Secure Firewall Threat Defense Virtual Auto Scale for Azure 구축

Azure의 Threat Defense Virtual용 Auto Scale 솔루션

Auto Scale 솔루션

위협 대응 가상 Auto Scale for Azure는 Azure에서 제공하는 서버리스 인프라(논리 앱, Azure 기능, 로드 밸런서, 보안 그룹, 가상 시스템 확장 집합 등)를 사용하며 완벽환 서버리스 방식으로 구현됩니다.

Azure용 위협 대응 가상 Auto Scale 구현의 몇 가지 주요 기능은 다음과 같습니다.

- ARM(Azure Resource Manager) 템플릿 기반 구축
- CPU 및 메모리(RAM) 기반의 메트릭 확장 지원:

참고 자세한 내용은 Auto Scale 논리, 37 페이지를 참조하십시오.

- 위협 대응 가상 구축 및 다중 가용성 영역 지원
- 완전 자동화된 threat defense virtual 인스턴스 등록 및 management center 등록 취소.
- 확장된 threat defense virtual 인스턴스에 자동으로 적용되는 NAT 정책, 액세스 정책 및 경로.
- 로드 밸런서 및 다중 가용성 영역 지원
- Auto Scale 기능 활성화 및 비활성화 지원
- management center에서만 작동하며 device manager는 지원하지 않습니다.
- PAYG 또는 BYOL 라이선싱 모드로 threat defense virtual 구축 지원 PAYG는 threat defense virtual 소프트웨어 버전 6.5 이상에만 적용됩니다. 지원되는 소프트웨어 플랫폼, 2 페이지의 내용을 참조하십시오.

- Cisco에서는 구축을 쉽게 수행할 수 있도록 Azure용 Auto Scale 구축 패키지를 제공합니다.

지원되는 소프트웨어 플랫폼

threat defense virtual Auto Scale 솔루션은 management center에서 관리하는 threat defense virtual에 적용 가능하며 소프트웨어 버전과 무관합니다. Cisco FirePOWER 호환성 가이드는 운영 체제 및 호스팅 환경 요구 사항을 포함해서 소프트웨어 및 하드웨어 호환성에 대한 내용을 제공합니다.

- Management Center의 가상 테이블은 management center virtual에 필요한 호환성 및 가상 호스팅 환경 요구 사항을 표시합니다.
- Threat Defense Virtual 호환성 테이블은 Azure의 threat defense virtual에 필요한 호환성 및 가상 호스팅 환경 요구 사항을 표시합니다.

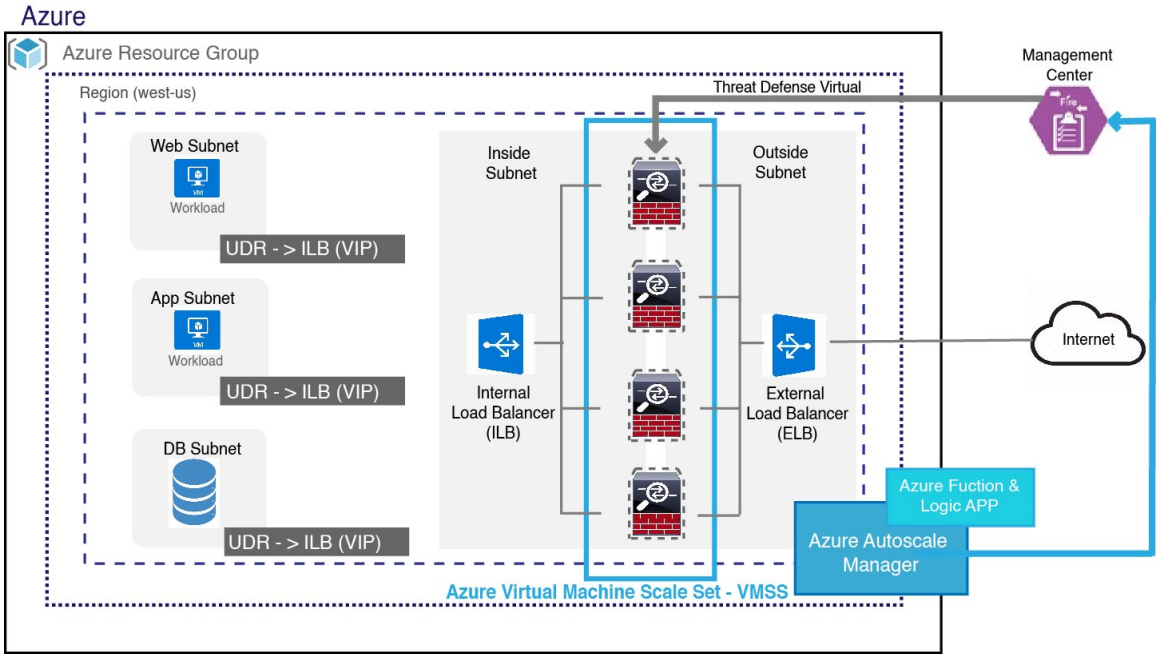
참고 Azure Auto Scale 솔루션 구축을 위해 Azure에서 threat defense virtual에 대해 지원되는 최소 버전은 버전 6.4입니다.

Auto Scale 사용 사례

Azure용 threat defense virtual Auto Scale은 Azure 내부 로드 밸런서(ILB)와 Azure 외부 로드 밸런서(ELB) 사이에 threat defense virtual 확장 집합을 배치하는 자동화된 수평 확장 솔루션입니다.

- ELB는 확장 집합에서 인터넷에서 threat defense virtual 인스턴스로 트래픽을 분산합니다. 그러면 방화벽이 애플리케이션에 트래픽을 전달합니다.
- ILB는 애플리케이션의 아웃 바운드 인터넷 트래픽을 확장 집합의 threat defense virtual 인스턴스로 분산합니다. 그러면 방화벽이 트래픽을 인터넷으로 전달합니다.
- 네트워크 패킷은 단일 연결에서 내부 및 외부 로드 밸런서를 모두 통과하지 않습니다.
- 확장 집합의 threat defense virtual 인스턴스 수는 로드 조건에 따라 자동으로 조정 및 구성됩니다.

그림 *1: Threat Defense Virtual Auto Scale* 사용 사례 다이어그램



범위

이 문서에서는 위협 대응 가상 Auto Scale for Azure 솔루션의 서버리스 구성 요소를 구축하는 자세한 절차를 설명합니다.

중요

- 구축을 시작하기 전에 전체 문서를 읽어보십시오.
- 구축을 시작하기 전에 전제 조건이 충족되었는지 확인합니다.
- 여기에 설명된 대로 단계 및 실행 순서를 따라야 합니다.

구축 패키지 다운로드

Azure용 위협 대응 가상 Auto Scale 솔루션은 Azure에서 제공하는 서버리스 인프라(논리 앱, Azure Functions, 로드 밸런서, 가상 시스템 확장 집합 등)를 활용하는 ARM(Azure Resource Manager) 템플릿 기반 구축입니다.

Azure용 위협 대응 가상 Auto Scale 솔루션을 시작하는 데 필요한 파일을 다운로드합니다. 사용자 버전의 구축 스크립트 및 템플릿은 GitHub 리포지토리에서 제공됩니다.

- GitHub Autoscale

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

ASM_Function.zip 패키지를 구축하는 방법에 대한 지침은 소스 코드로 Azure 기능 빌드, 40 페이지를 참고하십시오.

Auto Scale 솔루션 구성 요소

다음 구성 요소는 Azure용 위협 대응 가상 Auto Scale 솔루션을 구성합니다.

Azure Functions(Function 앱)

Function 앱은 Azure 함수의 집합입니다. 기본 기능은 다음과 같습니다.

- Azure 메트릭을 주기적으로 통신/프로브합니다.
- 위협 대응 가상 로드를 모니터링하고 축소/확장(Scale In/Scale Out 작업을 트리거합니다.
- 새 threat defense virtual를 management center에 등록합니다.
- management center를 통해 새 threat defense virtual를 구성합니다.
- management center에서 확장된 threat defense virtual을 등록 취소(제거)합니다.

이들 함수는 압축된 Zip 패키지 형식으로 제공됩니다(Azure Function 앱 패키지 빌드, 6 페이지 참조). 함수는 특정 작업을 수행하기 위해 가능한 한 개별적으로 유지되며, 개선 사항 및 새로운 릴리스 지원을 위해 필요에 따라 업그레이드할 수 있습니다.

오케스트레이터(Logic 앱)

Auto Scale Logic App은 하나의 워크플로우, 즉 시퀀스 단계 모음입니다. Azure 함수는 독립적인 엔터티이므로 서로 통신할 수 없습니다. 이 오케스트레이터는 이러한 함수의 실행을 시퀀싱하고 함수간 정보를 교환합니다.

- Logic App은 Auto Scale Azure 함수 간에 정보를 오케스트레이션하고 전달하는 데 사용됩니다.
- 각 단계는 Auto Scale Azure 함수 또는 기본 제공 표준 논리를 나타냅니다.
- Logic 앱은 JSON 파일로 제공됩니다.
- Logic 앱은 GUI 또는 JSON 파일을 통해 맞춤화할 수 있습니다.

VMSS(Virtual Machine Scale Set)

VMSS는 위협 대응 가상 디바이스와 같은 균일한 가상 시스템의 모음입니다.

- VMSS는 해당 집합에 동일한 새 VM을 추가할 수 있습니다.

- VMSS에 추가된 새 VM은 로드 밸런서, 보안 그룹 및 네트워크 인터페이스에 자동으로 연결됩니다.
- VMSS에는 Azure 위협 대응 가상용으로 사용하지 않도록 설정된 Auto Scale 기능이 내장되어 있습니다.
- VMSS에서 위협 대응 가상 인스턴스를 수동으로 추가하거나 삭제해서는 안 됩니다.

ARM(Azure Resource Manager) 템플릿

ARM 템플릿은 Azure용 위협 대응 가상 Auto Scale 솔루션에 필요한 리소스를 구축하는 데 사용됩니다.

ARM 템플릿은 다음을 포함하여 Auto Scale Manager 구성 요소에 대한 입력을 제공합니다.

- Azure Function 앱
- Azure Logic 앱
- VMSS(Virtual Machine Scale Set)
- 내부/외부 로드 밸런서
- 구축에 필요한 보안 그룹 및 기타 기타 구성 요소

중요 ARM 템플릿은 사용자 입력 검증과 관련하여 제한 사항이 있으므로 구축 중에 입력을 검증해야 합니다.

Auto Scale 솔루션 사전 요건

Azure 리소스

리소스 그룹

이 솔루션의 모든 구성 요소를 구축하려면 기존 또는 새로 생성 된 리소스 그룹이 필요합니다.

참고 나중에 사용할 수 있도록 리소스 그룹 이름, 리소스 그룹이 생성된 지역 및 Azure 구독 ID를 기록합니다.

네트워킹

가상 네트워크가 사용 가능/생성되었는지 확인합니다. Auto Scale 구축에서는 네트워킹 리소스를 생성, 변경 또는 관리하지 않습니다.

위협 대응 가상에는 4 개 네트워크 인터페이스가 필요하므로 가상 네트워크에는 4 개 서브넷이 필요합니다.

1. 관리 트래픽
2. 진단 트래픽
3. 내부 트래픽
4. 외부 트래픽

서브넷이 연결된 네트워크 보안 그룹에서 다음 포트를 열어야 합니다.

- SSH(TCP/22)

  로드 밸런서와 위협 대응 가상 사이의 상태 프로브에 필요합니다.

  서버리스 함수와 위협 대응 가상 간의 통신에 필요합니다.

- TCP/8305

  threat defense virtual와 management center 간 통신에 필요합니다.

- HTTPS(TCP/443)

  서버리스 구성 요소와 management center 간의 통신에 필요합니다.

- 애플리케이션별 프로토콜/포트

  모든 사용자 애플리케이션(예: TCP/80)에 필요합니다.

참고 가상 네트워크 이름, 가상 네트워크 CIDR, 4 개 서브넷의 이름, 외부 및 내부 서브넷의 게이트웨이 IP 주소를 기록합니다.

Azure Function 앱 패키지 빌드

위협 대응 가상 Auto Scale 솔루션은 아카이브 파일인 *ASM_Function.zip* 빌드가 필요하며, 이 파일은 압축된 ZIP 패키지로 개별 Azure 기능을 제공합니다.

ASM_Function.zip 패키지를 구축하는 방법에 대한 지침은 소스 코드로 Azure 기능 빌드, 40 페이지를 참고하십시오.

이들 함수는 특정 작업을 수행하기 위해 가능한 한 개별적이며, 개선 사항 및 새로운 릴리스 지원을 위해 필요에 따라 업그레이드할 수 있습니다.

Management Center 준비

모든 기능을 갖춘 멀티 디바이스 관리자인 management center를 사용해 threat defense virtual을 관리할 수 있습니다. threat defense virtual은 threat defense virtual 장비에 할당한 관리 인터페이스의 management center로 등록 및 통신합니다.

디바이스 그룹을 포함해 threat defense virtual 설정 및 관리를 위해 필요한 모든 객체를 생성하면 여러 디바이스에 정책을 쉽게 배포하고 업데이트를 설치할 수 있습니다. 디바이스 그룹에 적용되는 모든 설정은 threat defense virtual 인스턴스에 푸시됩니다.

다음 섹션에서는 management center 준비를 위한 기본 단계를 간략히 소개합니다. 자세한 내용은 Firepower Management Center Configuration Guide를 참조하세요. management center를 준비할 때는 다음 정보를 기록하십시오.

- management center 공용 IP 주소.
- management center 사용자 이름/비밀번호.
- 보안 정책 이름
- 내부 및 외부 보안 영역 개체 이름
- 디바이스 그룹 이름

새 **Management Center** 사용자 생성

AutoScale Manager에서만 사용할 관리자 권한이 있는 management center의 새 사용자를 생성합니다.

중요 다른 management center 세션과의 충돌을 방지하려면 threat defense virtual Auto Scale 솔루션 전용 management center 사용자 계정이 있어야 합니다.

프로시저

단계 **1** management center에서 관리자 권한으로 새 사용자를 생성합니다. **System(**시스템**)** > **Users(**사용자**)**를 선택하고 **Create User(**사용자 생성**)**를 클릭합니다.

사용자 이름은 다음과 같은 Linux 기준을 준수해야 합니다.

- 최대 32개의 영숫자 문자와 하이픈(-) 및 밑줄(_)
- 모두 소문자
- 하이픈(-)으로 시작할 수 없으며, 숫자만으로 구성할 수 없고, 마침표(.), 단가 기호(@) 또는 슬래시(/)를 포함할 수 없음

단계 **2** 환경에 필요한 대로 사용자 옵션을 완료합니다. 자세한 내용은 Cisco Secure Firewall Management Center 관리 가이드를 참조하십시오.

액세스 제어 구성

액세스 제어를 내부에서 외부로 향하는 트래픽을 허용하도록 구성합니다. 액세스 제어 정책 내에서 액세스 제어 규칙은 여러 매니지드 디바이스에서 네트워크 트래픽을 처리하는 세분화된 방법을 제

공합니다. 효과적인 구축을 위해서는 규칙을 올바르게 구성하고 그 순서를 지정해야 합니다. Cisco Secure Firewall Management Center 디바이스 구성 가이드의 "액세스 제어 모범 사례"를 참고하십시오.

프로시저

단계 **1** **Policies(**정책**)** > **Access Control(**액세스 제어**)**을 선택합니다.

단계 **2** **New Policy(**새로운 정책**)**를 클릭합니다.

단계 **3** 고유한 **Name(**이름**)**을 입력하고, 필요한 경우 **Description(**설명**)**을 입력합니다.

단계 **4** 고급 보안 설정 및 규칙을 구성하려면 Cisco Secure Firewall Management Center 디바이스 구성 가이드를 참고하십시오.

라이선싱 구성

모든 라이선스는 management center를 통해 threat defense에 제공됩니다. 선택적으로 다음 기능 라이선스를 구매할 수 있습니다.

- **Secure Firewall Threat Defense IPS** - 보안 인텔리전스 및 Cisco Secure IPS
- **Secure Firewall Threat Defense Malware Defense** - 악성코드 방어
- **Secure Firewall Threat Defense URL** 필터링 - 필터링
- **RA VPN**—AnyConnect Plus, AnyConnect Apex 또는 AnyConnect VPN 전용입니다.

참고 IPS, 악성코드 디펜스 또는 URL 필터링 라이선스를 구매하고 1년, 3년 또는 5년 동안 업데이트에 액세스하려면 그에 대응하는 구독 라이선스도 필요합니다.

시작하기 전에

- Cisco Smart Software Manager에서 마스터 계정을 만듭니다.

  아직 어카운트가 없는 경우 새 어카운트 설정 링크를 클릭합니다. Smart Software Manager에서 조직의 마스터 계정을 만들 수 있습니다.

- Cisco Smart Software Licensing 계정은 일부 기능(export-compliance 플래그를 사용하여 활성화됨)을 사용하려면 강력한 암호화(3DES/AES) 라이선스 자격을 얻어야 합니다.

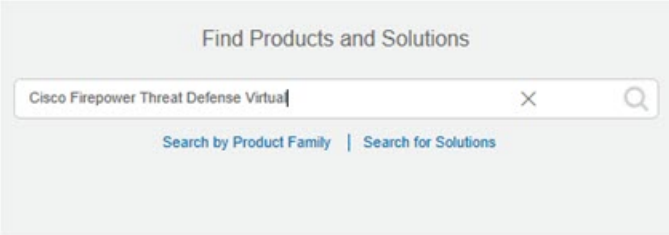
프로시저

단계 **1** Smart Licensing 계정에서 필요한 라이선스가 사용 가능한지 확인합니다.

Cisco 또는 리셀러에서 디바이스를 구매한 경우 라이선스는 Smart Software License 계정에 연결되어 있어야 합니다. 그러나 라이선스를 직접 추가해야 하는 경우 Cisco Commerce Workspace에서 **Find**

Products and Solutions(제품 및 솔루션 찾기**)** 검색 필드를 사용합니다. 다음 라이선스 PID를 검색합니다.

그림 *2:* 라이선스 검색



참고 PID를 찾을 수 없는 경우 주문에 수동으로 PID를 추가할 수 있습니다.

단계 **2** 아직 등록하지 않은 경우 management center을 Smart Licensing 서버에 등록합니다.

등록하려면 Smart Software Manager에서 등록 토큰을 생성해야 합니다. 자세한 지침은 Cisco Secure Firewall Management Center 관리 가이드 항목을 참조하십시오.

보안 영역 개체 생성

구축을 위해 내부 및 외부 보안 영역 개체를 생성합니다.

프로시저

단계 **1** **Objects(**개체**)** > **Object Management(**개체 관리**)**를 선택합니다.
단계 **2** 개체 유형 목록에서 **Interface(**인터페이스**)**를 선택합니다.
단계 **3** **Add** > **Security Zone(**보안 영역 추가**)**을 클릭합니다.
단계 **4** 이름을 입력합니다(예: *inside*, *outside*).
단계 **5** 인터페이스 유형으로 라우팅을 선택합니다.
단계 **6** **Save(**저장**)**를 클릭합니다.

디바이스 그룹 생성

디바이스 그룹을 사용하면 쉽게 정책을 할당하고 여러 디바이스에 업데이트를 설치할 수 있습니다.

프로시저

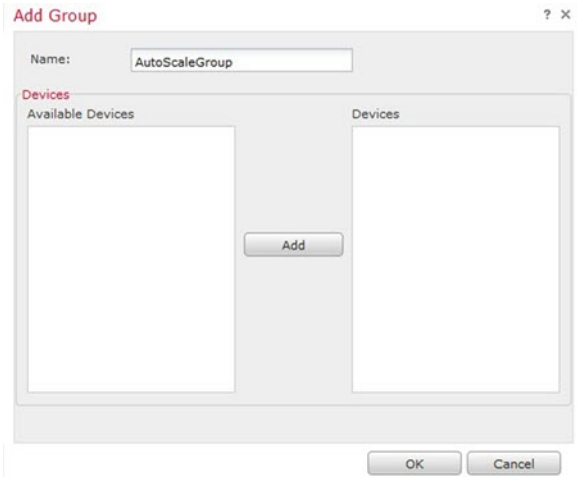
단계 **1** **Devices(**디바이스**)** > **Device Management(**디바이스 관리**)**를 선택합니다.

그림 *3:* 디바이스 관리

단계 **2** 드롭다운 메뉴의 **Add(**추가**)**에서 **Add Group(**그룹 추가**)**를 선택합니다.

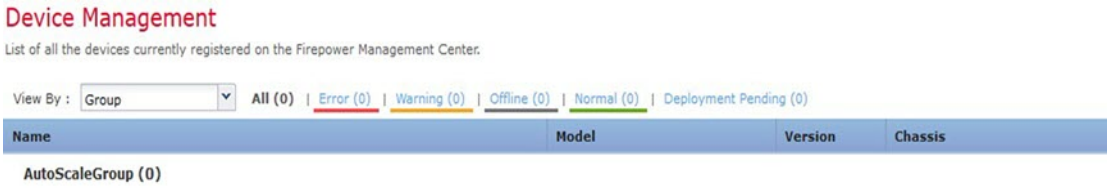
단계 **3** **Name(**이름**)**을 입력합니다. 예를 들면 *AutoScaleGroup*.

그림 *4:* 디바이스 그룹 추가



단계 **4** 디바이스 그룹에 추가하려면 **OK(**확인**)**를 클릭합니다.

그림 *5:* 디바이스 그룹 추가됨



보안 쉘 액세스 구성

threat defense 디바이스의 플랫폼 설정은 값을 여러 디바이스 간에 공유하려고 할 수 있는 비 관련 기능을 구성합니다. Threat Defense Virtual Auto Scale for Azure이 내부/외부 영역의 SSH 및 Auto Scale 그룹에 대해 생성된 디바이스 그룹을 허용하려면 threat defense 플랫폼 설정 정책이 필요합니다. 이는 threat defense virtual의 데이터 인터페이스가 로드 밸런서에서 상태 프로브에 응답할 수 있도록 하는 데 필요합니다.

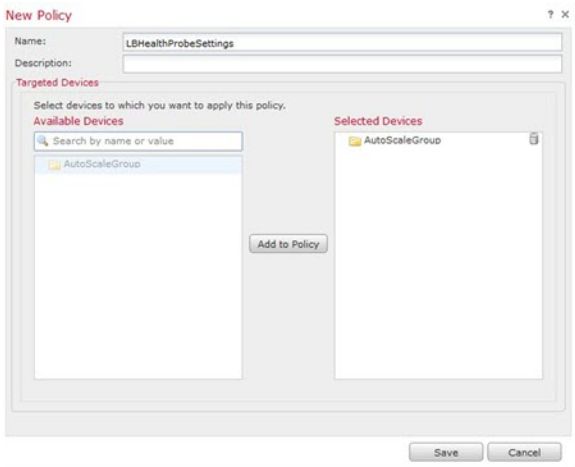
시작하기 전에

- 디바이스에 SSH 연결을 허용할 호스트 또는 네트워크를 정의하는 네트워크 개체가 필요합니다. 이 절차의 일부로 개체를 추가할 수 있지만 개체 그룹을 사용하여 IP 주소 그룹을 식별하려면 규칙에 필요한 그룹이 이미 있는지 확인합니다. **Objects(**개체**)** > **Object Management(**개체 관리**)**를 선택하여 개체를 설정합니다. 예를 들어 다음 절차의 *azure-utility-ip(168.63.129.16)* 개체를 참조하십시오.

프로시저

단계 **1** **Devices(**디바이스**)** > **Platform Settings(**플랫폼 설정**)**를 선택하고 threat defense 정책을(예: *LBHealthProbeSettings*) 생성하거나 수정합니다.

그림 ***6: Threat Defense*** 플랫폼 설정 정책



단계 **2** **Secure Shell**을 선택합니다.

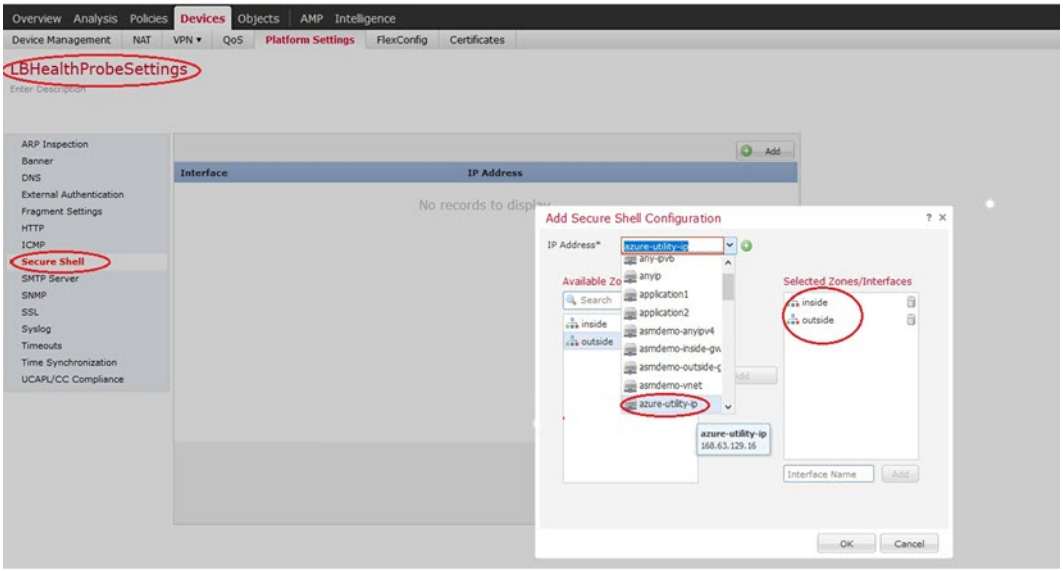
단계 **3** SSH 연결을 허용하는 인터페이스와 IP 주소를 확인합니다.

a) **Add(**추가**)**를 클릭해 새 규칙을 추가하거나, **Edit(**편집**)**을 클릭해 기존 규칙을 편집합니다.

b) 규칙 속성을 구성합니다.

- **IP Address(IP** 주소**)** - SSH 연결을 허용하는 호스트 또는 네트워크를 식별하는 네트워크 개체입니다(예: *azure-utility-ip (168.63.129.16)*). 드롭다운 메뉴에서 개체를 선택하거나 +를 클릭하여 새 네트워크 개체를 추가합니다.
- **Security Zones(**보안 영역**)** - SSH 연결을 허용할 인터페이스가 포함된 영역을 추가합니다. 예를 들어 내부 인터페이스는 내부 영역에, 외부 인터페이스는 외부 영역에 할당할 수 있습니다. management center의 개체 페이지에서 보안 영역을 생성할 수 있습니다. 보안 영역에 대한 자세한 내용은 Cisco Secure Firewall Management Center 디바이스 구성 가이드를 참조하십시오.
- **OK(**확인**)**를 클릭합니다.

그림 *7: Threat Defense Virtual Auto Scale*을 위한 *SSH* 액세스



단계 **4** **Save(**저장**)**를 클릭합니다.

이제 **Deploy(**구축**)** > **Deployment(**구축**)**로 이동하여 할당된 디바이스에 정책을 구축할 수 있습니다. 변경사항은 구축할 때까지 활성화되지 않습니다.

NAT 구성

NAT 정책을 생성하고 외부 인터페이스에서 애플리케이션으로 트래픽을 전달하는 데 필요한 NAT 규칙을 생성하고 이 정책을 자동 확장을 위해 생성한 디바이스 그룹에 연결합니다.

프로시저

단계 **1** **Devices(**디바이스**)** > **NAT**를 선택합니다.

단계 **2** **New Policy(**새 정책**)** 드롭다운 목록에서 **Threat Defense NAT**를 선택합니다.

단계 **3** 고유한 **Name(**이름**)**을 입력합니다.

단계 **4** 필요한 경우 **Description(**설명**)**을 입력합니다.

단계 **5** NAT 규칙을 구성합니다. NAT 규칙을 생성하고 NAT 정책을 적용하는 방법에 대한 지침은 Cisco Secure Firewall Management Center 디바이스 구성 가이드의 "Configure NAT for Threat Defense" 절차를 참조하십시오. 다음 그림에는 기본 접근 방식이 나와 있습니다.

그림 ***8: NAT*** 정책 예

참고 따라서 변환 문제와 까다로운 트러블슈팅 상황을 방지하기 위해 규칙을 최대한 단순하게 유지하는 것이 좋습니다. 그리고 NAT를 구현하기 전에 면밀한 계획을 세워야 합니다.

단계 **6** **Save(**저장**)**를 클릭합니다.

입력 매개변수

다음 표에서는 템플릿 매개 변수를 정의하고 일 예를 제공합니다. 이러한 값을 결정하고 나면 Azure 구독에 ARM 템플릿을 구축할 때 이러한 매개 변수를 사용하여 위협 대응 가상 디바이스를 생성할 수 있습니다. Auto Scale ARM 템플릿 구축, 21 페이지의 내용을 참조하십시오.

표 ***1:*** 템플릿 매개변수

매개 변수 이름	허용되는 값 / 유형	설명	리소스 생성 유형
resourceNamePrefix	문자열 * (3 ~ 10 자)	모든 리소스는 이 접두사를 포함하는 이름으로 생성됩니다. 참고: 소문자만 사용하십시오. 예: ftdv	New
virtualNetworkRg	문자열	가상 네트워크 리소스 그룹 이름입니다. 예: cisco-virtualnet-rg	기존
virtualNetworkName	문자열	가상 네트워크 이름(이미 생성됨) 예: cisco-virtualnet	기존
virtualNetworkCidr	CIDR 형식 x.x.x.x/y	가상 네트워크의 CIDR(이미 생성됨)	기존
mgmtSubnet	문자열	관리 서브넷 이름(이미 생성됨) 예: cisco-mgmt-subnet	기존

매개 변수 이름	허용되는 값 / 유형	설명	리소스 생성 유형
diagSubnet	문자열	진단 서브넷 이름(이미 생성됨) 예: cisco-diag-subnet	기존
insideSubnet	문자열	내부 서브넷 이름(이미 생성됨) 예: cisco-inside-subnet	기존
internalLbIp	문자열	내부 서브넷(이미 생성됨)의 내부 로드 밸런서 IP 주소입니다. 예: 1.2.3.4.	기존
insideNetworkGatewayIp	문자열	내부 서브넷 게이트웨이 IP 주소(이미 생성됨)	기존
outsideSubnet	문자열	외부 서브넷 이름(이미 생성됨) 예: cisco-outside-subnet	기존
outsideNetworkGatewayIp	문자열	외부 서브넷 게이트웨이 IP(이미 생성됨)	기존
deviceGroupName	문자열	management center의 디바이스 그룹(이미 생성됨)	기존
insideZoneName	문자열	management center의 내부 영역 이름(이미 생성됨)	기존
outsideZoneName	문자열	management center의 외부 영역 이름(이미 생성됨)	기존
softwareVersion	문자열	위협 대응 가상 버전(구축 중 드롭 다운에서 선택)	기존
vmSize	문자열	위협 대응 가상 인스턴스의 크기(구축 중 드롭 다운에서 선택).	해당 없음
ftdLicensingSku	문자열	Threat Defense Virtual 라이선싱 모드(PAYG / BYOL) 참고: PAYG는 버전 6.5 이상에서 지원됩니다.	해당 없음
licenseCapability	쉼표로 구분된 문자열	기본, 악성코드, URL 필터링, 위협	해당 없음

매개 변수 이름	허용되는 값 / 유형	설명	리소스 생성 유형
ftdVmManagementUserName	문자열*	threat defense virtual VM 관리 관리자 사용자 이름입니다. 이는 관리자가 될 수 없습니다. Azure for VM administrator user name guidelines를 참조하십시오.	New
ftdVmManagementUserPassword	문자열*	threat defense virtual VM 관리 관리자 사용자의 비밀번호입니다. 비밀번호는 12 ~ 72자여야 하며 소문자, 대문자, 숫자 및 특수 문자를 포함해야 합니다. 같은 문자를 세 번 이상 반복해서 사용할 수 없습니다. 참고 템플릿에는 이에 대한 규정 준수 확인이 없습니다.	New
fmcIpAddress	문자열 x.x.x.x	management center의 공용 IP 주소(이미 생성됨)	기존
fmcUserName	문자열	관리자 권한이 있는 Management Center 사용자 이름(이미 생성됨)	기존
fmcPassword	문자열	위의 management center 사용자 이름에 대한 Management Center 비밀번호(이미 생성됨)	기존
policyName	문자열	management center에서 생성된 보안 정책(이미 생성됨)	기존

매개 변수 이름	허용되는 값 / 유형	설명	리소스 생성 유형
scalingPolicy	POLICY-1 / POLICY-2	**POLICY-1**: 어떤 위협 대응 가상 평균로드가 구성된 기간 동안 확장 임계값을 초과하면 확장(Scale-Out)이 트리거됩니다. **POLICY-2**: 자동 확장 그룹 내의 모든 위협 대응 가상 평균로드가 구성된 기간 동안 확장 임계값을 초과하면 확장(Scale-Out)이 트리거됩니다. 두 경우 모두 축소(Scale-In) 논리는 동일하게 유지됩니다. 모든 위협 대응 가상 디바이스의 평균로드가 구성된 기간 동안 축소 임계값 미만이 되면 축소가 트리거됩니다.	해당 없음
scalingMetricsList	문자열	스케일링 결정을 내리는 데 사용되는 메트릭입니다. 허용됨: CPU CPU, 메모리 기본값: CPU	해당 없음
cpuScaleInThreshold	문자열	CPU 메트릭에 대한 축소 임계값입니다. 기본값: 10 위협 대응 가상 메트릭이 이 값보다 작으면 축소(Scale-In)가 트리거됩니다. Auto Scale 논리, 37 페이지의 내용을 참조하십시오.	해당 없음

매개 변수 이름	허용되는 값 / 유형	설명	리소스 생성 유형
cpuScaleOutThreshold	문자열	CPU 메트릭의 확장 임계값입니다. 기본값: 80 메트릭이이 값을 초과하면 스케일 아웃이 트리거됩니다.위협 대응 가상 'cpuScaleOutThreshold'는 항상 'cpuScaleInThreshold' 보다 커야 합니다. Auto Scale 논리, 37 페이지의 내용을 참조하십시오.	해당 없음
memoryScaleInThreshold	문자열	메모리 메트릭에 대한 축소(Scale-In) 임계값(%)입니다. 기본값: 0 위협 대응 가상 메트릭이 이 값보다 작으면 축소(Scale-In)가 트리거됩니다. Auto Scale 논리, 37 페이지의 내용을 참조하십시오.	해당 없음
memoryScaleOutThreshold	문자열	메모리 메트릭에 대한 확장(Scale-Out) 임계값(%)입니다. 기본값: 0 위협 대응 가상 메트릭이 이 값을 초과하면 확장(Scale-Out)이 트리거됩니다. 'memoryScaleOutThreshold'는 항상'memoryScaleInThreshold'보다 커야 합니다. Auto Scale 논리, 37 페이지의 내용을 참조하십시오.	해당 없음
minFtdCount	정수	지정된 시간에 설정된 확장 집합에서 사용 가능한 최소 위협 대응 가상 인스턴스. 예: 2	해당 없음

매개 변수 이름	허용되는 값 / 유형	설명	리소스 생성 유형
maxFtdCount	정수	확장 집합에서 허용되는 최대 위협 대응 가상 인스턴스 수입니다. 예: 10 참고 이 수는 management center 용량에 의해 제한됩니다. Auto Scale 논리는 이 변수의 범위를 확인하지 않으므로 신중하게 입력하십시오.	해당 없음
metricsAverageDuration	정수	드롭다운에서 선택 이 숫자는 메트릭이 평균화되는 시간(분)을 나타냅니다. 이 변수의 값이 5(즉, 5)인 경우, Auto Scale Manager가 예약되면 메트릭의 지난 5분 평균을 확인하고 이를 기반으로 하여 확장 결정을 내립니다. 참고 Azure 제한으로 인해 숫자 1, 5, 15, 30만 유효합니다.	해당 없음

매개 변수 이름	허용되는 값 / 유형	설명	리소스 생성 유형
initDeploymentMode	일괄(BULK) / 단계별(STEP)		

매개 변수 이름	허용되는 값 / 유형	설명	리소스 생성 유형
		기본적으로 첫 번째 구축 또는 확장 집합에 위협 대응 가상 인스턴스가 포함되지 않은 경우에 적용됩니다. 일괄(BULK): Aucto Scale Manager가 한 번에 'minFtdCount '개의 위협 대응 가상 인스턴스 수를 동시에 구축하려고 시도합니다. 참고 실행은 동시에 진행되지만 management center에 등록하는 것은 management center 제한으로 인해 순차적입니다. 단계별(STEP): Auto Scale Manager는 예약된 간격마다 하나씩 'minFtdCount '개의 위협 대응 가상 디바이스를 구축합니다. 참고 단계별 옵션은 'minFtdCount' 인스턴스가 management center와 함께 시작 및 구성되고 작동 상태가 되지만 디버깅에 유용할 때까지 시간이 오래 걸립니다. 일괄 옵션은 하나의 threat defense virtual 실행이 병렬로 실행되기 때문에 threat defense virtual의 'minFtdCount'개 전부를 시작하는 데 동일한 시간이 걸리지만 management center 등록은 순차적입니다. threat defense virtual의 'minFtdCount'개를 구축	

매개 변수 이름	허용되는 값 / 유형	설명	리소스 생성 유형
		하는 데 걸리는 총 시간 = (1개 threat defense virtual를 실행하는 시간 + 1개 threat defense virtual를 등록/구성하는 시간 * minFtdCount).	
* Azure에는 새 리소스의 명명 규칙에 제한 사항이 있습니다. 제한 사항을 검토하거나 간단히 모두 소문자를 사용하십시오. 공백이나 특수 문자는 사용하지 마십시오.			

Auto Scale 구축

구축 패키지 다운로드

Azure용 위협 대응 가상 Auto Scale 솔루션은 Azure에서 제공하는 서버리스 인프라(논리 앱, Azure Functions, 로드 밸런서, 가상 시스템 확장 집합 등)를 활용하는 ARM(Azure Resource Manager) 템플릿 기반 구축입니다.

Azure용 위협 대응 가상 Auto Scale 솔루션을 시작하는 데 필요한 파일을 다운로드합니다. 사용자 버전의 구축 스크립트 및 템플릿은 GitHub 리포지토리에서 제공됩니다.

- GitHub Autoscale

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

ASM_Function.zip 패키지를 구축하는 방법에 대한 지침은 소스 코드로 Azure 기능 빌드, 40 페이지를 참고하십시오.

Auto Scale ARM 템플릿 구축

ARM 템플릿은 위협 대응 가상 Auto Scale for Azure에 필요한 리소스를 구축하는 데 사용됩니다. 지정된 리소스 그룹 내에서 ARM 템플릿 구축은 다음을 생성합니다.

- VMSS(Virtual Machine Scale Set)
- 외부 로드 밸런서
- 내부 로드 밸런서
- Azure Function 앱

- Logic 앱
- 보안 그룹 (데이터 및 관리 인터페이스용)

시작하기 전에

- GitHub 리포지토리(https://github.com/CiscoDevNet/cisco-ftdv/tree/master/autoscale/azure)에서 ARM 템플릿 *azure_ ftdv _autoscale.json*을 다운로드합니다.

프로시저

단계 **1** 여러 Azure 영역에서 위협 대응 가상 인스턴스를 구축해야 하는 경우 구축 영역에서 사용 가능한 영역을 기준으로 하여 ARM 템플릿을 편집합니다.

예제:

```
"zones": [
    "1",
    "2",
    "3"
],
```

이 예에서는 3개의 영역이 있는“Central US” 지역을 보여줍니다.

단계 **2** 외부 로드 밸런서에 필요한 트래픽 규칙을 수정합니다. 이 'json' 어레이를 확장하여 원하는 수의 규칙을 추가할 수 있습니다.

예제:

```
{
 "type": "Microsoft.Network/loadBalancers",
 "name": "[variables('elbName')]",
 "location": "[resourceGroup().location]",
 "apiVersion": "2018-06-01",
 "sku": {
   "name": "Standard"
 },
 "dependsOn": [
   "[concat('Microsoft.Network/publicIPAddresses/', variables('elbPublicIpName'))]"

 ],
 "properties": {
   "frontendIPConfigurations": [
     {
       "name": "LoadBalancerFrontEnd",
         "properties": {
           "publicIPAddress": {
             "id": "[resourceId('Microsoft.Network/publicIPAddresses/',
variables('elbPublicIpName'))]"
           }
         }
     }
   ],
   "backendAddressPools": [
     {
```

```
            "name": "backendPool"
          }
        ],
        "loadBalancingRules": [
          {
            "properties": {
              "frontendIPConfiguration": {
                "Id": "[concat(resourceId('Microsoft.Network/loadBalancers',
variables('elbName')), '/frontendIpConfigurations/LoadBalancerFrontend')]"
              },
              "backendAddressPool": {
                "Id": "[concat(resourceId('Microsoft.Network/loadBalancers',
variables('elbName')), '/backendAddressPools/BackendPool')]"
              },
              "probe": {
                "Id": "[concat(resourceId('Microsoft.Network/loadBalancers',
variables('elbName')), '/probes/lbprobe')]"
              },
              "protocol": "TCP",
              "frontendPort": "80",
              "backendPort": "80",
              "idleTimeoutInMinutes": "[variables('idleTimeoutInMinutes')]"
            },
            "Name": "lbrule"
          }
        ],
```

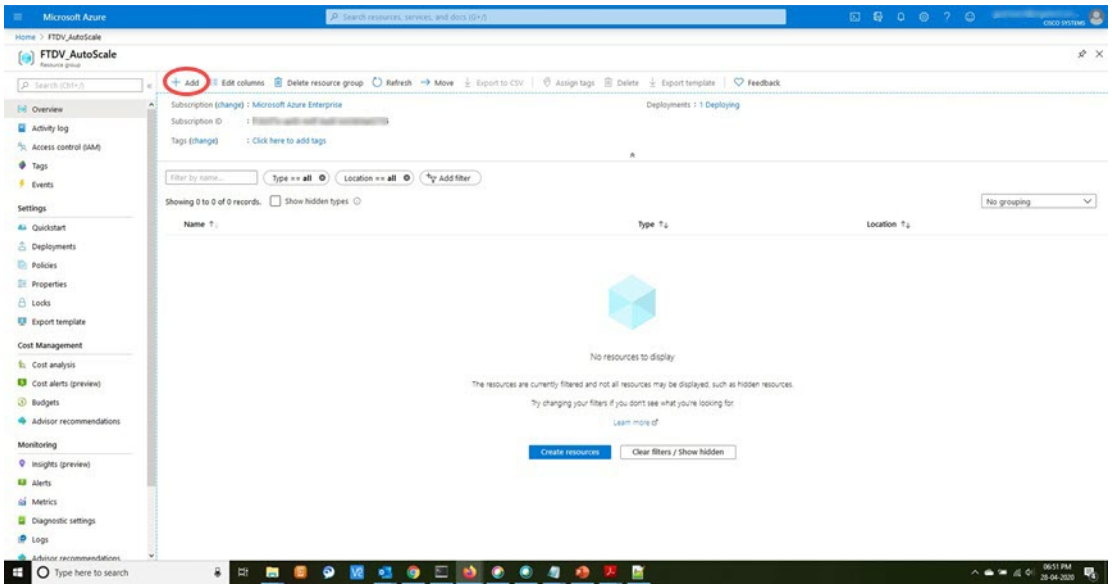
참고 이 파일을 편집하지 않으려는 경우 Azure 포털 구축후(post-deployment)에서 이를 편집할 수도 있습니다.

단계 **3** Microsoft 계정 사용자 이름 및 비밀번호를 사용하여 Microsoft Azure 포털에 로그인합니다.

단계 **4** 서비스 메뉴에서 **Resource groups(**리소스 그룹**)**를 클릭하여 리소스 그룹 블레이드에 액세스합니다. 블레이드에 나열된 구독의 모든 리소스 그룹이 표시됩니다.

새 리소스 그룹을 생성하거나 기존의 빈 리소스 그룹을 선택합니다(예:위협 대응 가상 *_AutoScale*).

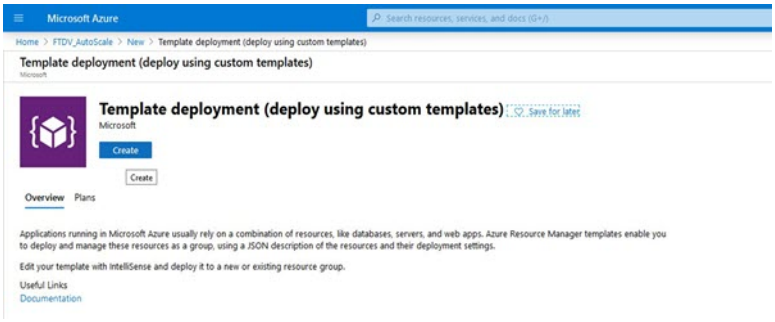
그림 ***9: Azure Portal***



단계 **5** **Create a resource(**리소스 생성**)(+)**를 클릭하여 템플릿 구축을 위한 새 리소스를 생성합니다. Create Resource Group(리소스 그룹 생성) 블레이드가 나타납니다.

단계 **6** **Search the Marketplace(**마켓플레이스 검색**)**에서 **Template deployment(**구축 **(**맞춤형 템플릿 사용**))**를 입력한 다음 **Enter** 키를 누릅니다.

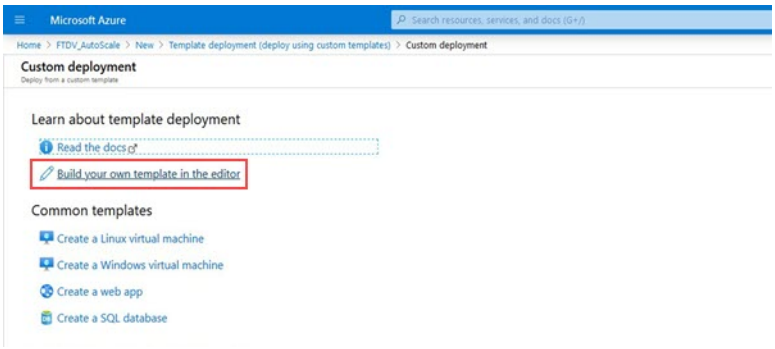
그림 ***10:*** 맞춤형 템플릿 구축



단계 **7** **Create(**생성**)**를 클릭합니다.

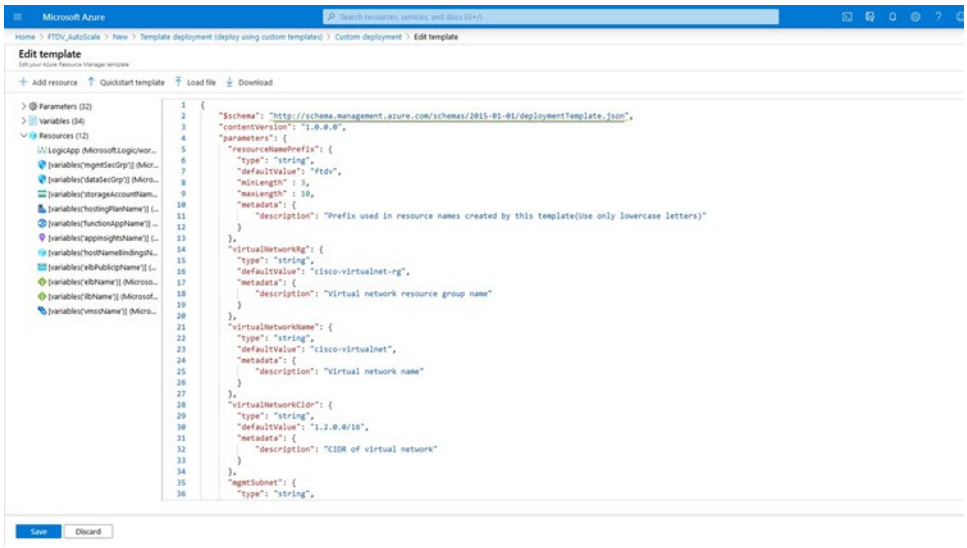
단계 **8** 템플릿을 생성하기 위한 몇 가지 옵션이 있습니다. **Build your own template in editor(**편집기에서 자체 템플릿 구축**)**를 선택합니다.

그림 ***11:*** 자체 템플릿 만들기



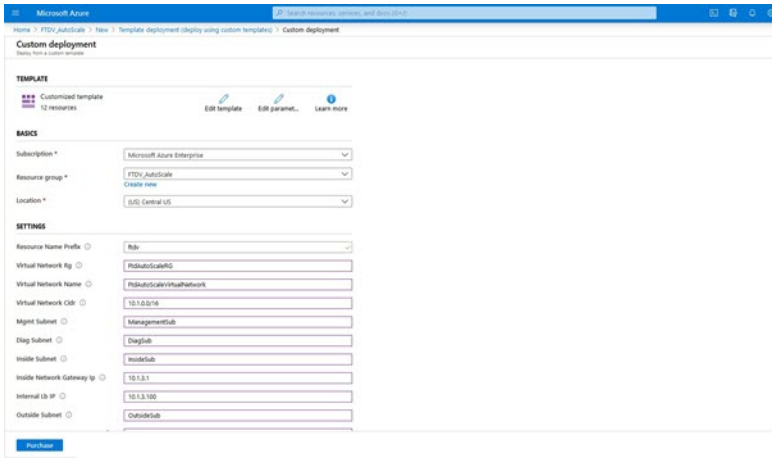
단계 **9** **Edit template(**템플릿 편집**)** 창에서 모든 기본 콘텐츠를 삭제하고 업데이트된 *azure_ftdv_autoscale.json* 에서 콘텐츠를 복사하고 **Save(**저장**)**를 클릭합니다.

그림 *12:* 템플릿 수정



단계 **10** 다음 섹션에서 모든 매개변수를 입력합니다. 각 매개변수에 대한 자세한 내용은 입력 매개변수, 13 페이지를 참조한 다음 **Purchase(**구매**)**를 클릭하십시오.

그림 *13: ARM* 템플릿 매개변수

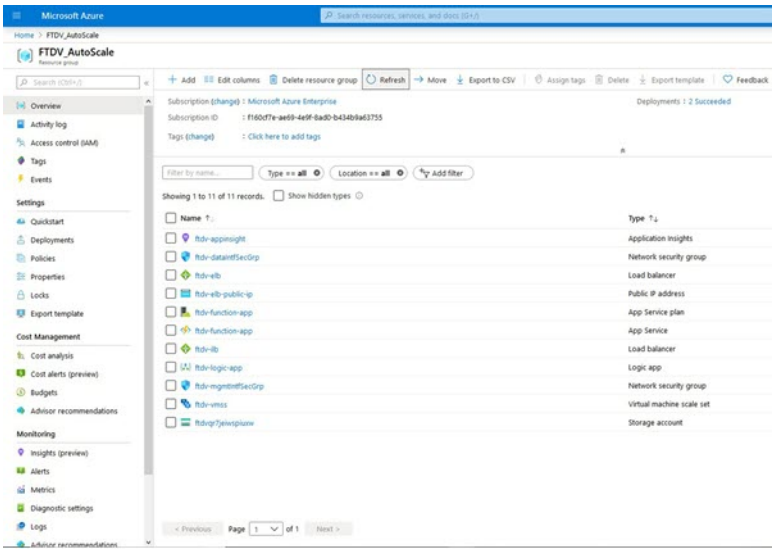


참고 **Edit Parameters(**매개변수 편집**)**를 클릭하고 JSON 파일을 편집하거나 미리 채워진 내용을 업로드할 수도 있습니다.

ARM 템플릿은 입력 검증 기능이 제한되어 있으므로 올바른 입력 검증을 제공하는 것은 사용자의 책임입니다.

단계 **11** 템플릿 구축이 성공하면 Azure용 위협 대응 가상 Auto Scale 솔루션에 필요한 모든 리소스가 생성됩니다. 다음 그림의 리소스를 참조하십시오. Type(유형) 열은 논리 앱, VMSS, 로드 밸런서, 공용 IP 주소 등 각 리소스에 대해 설명합니다.

그림 ***14: Threat Defense Virtual Auto Scale Template*** 구축



Azure Function 앱 구축

ARM 템플릿을 구축할 때 Azure는 기본 Function 앱을 생성합니다. 그러면 Auto Scale Manager 논리에 필요한 함수를 사용하여 수동으로 업데이트하고 구성해야 합니다.

시작하기 전에

- *ASM_Function.zip* 패키지를 빌드합니다. 소스 코드로 Azure 기능 빌드, 40 페이지의 내용을 참조하십시오.

프로시저

단계 **1** ARM 템플릿을 구축할 때 생성한 Function 앱으로 이동하여 함수가 없는지 확인합니다. 브라우저에서 다음 URL로 이동합니다.

https://<Function App Name>.scm.azurewebsites.net/DebugConsole

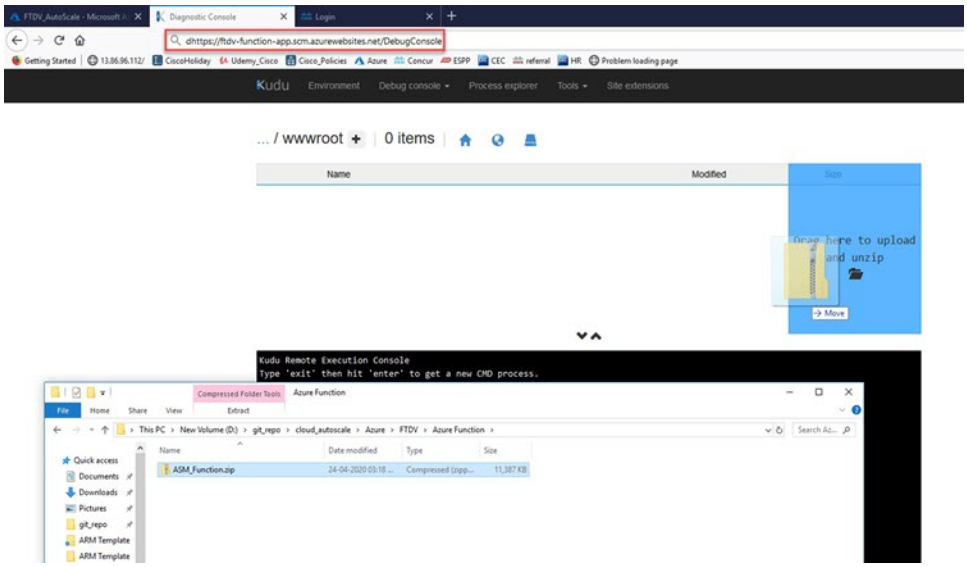
Auto Scale ARM 템플릿 구축, 21 페이지의 예:

https://ftdv-function-app.scm.azurewebsites.net/DebugConsole

단계 **2** 파일 탐색기에서 **site/wwwroot**로 이동합니다.

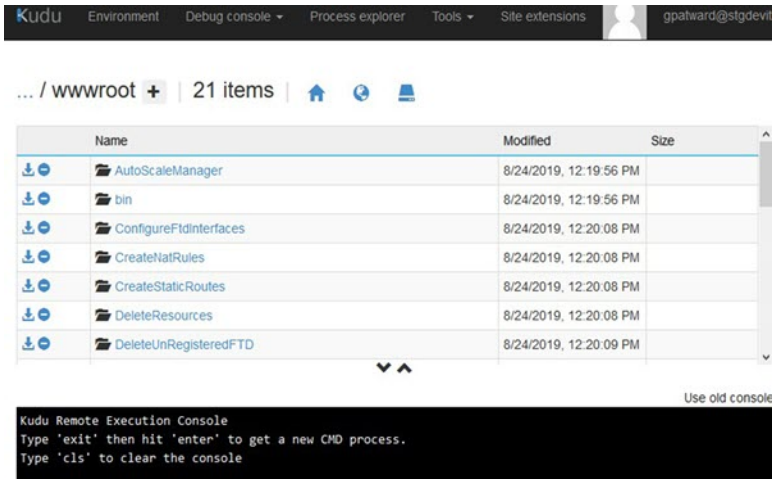
단계 **3** **ASM_Function.zip**을 파일 탐색기의 오른쪽 모서리로 끌어다 놓습니다.

그림 ***15: Threat Defense Virtual Auto Scale*** 기능 업로드



단계 **4** 업로드에 성공하면 모든 서버리스 함수가 표시됩니다.

그림 ***16: Threat Defense*** 가상 서버리스 기능

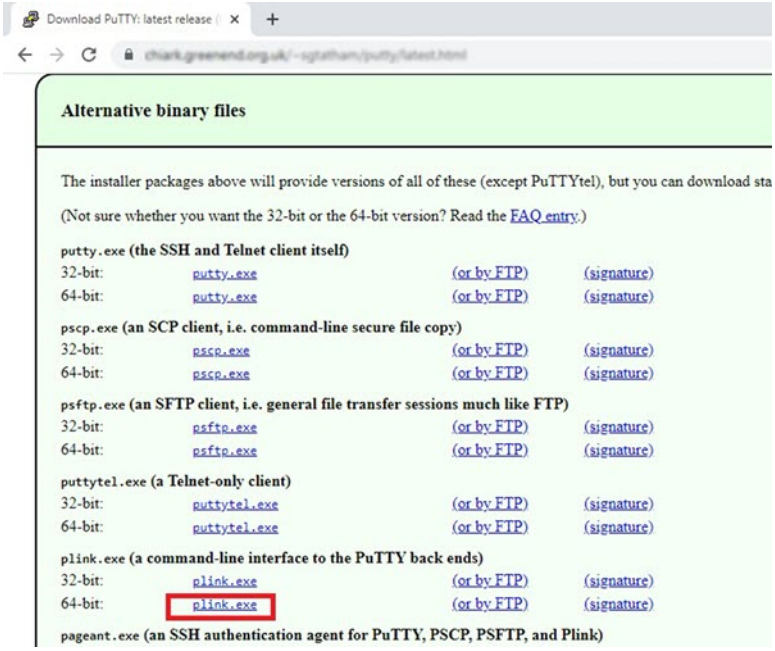


단계 **5** PuTTY SSH 클라이언트를 다운로드합니다.

Azure 함수는 SSH 연결을 통해 위협 대응 가상에 액세스해야 합니다. 그러나 서버리스 코드에서 사용되는 오픈 소스 라이브러리는 위협 대응 가상에서 사용하는 SSH 키 교환 알고리즘을 지원하지 않습니다. 따라서 사전 구축된 SSH 클라이언트를 다운로드해야 합니다.

www.putty.org에서 PuTTY 명령줄 인터페이스를 PuTTY 백엔드(*plink.exe*)에 다운로드합니다.

그림 ***17: PuTTY*** 다운로드



단계 **6** SSH 클라이언트 실행 파일의 이름 **plink.exe**를 **ftdssh.exe** 로 변경합니다.

단계 **7** 파일 탐색기의 오른쪽 모서리, 즉 이전 단계에서 **ASM_Function.zip**이 업로드된 위치에 **ftdssh.exe** 를 끌어다 놓습니다.

단계 **8** SSH 클라이언트에 해당 함수 애플리케이션이 있는지 확인합니다. 필요한 경우 페이지를 새로 고칩니다.

컨피그레이션 조정

Auto Scale Manager를 조정하거나 디버깅에 사용할 수 있는 몇 가지 컨피그레이션이 있습니다. 이러한 옵션은 ARM 템플릿에 표시되지 않지만 Function 앱 아래에서 수정할 수 있습니다.

시작하기 전에

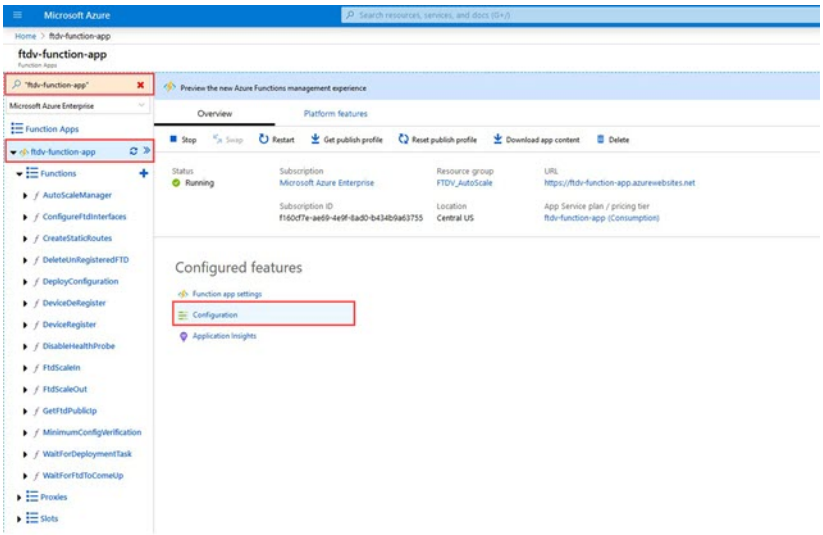
참고 이즌 언제든지 수정할 수 있습니다. 컨피그레이션을 수정하려면 이 순서를 따릅니다.

- Function 앱을 비활성화합니다.
- 기존의 예약된 작업이 완료 될 때까지 기다립니다.
- 컨피그레이션을 수정하고 저장합니다.
- Function 앱을 활성화합니다.

프로시저

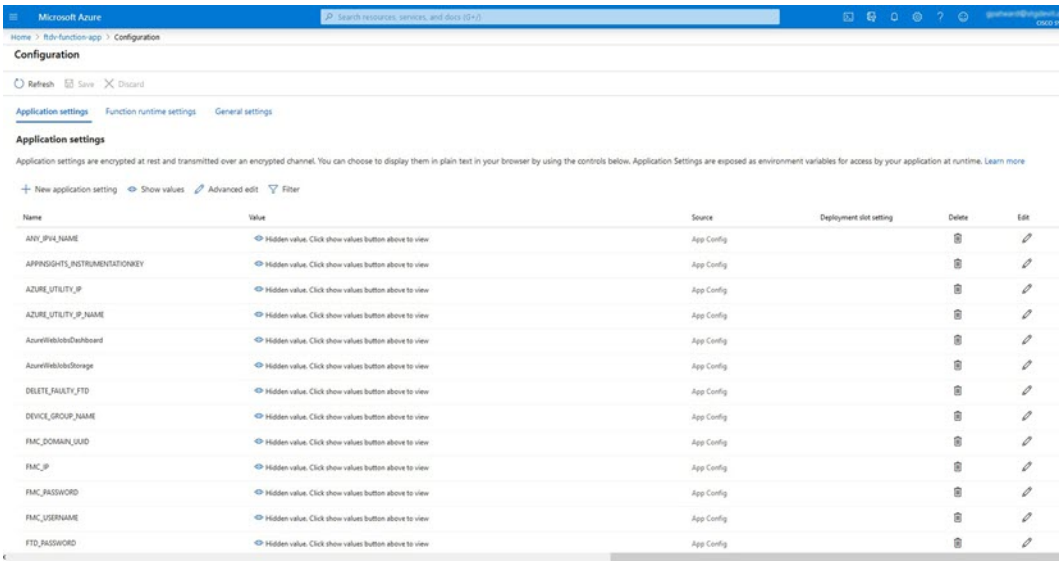
단계 **1** Azure Portal에서 위협 대응 가상 함수 애플리케이션을 검색하여 선택합니다.

그림 ***18: Threat Defense*** 가상 기능 애플리케이션



단계 **2** 여기서는 ARM 템플릿을 통해 전달된 컨피그레이션을 수정할 수도 있습니다. 변수 이름은 ARM 템플릿과 다르게 표시될 수 있지만 이러한 변수의 용도를 해당 이름에서 쉽게 식별할 수 있습니다.

그림 ***19:*** 애플리케이션 설정



대부분의 옵션은 이름에 그 설명을 담고 있습니다. 대표적인 예는 다음과 같습니다.

- 컨피그레이션 이름: “DELETE_FAULTY_FTD”(기본값: YES)

확장 중에 새 위협 대응 가상 인스턴스가 시작되고 management center에 등록됩니다. 등록 이 실패할 경우 이 옵션을 기반으로 Auto Scale Manager는 해당 위협 대응 가상 인스턴스를 유지하거나 삭제할지 결정합니다. (예: 결함 위협 대응 가상 삭제 / 아니요: management center에 등록하지 못하더라도 위협 대응 가상 인스턴스를 유지합니다.)

- Function 앱 설정에서는 Azure 구독에 대한 액세스 권한이 있는 사용자가 모든 변수('password'와 같은 보안 문자열을 포함하는 변수 포함)를 일반 텍스트 형식으로 볼 수 있습니다.

  사용자가 이에 대해 보안 문제가 있는 경우(예: 조직 내에서 권한이 낮은 사용자 간에 Azure 구독이 공유되는 경우) 사용자는 Azure의 *Key Vault* 서비스를 사용하여 비밀번호를 보호할 수 있습니다. 이 기능이 구성되면 기능 설정에서 일반 텍스트 '비밀번호'를 제공하는 대신 비밀번호가 저장된 키 저장소에서 생성된 보안 식별자를 제공해야 합니다.

  참고 Azure 문서를 검색하여 애플리케이션 데이터를 보호하는 모범 사례를 찾습니다.

가상 시스템 확장 집합의 **IAM** 역할 구성

Azure IAM (Identity and Access Management)은 사용자 ID를 관리하고 제어하기 위해 Azure Security and Access Control의 일부로 사용됩니다. Azure 리소스의 관리되는 ID는 Azure Active Directory의 자동으로 관리되는 ID를 Azure 서비스에 제공합니다.

이를 통해 Function 앱은 명시적 인증 자격 증명 없이 VMSS(Virtual Machine Scale Sets)를 제어할 수 있습니다.

프로시저

단계 **1** Azure 포털에서 VMSS로 이동합니다.

단계 **2** 액세스 제어**(IAM)**를 클릭합니다.

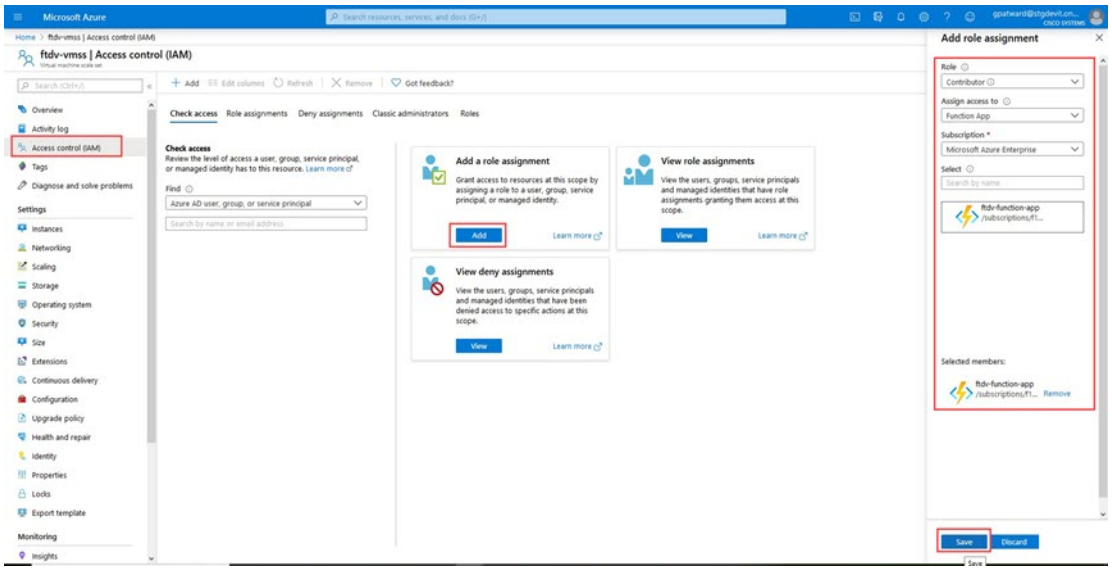
단계 **3** **Add(**추가**)**를 클릭하여 역할 할당을 추가합니다.

단계 **4** **Add role Assignment(**역할 할당 추가**)** 드롭 다운에서 **Contributor(**기여자**)**를 선택합니다.

단계 **5** **Assign access to(**액세스 할당 대상**)** 드롭 다운에서 **Function App(**기능 앱**)**을 선택합니다.

단계 **6** 위협 대응 가상 함수 애플리케이션을 선택합니다.

그림 ***20: AIM*** 역할 할당



단계 **7** **Save(**저장**)**를 클릭합니다.

참고 또한 아직 시작된 위협 대응 가상 인스턴스가 없는지 확인해야 합니다.

보안 그룹 업데이트

ARM 템플릿은 Management 인터페이스용과 데이터 인터페이스용의 두 가지 보안 그룹을 생성합니다. Management 보안 그룹은 위협 대응 가상 관리 활동에 필요한 트래픽만 허용합니다. 그러나 데이터 인터페이스 보안 그룹은 모든 트래픽을 허용합니다.

프로시저

구축의 토폴로지 및 애플리케이션 요구 사항에 따라 보안 그룹 규칙을 세부적으로 조정합니다.

참고 데이터 인터페이스 보안 그룹은 로드 밸런서의 최소 SSH 트래픽을 허용해야 합니다.

Azure Logic 앱 업데이트

Logic 앱은 Autoscale 기능의 오케스트레이터 역할을 합니다. ARM 템플릿은 기본 Logic 앱을 생성합니다. 그러면 Auto Scale 오케스트레이터로 작동하는 데 필요한 정보를 제공할 수 있도록 수동으로 업데이트해야 합니다.

프로시저

단계 **1** 리포지토리에서 *LogicApp.txt* 파일을 로컬 시스템으로 검색하고 아래 표시된 대로 수정합니다.

중요 계속하기 전에 이 단계를 모두 읽고 숙지하십시오.

이러한 수동 단계는 ARM 템플릿에서 자동화되지 않으므로 나중에 Logic 앱만 독립적으로 업그레이드 할 수 있습니다.

a) 필수: "SUBSCRIPTION_ID"의 모든 어커런스를 찾아서 구독 ID 정보로 교체합니다.
b) 필수: "RG_NAME" 어커런스를 모두 찾아서 리소스 그룹 이름으로 바꿉니다.
c) 필수: "FUNCTIONAPPNAME" 어커런스를 모두 찾아서 함수 앱 이름으로 바꿉니다.

다음 예에서는 *LogicApp.txt* 파일에서 이러한 행 중 일부를 보여줍니다.

```
  "AutoScaleManager": {
      "inputs": {
          "function": {
              "id":
"/subscriptions/SUBSCRIPTION_ID/resourceGroups/RG_NAME/providers/Microsoft.Web/sites/FUNCTIONAPPNAME/functions/AutoScaleManager"

          }
.
.
                      },
                      "Deploy_Changes_to_FTD": {
                          "inputs": {
                              "body": "@body('AutoScaleManager')",
                              "function": {
                                  "id":
"/subscriptions/SUBSCRIPTION_ID/resourceGroups/RG_NAME/providers/Microsoft.Web/sites/FUNCTIONAPPNAME/functions/DeployConfiguration"

                              }
.
.
                      "DeviceDeRegister": {
                          "inputs": {
                              "body": "@body('AutoScaleManager')",
                              "function": {
                                  "id":
"/subscriptions/SUBSCRIPTION_ID/resourceGroups/RG_NAME/providers/Microsoft.Web/sites/FUNCTIONAPPNAME/functions/DeviceDeRegister"

                              }
                          },
                          "runAfter": {
                              "Delay_For_connection_Draining": [
```

d) (선택 사항) 트리거 간격을 수정하거나 기본값(5)을 유지합니다. 이는 Autoscale 기능이 주기적으로 트리거되는 시간 간격입니다. 다음 예는 *LogicApp.txt* 파일에서 이러한 행을 보여줍니다.

```
        "triggers": {
            "Recurrence": {
                "conditions": [],
                "inputs": {},
                "recurrence": {
                    "frequency": "Minute",
                    "interval": 5
```

```
},
```

e) (선택 사항) 드레인 시간을 수정하거나 기본값(5)을 유지합니다. 이는 축소(Scale-In) 작업 중에 디바이스를 삭제하기 전에 위협 대응 가상에서 기존 연결을 드레 이닝하는 시간 간격입니다. 다음 예는 *LogicApp.txt* 파일에서 이러한 행을 보여줍니다.

```
"actions": {
     "Branch_based_on_Scale-In_or_Scale-Out_condition": {
         "actions": {
              "Delay_For_connection_Draining": {
                  "inputs": {
                      "interval": {
                          "count": 5,
                          "unit": "Minute"
                      }
```

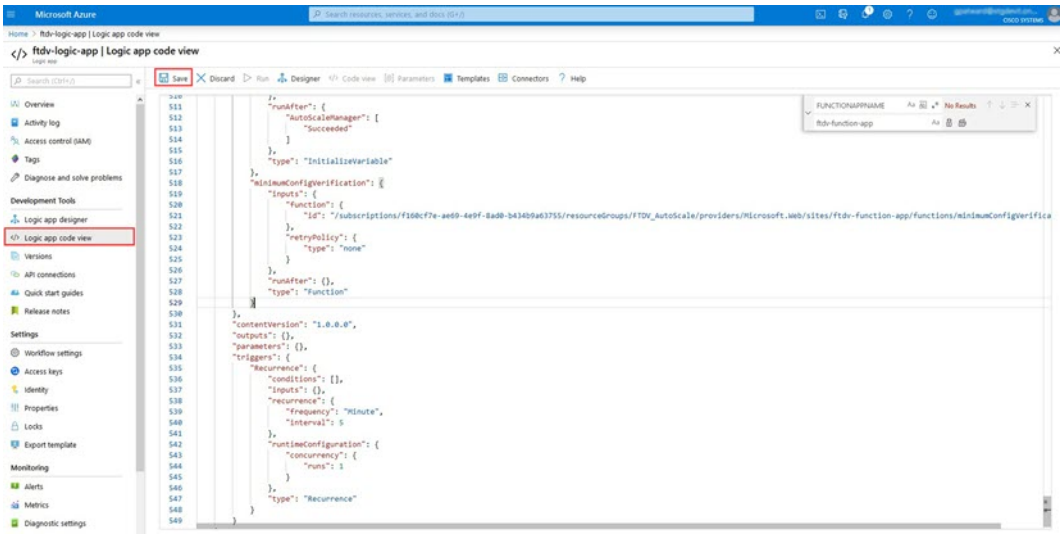
f) (선택 사항) 냉각 시간을 수정하거나 기본값(10)을 유지합니다. 이 시간은 확장(Scale-Out)이 완료된 후 작업 없음을 유지하는 시간입니다. 다음 예는 *LogicApp.txt* 파일에서 이러한 행을 보여줍니다.

```
"actions": {
    "Branch_based_on_Scale-Out_or_Invalid_condition": {
        "actions": {
            "Cooldown_time": {
                "inputs": {
                    "interval": {
                        "count": 10,
                        "unit": "Second"
                    }
```

참고 이러한 단계는 Azure 포털에서도 수행할 수 있습니다. 자세한 내용은 Azure 문서를 참조하십시오.

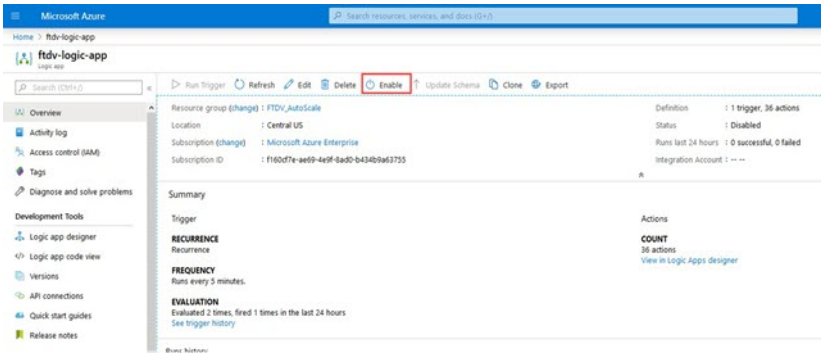
단계 **2** **Logic** 앱 코드 보기로 이동하여 기본 콘텐츠를 삭제하고 수정된 *LogicApp.txt* 파일에서 콘텐츠를 붙여넣고 **Save(**저장**)**을 클릭합니다.

그림 ***21: Logic*** 앱 코드 보기



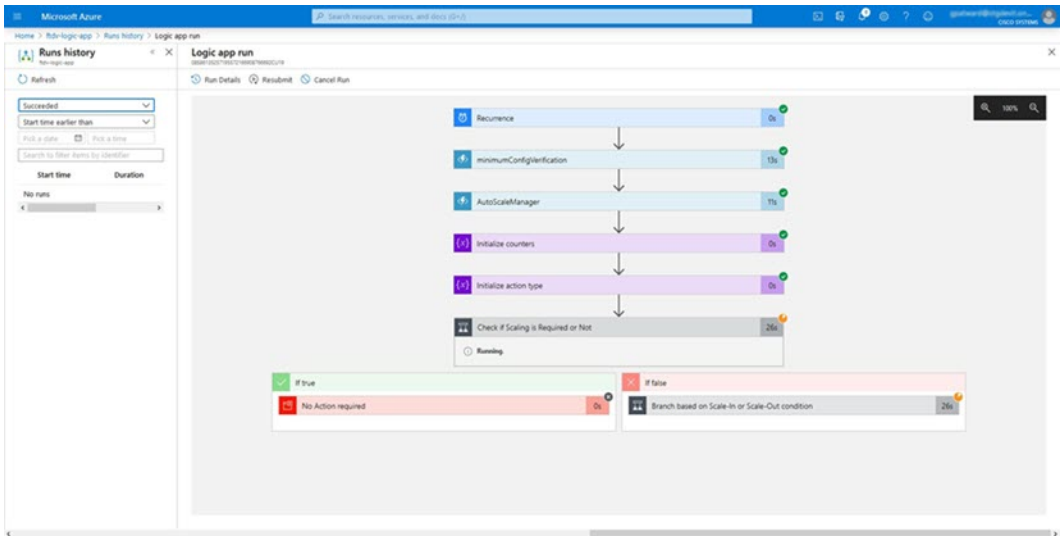
단계 **3** Logic 앱을 저장하면 '비활성화' 상태가 됩니다. Auto Scale Manager를 시작하려면 **Enable(**활성화**)**을 클릭합니다.

그림 ***22: Logic*** 앱 활성화



단계 **4** 활성화되면 작업이 실행되기 시작합니다. 활동을 보려면 '실행 중' 상태를 클릭하십시오.

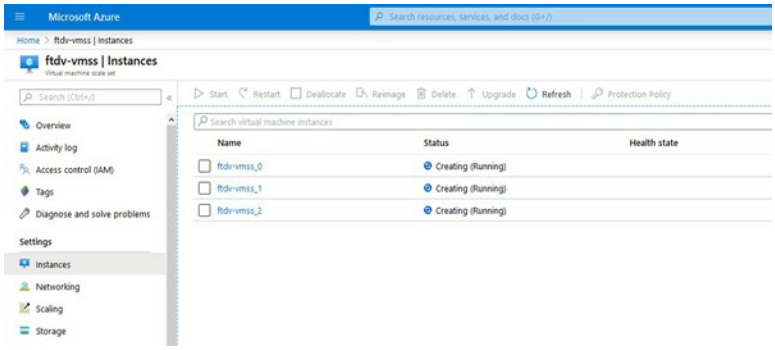
그림 *23: Logic* 앱 실행 상태



단계 **5** Logic 앱이 시작되면 모든 구축 관련 단계가 완료됩니다.

단계 **6** VMSS에서 위협 대응 가상 인스턴스가 생성되고 있는지 확인합니다.

그림 *24:* 실행 중인 *Threat Defense Virtual* 인스턴스



이 예에서는 ARM 템플릿 구축에서 'minFtdCount' 가 '3'으로, 'initDeploymentMode'가 'BULK'로 설정되었으므로 3개의 위협 대응 가상 인스턴스가 시작됩니다.

Threat Defense Virtual 업그레이드

위협 대응 가상 업그레이드는 VMSS(Virtual Machine Scale Set)의 이미지 업그레이드 형식으로만 지원됩니다. 따라서 Azure REST API 인터페이스를 통해 위협 대응 가상를 업그레이드합니다.

참고 모든 REST 클라이언트를 사용하여 위협 대응 가상를 업그레이드할 수 있습니다.

시작하기 전에

- 마켓플레이스에서 사용 가능한 새 위협 대응 가상 이미지 버전을 가져옵니다(예: 650.32.0).
- 원래 스케일 세트를 구축하는 데 사용된 SKU를 가져옵니다(예: ftdv -azure-byol).
- 리소스 그룹 및 가상 시스템 확장 집합 이름을 가져옵니다.

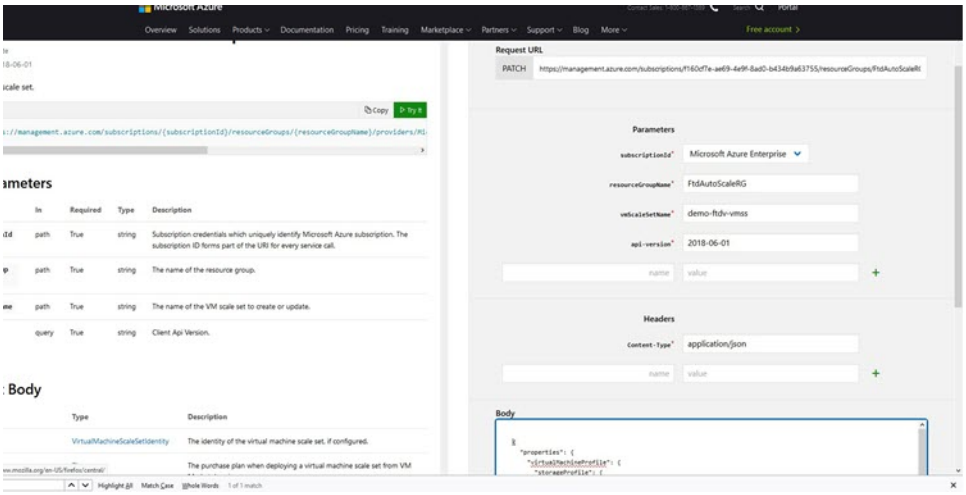
프로시저

단계 **1** 브라우저에서 다음 URL로 이동합니다.

https://docs.microsoft.com/en-us/rest/api/compute/virtualmachinescalesets/update#code-try-0

단계 **2** 매개변수 섹션에 세부 사항을 입력합니다.

그림 ***25: Threat Defense Virtual*** 업그레이드



단계 **3** 본문 섹션에 새로운 위협 대응 가상 이미지 버전, SKU 및 트리거 RUN을 포함하는 JSON 입력을 입력합니다.

```
{
 "properties": {
        "virtualMachineProfile": {
              "storageProfile": {
                     "imageReference": {
                          "publisher": "cisco",
                          "offer": "cisco-ftdv",
                          "sku": "ftdv-azure-byol",
                           "version": "650.32.0"
                     }
                },
            }
      }
}
```

단계 **4** Azure의 성공적인 응답은 VMSS가 변경 사항을 수락했음을 의미합니다.

새 위협 대응 가상 이미지는 새 인스턴스에서 사용되며, 이는 확장 작업의 일부로 시작됩니다.

- 기존 위협 대응 가상 인스턴스는 확장 집합에 있는 동안 기존 소프트웨어 이미지를 계속 사용합니다.
- 위의 동작을 재정의하고 기존 위협 대응 가상 인스턴스를 수동으로 업그레이드할 수 있습니다. 이렇게 하려면 VMSS에서 **Upgrade(**업그레이드**)** 버튼을 클릭합니다. 선택한 위협 대응 가상 인스턴스가 재부팅되고 업그레이드됩니다. 이러한 업그레이드된 위협 대응 가상 인스턴스를 수동으로 다시 등록하고 재구성해야 합니다. 이 방법은 권장되지 않습니다**.**

Auto Scale 논리

확장 메트릭

ARM 템플릿을 사용하여 threat defense virtual Auto Scale 솔루션에 필요한 리소스를 구축합니다. ARM 템플릿 구축 중에는 다음과 같은 확장 메트릭 옵션이 제공됩니다.

- CPU
- CPU, 메모리(버전 6.7 이상)

참고 CPU 메트릭은 Azure에서 수집되며 메모리 메트릭은 management center에서 수집됩니다.

확장 논리

- **POLICY-1**: 어떤 경우든 위협 대응 가상 평균로드가 구성된 기간 동안 확장 임계값을 초과하면 확장(Scale-Out)이 트리거됩니다. 'CPU, 메모리' 확장 메트릭을 사용하는 경우 확장 임계값은 확장 집합에 있는 모든 threat defense virtual의 평균 CPU 또는 메모리 사용률입니다.
- **POLICY-2**: 구성된 기간 동안 모든 위협 대응 가상 디바이스의 평균로드가 확장 임계값을 초과하면 확장(Scale-Out)이 트리거됩니다. 'CPU, 메모리' 확장 메트릭을 사용하는 경우 확장 임계값은 확장 집합에 있는 모든 threat defense virtual디바이스의 평균 CPU 또는 메모리 사용률입니다.

축소 논리

- 모든 위협 대응 가상 디바이스의 CPU 사용률이 구성된 기간 동안 구성된 축소 임계값 미만인 경우. 'CPU, 메모리' 확장 메트릭을 사용할 때 확장 집합의 모든 threat defense virtual 디바이스의 CPU 및 메모리 사용률이 구성된 기간 동안 구성된 축소 임계값 아래로 내려가면 CPU로드가 가장 적은 threat defense virtual가 종료되도록 선택됩니다.

참고

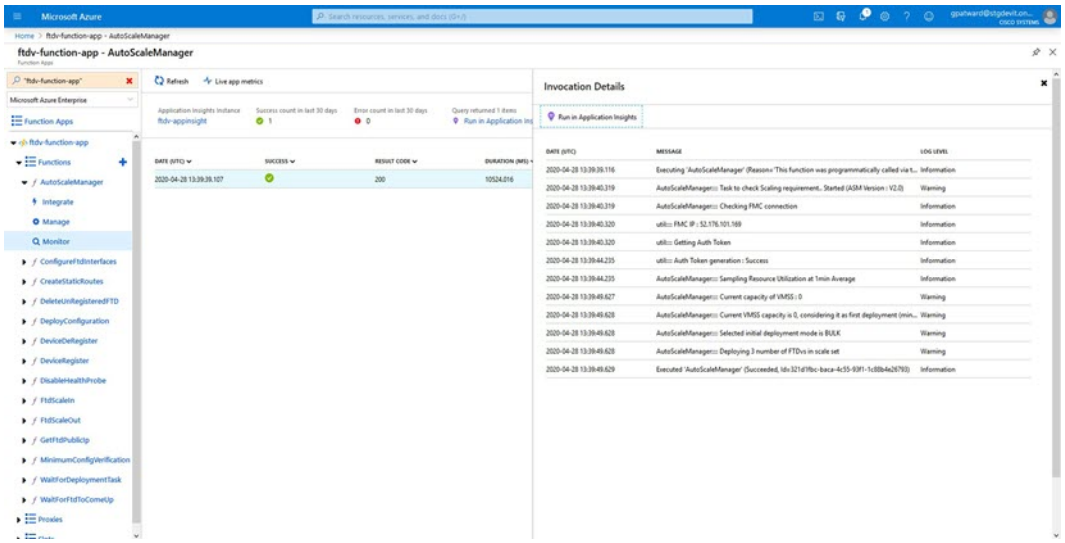
- 축소(Scale-In)/확장(Scale-Out)은 1단계로 수행됩니다(즉, 한 번에 1개 위협 대응 가상만 축소/확장).
- management center에서 수신한 메모리 사용량 메트릭은 시간 경과에 따라 계산된 평균 값이 아니라 순간 스냅 샷/샘플 값입니다. 따라서 메모리 메트릭만으로는 확장 결정을 내릴 수 없습니다. 구축 중에는 메모리 전용 메트릭을 사용할 수 있는 옵션이 없습니다.

Auto Scale 로깅 및 디버깅

서버리스 코드의 각 구성 요소에는 자체 로깅 메커니즘이 있습니다. 또한 로그는 애플리케이션 인사이트에 게시됩니다.

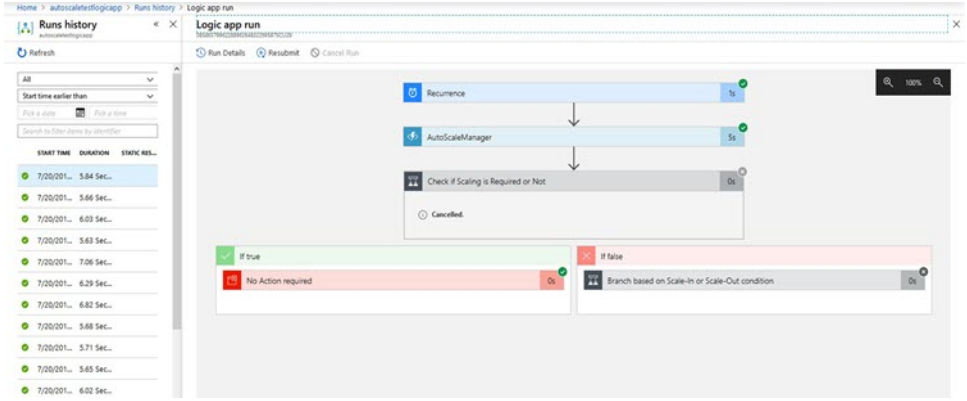
- 개별 Azure 함수의 로그를 볼 수 있습니다.

  그림 ***26: Azure Function*** 로그



- 각 Logic App 실행 및 해당 개별 구성 요소에 대한 유사한 로그를 볼 수 있습니다.

그림 ***27: Logic*** 앱 실행 로그



- 필요한 경우 Logic App에서 실행중인 작업을 언제든지 중지/종료할 수 있습니다. 그러나 현재 실행 중이거나 종료되는 위협 대응 가상 디바이스는 일관성이 없는 상태로 유지됩니다.
- 각 실행/개별 작업에 소요되는 시간은 Logic 앱에서 확인할 수 있습니다.
- 언제든지 새 zip을 업로드하여 Function 앱을 업그레이드할 수 있습니다. Function 앱을 업그레이드하기 전에 Logic 앱을 중지하고 모든 작업이 완료될 때까지 기다립니다.

Auto Scale 지침 및 제한 사항

위협 대응 가상 Auto Scale for Azure를 구축할 때 다음 지침 및 제한 사항에 유의하십시오.

- (버전 6.6 이하) 확장 결정은 CPU 사용률을 기반으로 합니다.
- (버전 6.7 이상) 확장 결정에서는 CPU 전용 사용률 또는 CPU 및 메모리 사용률을 사용할 수 있습니다.
- Management Center 관리가 필요합니다. Device Manager는 지원되지 않습니다.
- management center에는 공용 IP 주소가 있어야 합니다.
- 위협 대응 가상 Management 인터페이스가 공용 IP 주소를 갖도록 구성되었습니다.
- IPv4만 지원됩니다.
- Threat Defense Virtual Auto Scale for Azure는 액세스 정책, NAT 정책, 플랫폼 설정 등 디바이스 그룹에 적용되며 확장된 threat defense virtual 인스턴스에 전파되는 설정만을 지원합니다. management center을 사용한 디바이스 그룹 설정만을 수정할 수 있습니다. 디바이스별 컨피그레이션은 지원되지 않습니다.
- ARM 템플릿은 입력 검증 기능이 제한되어 있으므로 올바른 입력 검증을 제공하는 것은 사용자의 책임입니다.
- Azure 관리자는 Function 앱 환경 내에서 민감한 데이터(예: 관리자 로그인 자격 증명 및 비밀번호)를 일반 텍스트 형식으로 볼 수 있습니다. *Azure Key Vault* 서비스를 사용하여 민감한 데이터를 보호할 수 있습니다.

Auto Scale 문제 해결

다음은 일반적인 오류 시나리오 및 위협 대응 가상 Auto Scale for Azure에 대한 디버깅 팁입니다.

- management center에 연결 실패: management center IP / 자격 증명을 확인하십시오. management center에 결함이 있거나 연결할 수 없는지 확인합니다.
- 위협 대응 가상로 SSH할 수 없음 : 템플릿을 통해 복잡한 비밀번호가 위협 대응 가상에 전달되는지 확인합니다. 보안 그룹에서 SSH 연결을 허용하는지 확인하십시오.
- 로드 밸런서 상태 확인 실패: 위협 대응 가상에서 데이터 인터페이스의 SSH에 응답하는지 확인합니다. 보안 그룹 설정을 확인합니다.
- 트래픽 문제: 로드 밸런서 규칙, NAT 규칙 / 위협 대응 가상에 구성된 고정 경로를 확인합니다. 템플릿 및 보안 그룹 규칙에 제공된 Azure 가상 네트워크 / 서브넷 / 게이트웨이 세부 정보를 확인합니다.
- threat defense virtual가 management center에 등록 실패: 새 threat defense virtual 디바이스를 수용할 수 있도록 management center 용량을 확인하십시오. 라이선싱을 확인합니다. threat defense virtual 버전 호환성을 확인하십시오.
- Logic 앱이 VMSS에 액세스하지 못함: VMSS의 IAM 역할 컨피그레이션이 올바른지 확인하십시오.
- Logic 앱이 매우 오랫동안 실행 됨: 확장된 위협 대응 가상 디바이스에서 SSH 액세스를 확인합니다. management center에서 디바이스 등록 문제를 확인합니다. Azure VMSS에서 위협 대응 가상 디바이스의 상태를 확인합니다.
- 구독 ID와 관련된 오류 발생 Azure Function: 계정에서 기본 구독이 선택되었는지 확인하십시오.
- 축소(Scale-In) 작업 실패: 경우에 따라 Azure에서 인스턴스를 삭제하는 데 시간이 오래 걸리는 경우가 있습니다. 이러한 상황에서는 축소 작업이 시간 초과되고 오류를 보고할 수 있지만 결국엔 인스턴스가 삭제됩니다.
- 컨피그레이션 변경을 수행하기 전에 논리 애플리케이션을 비활성화하고 실행중인 모든 작업이 완료될 때까지 기다리십시오.

소스 코드로 Azure 기능 빌드

시스템 요구 사항

- Microsoft Windows 데스크톱 / 노트북
- Visual Studio(Visual Studio 2019 버전 16.1.3에서 테스트)

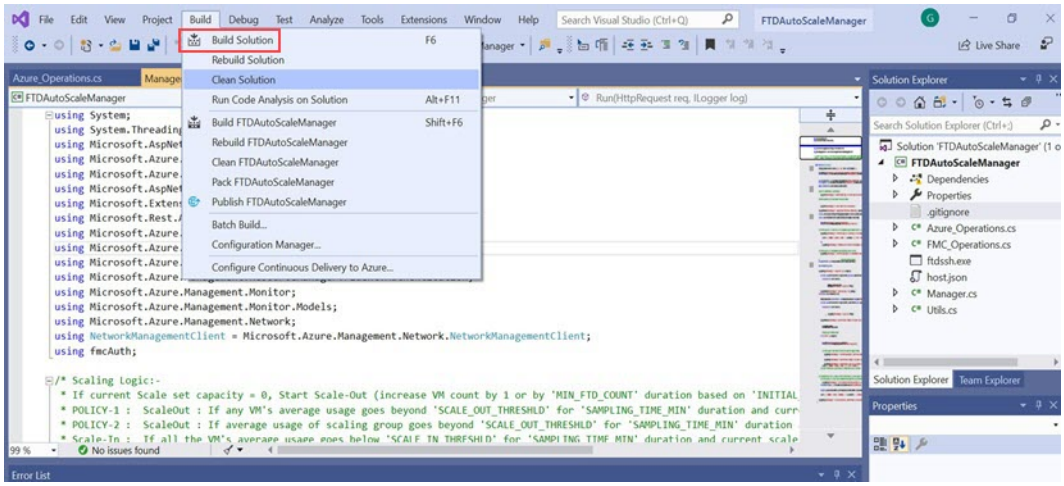
참고 Azure 함수는 C#을 사용하여 작성됩니다.

- "Azure 개발" 워크로드를 Visual Studio에 설치해야 합니다.

Visual Studio로 빌드

1. 'code' 폴더를 로컬 시스템에 다운로드합니다.
2. 'FTDAutoScaleManager '폴더로 이동합니다.
3. Visual Studio에서 'FTDAutoScaleManager ' 프로젝트 파일을 엽니다.
4. Visual Studio 표준 절차를 사용하여 정리 및 빌드합니다.

그림 ***28: Visual Studio*** 빌드



5. 빌드가 성공적으로 컴파일되면 **\bin\Release\netcoreapp2.1** 폴더로 이동합니다.
6. 모든 내용을 선택하고 **Send to** > **Compressed(zipped)(**압축 폴더로 전송**)**을 클릭하고 ZIP 파일을 *ASM_Function.zip*으로 저장합니다.

그림 ***29: Build ASM_Function.zip***