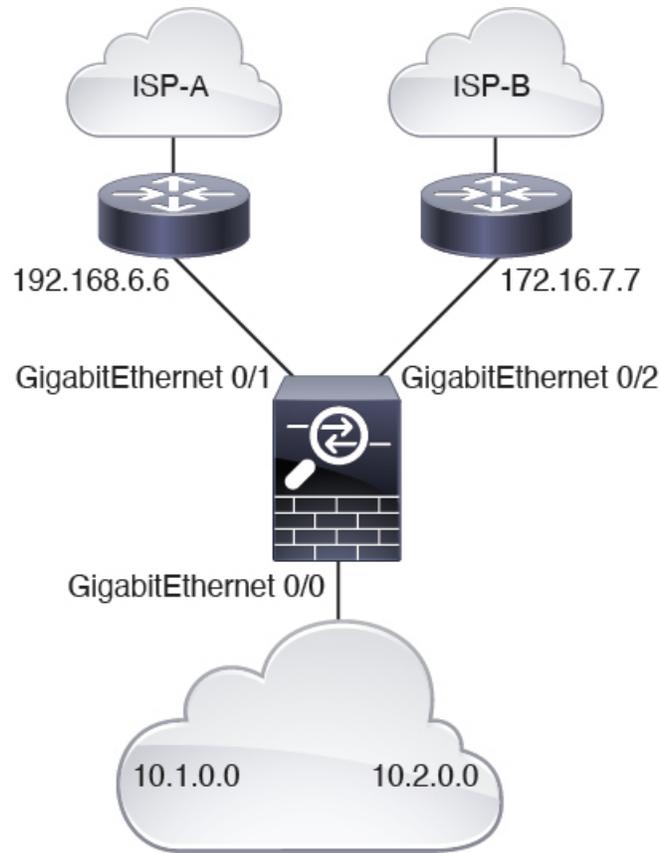
```
no hardware-bypass manual GigabitEthernet 1/1-1/2
no hardware-bypass manual GigabitEthernet 1/3-1/4
```

그리고 나서 우회를 켜거나 끄려면 FlexConfig 정책에 둘 중 하나의 개체를 추가하고 변경 사항을 구축해야 합니다. 또한 구축 후 FlexConfig 정책에서 개체를 즉시 제거해야 합니다. 수동으로 우회를 호출하는 경우, 프로세스를 반복하여 우회를 다시 해제해야 합니다. 따라서 이 수동 방법을 사용하려면 FlexConfig 정책 및 추가 구축을 자주 신중하게 편집해야 합니다.

정책 기반 라우팅 구성 방법

FlexConfig를 사용하여 PBR(정책 기반 라우팅) 기능을 구현할 수 있습니다. 예를 들어 다음 그림은 소스 IP 주소를 기반으로 네트워크 간의 트래픽 로드밸런싱을 보여줍니다. 이 경우 10.1.0.0/16 네트워크가 높은 우선 순위 트래픽을 생성하고, 이 트래픽은 ISP-A에 대한 더 높은 대역폭 링크를 통과해야 하

며 10.2.0.0/16은 더 낮은 우선 순위를 가져야 하며 -ISP-B에 대한 더 낮은 대역폭 링크를 통과해야 한다고 가정합니다.



시작하기 전에

이 절차에서는 다음과 같이 인터페이스를 이미 구성했다고 가정합니다.

- GigabitEthernet0/0.
 - 인터페이스 이름: inside
 - IP 주소: 10.1.1.1/24
 - 네트워크의 다른 라우터는 이 인터페이스를 10.1.0.0/16 및 10.2.0.0/16 주소 공간의 경로에 대한 게이트웨이로 사용합니다.
- GigabitEthernet0/1.
 - 인터페이스 이름: outside-1
 - IP 주소: 192.168.6.5/24
- GigabitEthernet0/2.
 - 인터페이스 이름: outside-2

- IP 주소: 172.16.7.6/24

프로시저

단계 1 10.1.0.0/16 및 10.2.0.0/16 주소 공간의 트래픽과 일치하도록 확장 ACL 개체를 생성합니다. 경로 맵의 트래픽에 다른 작업을 적용하므로 별도의 ACL을 생성해야 합니다.

- Objects(개체) > Object Management(개체 관리)**를 선택합니다.
- 목차에서 **Access List(액세스 목록) > Extended(확장)**를 선택합니다. 트래픽 소스 주소를 지정하려면 확장 액세스 목록을 구성해야 합니다.
- Add Extended Access List(확장된 액세스 목록 추가)** 버튼을 클릭합니다.
- 액세스 목록의 이름(예 : 높은 우선 순위)을 입력합니다.
- Add(추가)** 버튼을 클릭하고 우선 순위가 높은 주소 공간에 대한 규칙을 생성합니다. 주요 특징:
 - **Action(작업) - Allow(허용).**
 - **Source Networks(소스 네트워크)-**목록 아래의 수정 상자에 10.1.0.0/16을 입력하고 **Add(추가)**를 클릭합니다. 또는 이 네트워크 주소에 대한 네트워크 개체를 정의할 수 있습니다.
- 대화 상자의 하단에 있는 **Add(추가)**를 클릭합니다. 그러면 액세스 목록에 규칙이 추가됩니다.

Name

high-priority

Entries (1)

Add

Sequence	Action	Source	Source Port	Destination	Destination Port	
1	Allow	10.1.0.0/16	Any	Any	Any	

- Save(저장)**를 클릭합니다.
- 이 프로세스를 반복하여 다음 속성의 두 번째 액세스 목록을 생성합니다.
 - 이름-낮은 우선 순위
 - **Action(작업) - Allow(허용).**
 - **Source Networks(소스 네트워크)-**목록 아래의 수정 상자에 10.2.0.0/16을 입력하고 **Add(추가)**를 클릭합니다. 또는 이 네트워크 주소에 대한 네트워크 개체를 정의할 수 있습니다.

Name

low-priority

Entries (1)

Add

Sequence	Action	Source	Source Port	Destination	Destination Port	
1	➔ Allow	10.2.0.0/16	Any	Any	Any	 

단계 2 이러한 주소 공간에 대한 다음 홉 주소를 정의하는 경로 맵을 만듭니다.

- 개체 페이지에 있는 동안 목차에서 **Route Map**(경로 맵)을 클릭합니다.
- Add Route Map**(경로 맵 추가) 버튼을 클릭합니다.
- 개체의 이름(예: **load-balance**)을 입력합니다.
- Add**(추가)를 클릭하고 다음 특성의 우선 순위가 높은 트래픽에 대한 규칙을 생성합니다.

- 시퀀스 번호—10.

- 재배포—허용.

- **Match Clauses**(일치 절) > **IPv4** > **Address**(주소)-**Access List**(액세스 목록) 라디오 버튼을 선택한 다음 **Available Access Lists**(사용 가능한 액세스 목록) > **Extended**(확장)를 선택하고 우선 순위가 높은 ACL을 선택한 목록으로 이동합니다.

Sequence No:

10

Redistribution:

➔ Allow

Match Clauses

Set Clauses

Security Zones

IPv4

IPv6

BGP

Others

Address (2)

Next Hop (0)

Route Source (0)

Select addresses to match as access list or prefix list addresses of route.

 Access List Prefix List

Available Access Lists :

Extended

Available Extended Access List*

Q Search

high-priority

low-priority

Add

Selected Extended Access List

high-priority

- **Set Clauses**(절 설정) > **BGP Clauses Others**(BGP 절) > **Others**(기타)-**IPv4 Settings**(IPv4 설정) > **Next Hop**(다음 홉)에서 **Specific IP**(특정 IP)를 선택한 다음 ISP-A용 게이트웨이, **192.168.6.6**을 **Specific IP**(특정 IP) 수정 상자에 입력합니다.

Sequence No:
10

Redistribution:
Allow

Match Clauses **Set Clauses**

Metric Values AS Path Community List **Others**

BGP Clauses

Incomplete

IPv4 settings:

Next Hop:
Specific IP

Specific IP :
192.168.6.6
Use comma to separate multiple values

- e) 경로 맵에 규칙을 추가하려면 **Add(추가)**를 클릭합니다.
- f) **Add(추가)**를 클릭하고 다음 속성을 가진 낮은 우선 순위 트래픽에 대한 규칙을 생성합니다.
- 시퀀스 번호—**20**.
 - 재배포—허용.
 - **Match Clauses(일치 절) > IPv4 > Address(주소)-Access List(액세스 목록)** 라디오 버튼을 선택한 다음 **Available Access Lists(사용 가능한 액세스 목록) > Extended(확장)**를 선택하고 우선 순위가 낮은 ACL을 선택한 목록으로 이동합니다.
 - **Set Clauses(절 설정) > BGP Clauses Others(BGP 절) > Others(기타)-IPv4 Settings(IPv4 설정) > Next Hop(다음 홉)**에서 **Specific IP(특정 IP)**를 선택한 다음 ISP-B용 게이트웨이, **172.16.7.7**을 **Specific IP(특정 IP)** 수정 상자에 입력합니다.
- g) 경로 맵에 규칙을 추가하려면 **Add(추가)**를 클릭합니다.

Name
load-balance

▼ Entries (2)

Add

Sequence No ▲	Redistribution	
10	Allow	 
20	Allow	 

- h) **Save(저장)**를 클릭합니다.

단계 3 경로 맵을 사용하여 내부 인터페이스에서 PBR을 활성화하는 FlexConfig 개체를 생성합니다.

- a) 개체 페이지에 있는 동안 목차에서 **FlexConfig > FlexConfig** 개체를 클릭합니다.

- b) Policy_Based_Routing 개체를 찾은 다음 **Copy(복사)** ()를 클릭합니다.
- 이 개체는 시스템 정의 개체이지만 편집하기 전까지는 사용할 수 없습니다. 경로 맵의 이름으로 간단하게 업데이트할 수 있는 텍스트 개체를 가리키지는 않습니다. 이 시스템 정의 개체에 대해 항상 사용자 지정 개체를 생성해야 합니다.
- c) 복사 아이콘을 클릭하면 기본 이름이 Policy_Based_Routing_Copy인 새 개체가 포함된 대화 상자가 열립니다. 다음과 같이 기본 변경을 수행합니다.
- **Name(이름)**-의미 있는 이름을 입력합니다. 예를 들어 디바이스 FTD1에 대해 PBR을 구성하는 경우 **PBR_FTD1**일 수 있습니다.
 - **Description(설명)**-설명을 삭제하거나 용도에 맞게 의미를 부여합니다.
 - **Deployment(구축)** -한 번을 유지합니다.
 - **Type(유형)** = **Append(뒤에 추가)**를 유지합니다.
- d) 개체의 본문에는 다음 줄이 있습니다.

```
interface GigabitEthernet0/0
  policy-route route-map $r-map-object
```

"interface GigabitEthernet0/0" 행은 이 예에서 올바른 인터페이스를 구성하도록 이미 설정되어 있습니다. 다른 인터페이스에 PBR을 적용하려면 인터페이스 하드웨어 이름을 수정해야 합니다.

\$r-map-object 문자열은 사실 실제 변수가 아니며 아무 것도 가리키지 않습니다. 이 문자열을 교체해야 합니다.

- e) \$r-map-object 문자열을 삭제하고 "policy-route route-map" 줄(경로 맵 뒤 공백)에 커서를 둡니다.
- f) **Insert(삽입)** > **Insert Policy Object(정책 개체 삽입)** > **Route Map(경로 맵)**을 선택합니다.
- g) Route Map Variable(경로 맵 변수) 대화 상자에서 다음을 구성합니다.
- **Variable Name(변수 이름)**-**pbr-route-map**과 같은 이름입니다.
 - **Selected Object(선택한 개체)**-로드 밸런싱 경로 맵 개체를 사용 가능한 목록에서 선택한 목록으로 이동합니다.

Insert Route Map Variable

Variable Name:

Description:

Available Objects ↻

load-balance

Selected Object
load-balance 🗑

h) Variable(변수) 대화 상자에서 **Save**(저장)를 클릭합니다.

이제 FlexConfig 개체가 다음과 같이 표시됩니다. 여기서 변수는 대화 상자의 하단에 있는 변수 목록에 있습니다.

Add FlexConfig Object

Name:

Description:

▲ Copy-pasting any rich text might introduce line breaks while generating CLI. Please v

Insert ▼



Deployment:

Once ▼

```
interface GigabitEthernet0/0
  policy-route route-map $r-map-object
```

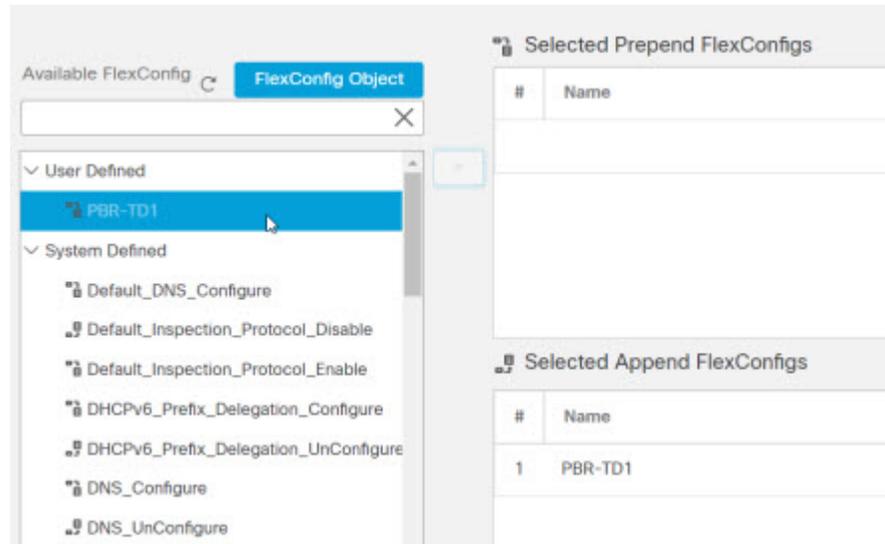
i) **Save**(저장)를 클릭합니다.

단계 4 디바이스에 할당된 FlexConfig 정책에 FlexConfig 개체를 추가합니다.

a) **Devices**(디바이스) > **FlexConfig**를 선택합니다.

b) 이 디바이스에 아직 FlexConfig 정책이 할당되지 않았다고 가정하고 **New Policy**(새 정책)를 클릭하고 정책에 이름을 지정한 다음 정책에 할당할 FTD1 디바이스를 선택하고 **Save**(저장)를 클릭합니다.

c) 사용 가능한 개체 목록의 **User Defined**(사용자 정의) 폴더 아래에서 개체를 찾은 다음, >을 클릭하여 선택한 개체 목록에 추가합니다.



- d) **Save**를 클릭하여 정책을 저장합니다.
- e) **Preview Config**(구성 미리보기)를 클릭하고, **Preview**(미리보기) 대화 상자에서 FTD1 디바이스를 선택합니다.

미리보기에는 FlexConfig 개체와 구성 명령을 사용하여 구현된 **management center** 매니지드 관리 구성 부분에서 생성된 CLI가 포함되어 있습니다. 이들은 섹션으로 구분됩니다. 이 예에서 수행한 작업을 기반으로 구성할 명령은 다음과 같습니다. 이 미리보기를 사용하여 예상한 결과를 얻고 있는지 확인할 수 있습니다.

```
###Flex-config Prepended CLI ###

###CLI generated from managed features ###
configure session OBJECT
object-group service ProxySG_ExtendedACL_4294969626
  service-object ip
object-group service ProxySG_ExtendedACL_4294969648
  service-object ip
commit noconfirm revert-save
configure session FMC_SESSION_1
access-list high-priority extended permit object-group
  ProxySG_ExtendedACL_4294969626 10.1.0.0 255.255.0.0 any
access-list low-priority extended permit object-group
  ProxySG_ExtendedACL_4294969648 10.2.0.0 255.255.0.0 any
commit noconfirm revert-save
route-map load-balance permit 10
  match ip address high-priority
  set ip next-hop 192.168.6.6
route-map load-balance permit 20
  match ip address low-priority
  set ip next-hop 172.16.7.7

###Flex-config Appended CLI ###
interface GigabitEthernet0/0
  policy-route route-map load-balance
```

f) **Close**(닫기)를 클릭하여 미리보기 대화 상자를 종료합니다.

다음에 수행할 작업

이제 디바이스에 구성을 구축할 수 있습니다.

번역에 관하여

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