



# Microsoft Azure에서 ASA 가상 Auto Scale 솔루션 구축

- Azure의 ASA 가상용 Auto Scale 솔루션, 1 페이지
- 구축 패키지 다운로드, 3 페이지
- Auto Scale 솔루션 구성 요소, 3 페이지
- Auto Scale 솔루션 사전 요건, 5 페이지
- Auto Scale 구축, 11 페이지
- Auto Scale 논리, 26 페이지
- Auto Scale 로깅 및 디버깅, 26 페이지
- Auto Scale 지침 및 제한 사항, 27 페이지
- Auto Scale 문제 해결, 28 페이지
- 소스 코드로 Azure 기능 빌드, 28 페이지

## Azure의 ASA 가상용 Auto Scale 솔루션

### Auto Scale 솔루션

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

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

- ARM(Azure Resource Manager) 템플릿 기반 구축
- CPU 기반의 메트릭 확장을 지원합니다.



참고 자세한 내용은 [Auto Scale 논리, 26 페이지](#)를 참조하십시오.

- ASA 가상 구축 및 다중 가용성 영역 지원

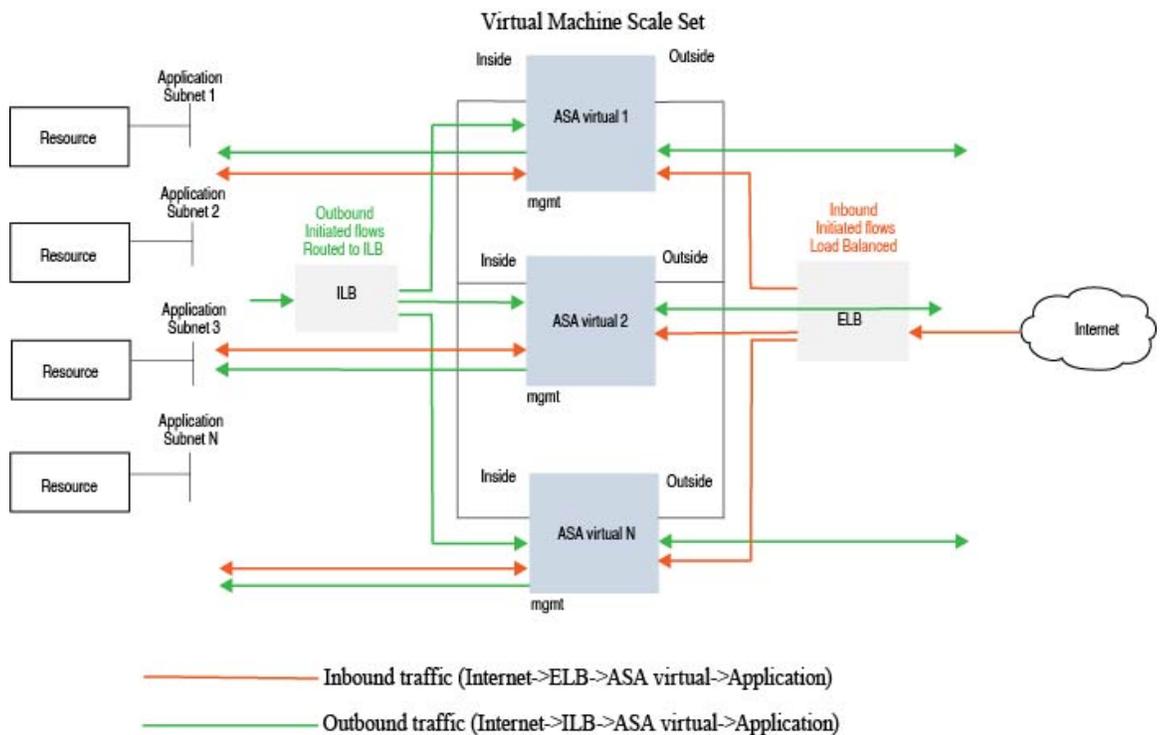
- 확장된 ASA 가상 인스턴스에 자동으로 적용되는 완전히 자동화된 컨피그레이션.
- 로드 밸런서 및 다중 가용성 영역 지원
- Auto Scale 기능 활성화 및 비활성화 지원
- Cisco에서는 구축을 쉽게 수행할 수 있도록 Azure용 Auto Scale 구축 패키지를 제공합니다.

## 사용 사례

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

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

그림 1: 사용 사례



## 범위

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



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

## 구축 패키지 다운로드

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

Azure용 ASA 가상 Auto Scale 솔루션을 시작하는 데 필요한 파일을 다운로드합니다. 사용자 버전에 맞는 구축 스크립트 및 템플릿은 [GitHub](#) 리포지토리에서 제공됩니다.



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

*ASM\_Function.zip* 패키지를 구축하는 방법에 대한 지침은 [소스 코드로 Azure 기능 빌드, 28 페이지](#)를 참고하십시오.

---

## Auto Scale 솔루션 구성 요소

다음 구성 요소는 Azure용 ASA 가상 Auto Scale 솔루션을 구성합니다.

### Azure Functions(Function 앱)

Function 앱은 Azure 함수의 집합입니다. 기본 기능은 다음과 같습니다.

- Azure 메트릭을 주기적으로 통신/프로브합니다.
- ASA 가상 로드를 모니터링하고 축소/확장(Scale In/Scale Out) 작업을 트리거합니다.

이들 함수는 압축된 Zip 패키지 형식으로 제공됩니다(Azure Function 앱 패키지 빌드, 7 페이지 참조). 함수는 특정 작업을 수행하기 위해 가능한 한 개별적으로 유지되며, 개선 사항 및 새로운 릴리스 지원을 위해 필요에 따라 업그레이드할 수 있습니다.

**오케스트레이터(Logic 앱)**

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

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

**VMSS(Virtual Machine Scale Set)**

VMSS는 ASA 가상 디바이스와 같은 균일한 가상 시스템의 모음입니다.

- VMSS는 해당 집합에 동일한 새 VM을 추가할 수 있습니다.
- VMSS에 추가된 새 VM은 로드 밸런서, 보안 그룹 및 네트워크 인터페이스에 자동으로 연결됩니다.
- VMSS에는 Azure ASA 가상용으로 사용하지 않도록 설정된 Auto Scale 기능이 내장되어 있습니다.
- VMSS에서 ASA 가상 인스턴스를 수동으로 추가하거나 삭제해서는 안 됩니다.

**ARM(Azure Resource Manager) 템플릿**

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

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

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




---

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

---

# Auto Scale 솔루션 사전 요건

## Azure 리소스

### 리소스 그룹

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



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

### 네트워킹

가상 네트워크가 사용 가능하거나 생성되었는지 확인합니다. 를 이용한 Auto Scale 구축은 네트워킹 리소스를 생성, 변경 또는 관리하지 않습니다.

ASA 가상에는 3개 네트워크 인터페이스가 필요하므로 가상 네트워크에는 3개 서브넷이 필요합니다.

1. 관리 트래픽
2. 내부 트래픽
3. 외부 트래픽

서브넷이 연결된 네트워크 보안 그룹에서 다음 포트를 열어야 합니다.

- SSH(TCP/22)

로드 밸런서와 ASA 가상 사이의 상태 프로브에 필요합니다.

서버리스 함수와 ASA 가상 간의 통신에 필요합니다.

- 애플리케이션별 프로토콜/포트

모든 사용자 애플리케이션(예: TCP/80)에 필요합니다.



**참고** 가상 네트워크 이름, 가상 네트워크 CIDR, 3개 서브넷의 이름, 외부 및 내부 서브넷의 게이트웨이 IP 주소를 기록합니다.

## ASA 컨피그레이션 파일 준비

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

ASA 컨피그레이션 파일에는 최소한 다음이 포함되어야 합니다.

- 모든 인터페이스에 DHCP IP 할당을 설정합니다.
- GigabitEthernet0/1은 '내부' 인터페이스여야 합니다.
- GigabitEthernet0/0은 '외부' 인터페이스여야 합니다.
- 게이트웨이를 내부 및 외부 인터페이스로 설정합니다.
- (상태 프로브를 위해) Azure 유틸리티 IP의 내부 및 외부 인터페이스에서 SSH를 활성화합니다.
- 외부에서 내부 인터페이스로 트래픽을 전달하도록 NAT 구성을 생성합니다.
- 원하는 트래픽을 허용하는 액세스 정책을 생성합니다.
- 컨피그레이션에 라이선스를 부여합니다. PAYG 청구는 지원되지 않습니다.



참고 관리 인터페이스를 별도로 구성하지 않아도 됩니다.

다음은 .

```
ASA Version 9.13(1)
!
interface GigabitEthernet0/1
nameif inside
security-level 100
ip address dhcp setroute
!
interface GigabitEthernet0/0
nameif outside
security-level 0
ip address dhcp setroute
!
route outside 0.0.0.0 0.0.0.0 10.12.3.1 2
!
route inside 0.0.0.0 0.0.0.0 10.12.2.1 3
!
ssh 168.63.129.0 255.255.255.0 outside
!
ssh 168.63.129.0 255.255.255.0 inside
!
object network webserver
host 10.12.2.5
object service myport
service tcp source range 1 65535 destination range 1 65535
access-list outowebaccess extended permit object myport any any log disable
access-group outowebaccess in interface outside
object service app
```

```

service tcp source eq www
nat (inside,outside) source static webserver interface destination static interface any
service app app
object network obj-any
subnet 0.0.0.0 0.0.0.0
nat (inside,outside) source dynamic obj-any interface destination static obj-any obj-any
configure terminal
dns domain-lookup management
policy-map global_policy
class inspection_default
inspect icmp
call-home
profile License
destination transport-method http
destination address http https://tools.cisco.com/its/service/oddce/services/DDCEService
license smart
feature tier standard
throughput level 2G
license smart register idtoken <TOKEN>
: end

```

## Azure Function 앱 패키지 빌드

ASA 가상 Auto Scale 솔루션은 아카이브 파일인 *ASM\_Function.zip* 빌드가 필요하며, 이 파일은 압축된 ZIP 패키지로 개별 Azure 기능을 제공합니다.

*ASM\_Function.zip* 패키지를 구축하는 방법에 대한 지침은 [소스 코드로 Azure 기능 빌드, 28 페이지](#)를 참고하십시오.

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

## 입력 매개변수

다음 표에서는 템플릿 매개 변수를 정의하고 일 예를 제공합니다. 이러한 값을 결정하고 나면 Azure 구독에 ARM 템플릿을 구축할 때 이러한 매개 변수를 사용하여 ASA 가상 디바이스를 생성할 수 있습니다. [Auto Scale ARM 템플릿 구축, 11 페이지](#)의 내용을 참조하십시오.

표 1: 템플릿 매개변수

매개 변수 이름	허용되는 값 / 유형	설명	리소스 생성 유형
resourceNamePrefix	문자열*(3~10자)	모든 리소스는 이 접두사를 포함하는 이름으로 생성됩니다. 참고: 소문자만 사용하십시오. 예: asav	New
virtualNetworkRg	문자열	가상 네트워크 리소스 그룹 이름입니다. 예: cisco-virtualnet-rg	기존

매개 변수 이름	허용되는 값 / 유형	설명	리소스 생성 유형
virtualNetworkName	문자열	가상 네트워크 이름(이미 생성됨) 예: cisco-virtualnet	기존
mgmtSubnet	문자열	관리 서브넷 이름(이미 생성됨) 예: cisco-mgmt-subnet	기존
insideSubnet	문자열	내부 서브넷 이름(이미 생성됨) 예: cisco-inside-subnet	기존
internalLbIp	문자열	내부 서브넷(이미 생성됨)의 내부 로드 밸런서 IP 주소입니다. 예: 1.2.3.4.	기존
outsideSubnet	문자열	외부 서브넷 이름(이미 생성됨) 예: cisco-outside-subnet	기존
softwareVersion	문자열	ASA 가상 버전(구축 중 드롭 다운에서 선택) 기본값: 914.1.0 허용됨: 914.1.0, 913.1.0	기존
vmSize	문자열	ASA 가상 인스턴스의 크기(구축 중 드롭 다운에서 선택).	해당 없음
asaAdminUserName	문자열*	ASA 가상 '관리자' 사용자의 사용자 이름입니다.  비밀번호는 12 ~ 72자여야 하며 소문자, 대문자, 숫자 및 특수 문자를 포함해야 합니다. 같은 문자를 세 번 이상 반복해서 사용할 수 없습니다.  이는 관리자가 될 수 없습니다. Azure for VM administrator user name guidelines를 참조하십시오.  참고 템플릿에는 이에 대한 규정 준수 확인이 없습니다.	New

매개 변수 이름	허용되는 값 / 유형	설명	리소스 생성 유형
asaAdminUserPassword	문자열*	<p>ASA 가상 관리자 사용자의 비밀번호입니다.</p> <p>비밀번호는 12 ~ 72자여야 하며 소문자, 대문자, 숫자 및 특수 문자를 포함해야 합니다. 같은 문자를 세 번 이상 반복해서 사용할 수 없습니다.</p> <p>참고    템플릿에는 이에 대한 규정 준수 확인이 없습니다.</p>	New
scalingPolicy	POLICY-1 / POLICY-2	<p><b>POLICY-1:</b> 어떤 ASA 가상 평균 로드 구성 기간 동안 확장 임계값을 초과하면 확장(Scale-Out)이 트리거됩니다.</p> <p><b>POLICY-2:</b> 자동 확장 그룹 내의 모든 ASA 가상 평균로드 구성 기간 동안 확장 임계값을 초과하면 확장(Scale-Out)이 트리거됩니다.</p> <p>두 경우 모두 축소(Scale-In) 논리는 동일하게 유지됩니다. 모든 ASA 가상 디바이스의 평균로드 구성 기간 동안 축소 임계값 미만이 되면 축소가 트리거됩니다.</p>	해당 없음
scalingMetricsList	문자열	<p>스케일링 결정을 내리는 데 사용되는 메트릭입니다.</p> <p>허용됨: CPU 기본값: CPU</p>	해당 없음
scaleInThreshold	문자열	<p>에 대한 축소 임계값입니다.</p> <p>기본값: 10</p> <p>ASA 가상 메트릭이 이 값보다 작으면 축소(Scale-In)가 트리거됩니다.</p> <p><a href="#">Auto Scale 논리, 26 페이지</a>의 내용을 참조하십시오.</p>	해당 없음

매개 변수 이름	허용되는 값 / 유형	설명	리소스 생성 유형
scaleOutThreshold	문자열	<p>의 확장 임계값입니다.</p> <p>기본값: 80</p> <p>메트릭이 이 값을 초과하면 스케일 아웃이 트리거됩니다. ASA 가상</p> <p>‘scaleOutThreshold’는 항상 ‘scaleInThreshold’보다 커야 합니다.</p> <p><a href="#">Auto Scale 논리, 26 페이지</a>의 내용을 참조하십시오.</p>	해당 없음
minAsaCount	정수	<p>지정된 시간에 설정된 확장 집합에서 사용 가능한 최소 ASA 가상 인스턴스.</p> <p>예: 2</p>	해당 없음
maxAsaCount	정수	<p>확장 집합에서 허용되는 최대 ASA 가상 인스턴스 수입니다.</p> <p>예: 10</p> <p>참고 <a href="#">Auto Scale 논리</a>는 이 변수의 범위를 확인하지 않으므로 신중하게 입력하십시오.</p>	해당 없음
metricsAverageDuration	정수	<p>드롭다운에서 선택</p> <p>이 숫자는 메트릭이 평균화되는 시간(분)을 나타냅니다.</p> <p>이 변수의 값이 5(즉, 5)인 경우, Auto Scale Manager가 예약되면 메트릭의 지난 5분 평균을 확인하고 이를 기반으로 하여 확장 결정을 내립니다.</p> <p>참고 <a href="#">Azure 제한</a>으로 인해 숫자 1, 5, 15, 30만 유효합니다.</p>	해당 없음

매개 변수 이름	허용되는 값 / 유형	설명	리소스 생성 유형
initDeploymentMode	일괄(BULK) / 단계별(STEP)	기본적으로 첫 번째 구축 또는 확장 집합에 ASA 가상 인스턴스가 포함되지 않은 경우에 적용됩니다.  일괄(BULK): Auto Scale Manager가 한 번에 'minAsaCount'개의 ASA 가상 인스턴스 수를 동시에 구축하려고 시도합니다.  단계별(STEP): Auto Scale Manager는 예약된 간격마다 하나씩 'minAsaCount'개의 ASA 가상 디바이스를 구축합니다.	
configurationFile	문자열	ASA 가상 컨피그레이션 파일의 전체 경로.  예: https : //myserver/asavconfig/asaconfig.txt	해당 없음
* Azure에는 새 리소스의 명명 규칙에 제한 사항이 있습니다. 제한 사항을 검토하거나 간단히 모두 소문자를 사용하십시오. 공백이나 특수 문자는 사용하지 마십시오.			

## Auto Scale 구축

### Auto Scale ARM 템플릿 구축

ASA 가상 Auto Scale for Azure에서 요구하는 리소스를 ARM 템플릿을 사용하여 구축합니다. 지정된 리소스 그룹 내에서 ARM 템플릿 구축은 다음을 생성합니다.

- VMSS(Virtual Machine Scale Set)
- 외부 로드 밸런서
- 내부 로드 밸런서
- Azure Function 앱
- Logic 앱
- 보안 그룹 (데이터 및 관리 인터페이스용)

시작하기 전에

- GitHub 리포지토리(<https://github.com/CiscoDevNet/cisco-asav>)에서 ARM 템플릿 `azure_asav_autoscale.json`을 다운로드합니다.

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

예제:

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

이 예에서는 3개의 영역이 있는 "Central US" 지역을 보여줍니다.

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

예제:

```
{
  "type": "Microsoft.Network/loadBalancers",
  "name": "[variables('elbName')]",
  "location": "[resourceGroup().location]",
  "apiVersion": "2018-06-01",
  "sku": {
    "name": "Standard"
  },
  "dependsOn": [
    "[concat('Microsoft.Network/publicIPAddresses/', variables('elbPublicIpName'))]"
  ],
  "properties": {
    "frontendIPConfigurations": [
      {
        "name": "LoadBalancerFrontEnd",
        "properties": {
          "publicIPAddress": {
            "id": "[resourceId('Microsoft.Network/publicIPAddresses/',
variables('elbPublicIpName'))]"
          }
        }
      }
    ],
    "backendAddressPools": [
      {
        "name": "backendPool"
      }
    ],
    "loadBalancingRules": [
      {
        "properties": {
          "frontendIPConfiguration": {
            "Id": "[concat(resourceId('Microsoft.Network/loadBalancers', variables('elbName')),
'/frontendIpConfigurations/LoadBalancerFrontend')]"
          }
        }
      }
    ]
  }
}
```

```

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

```

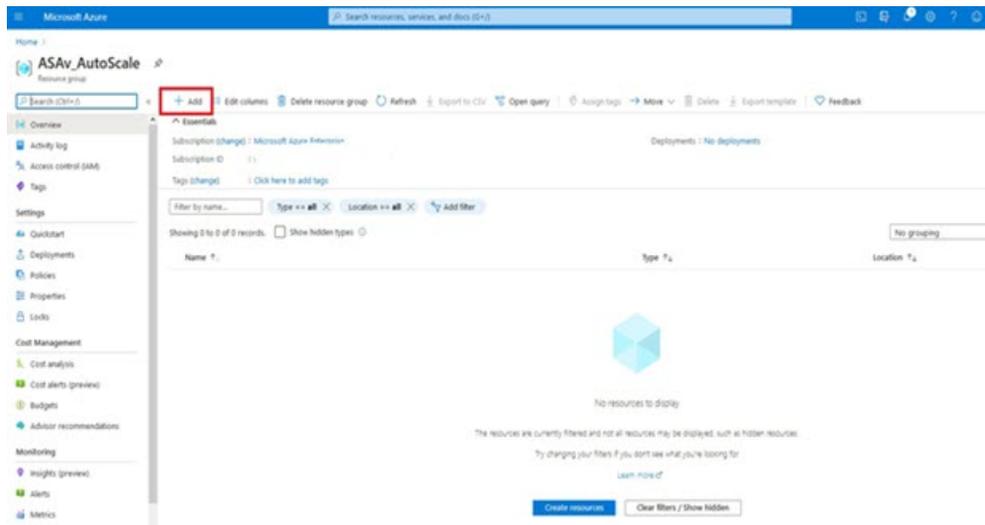
참고 이 파일을 편집하지 않으려는 경우 Azure 포털 구축후(post-deployment)에서 이를 편집할 수도 있습니다.

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

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

새 리소스 그룹을 생성하거나 기존의 빈 리소스 그룹을 선택합니다(예:ASA 가상 *\_AutoScale*).

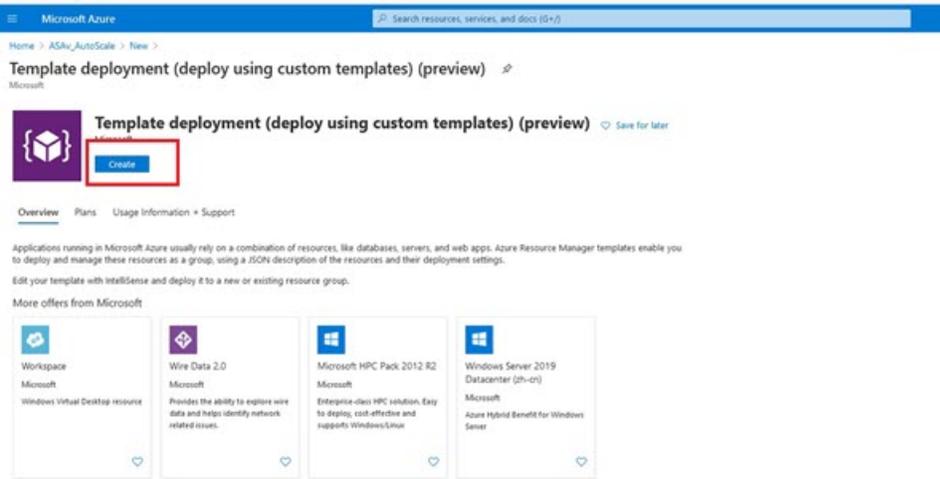
그림 2: Azure Portal



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.

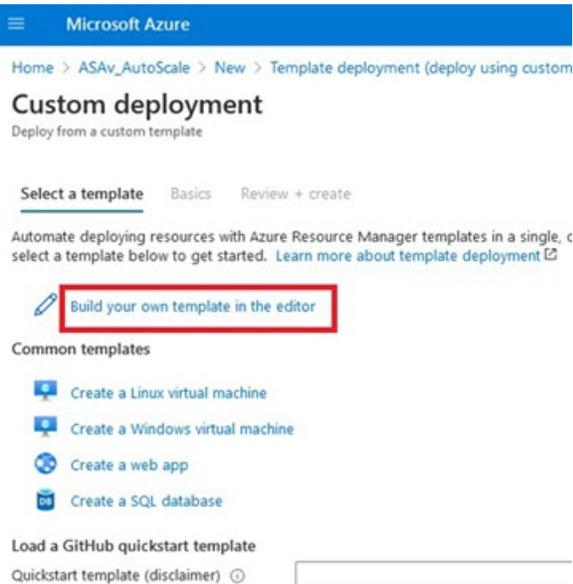
그림 3: 맞춤형 템플릿 구축



단계 7 **Create**(생성)를 클릭합니다.

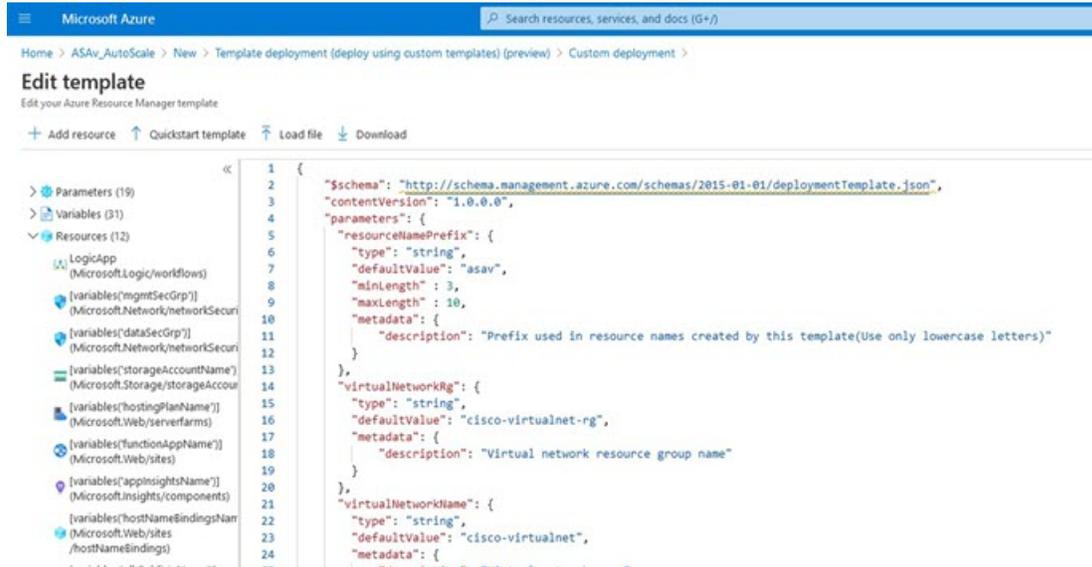
단계 8 템플릿을 생성하기 위한 몇 가지 옵션이 있습니다. **Build your own template in editor**(편집기에서 자체 템플릿 구축)를 선택합니다.

그림 4: 자체 템플릿 만들기



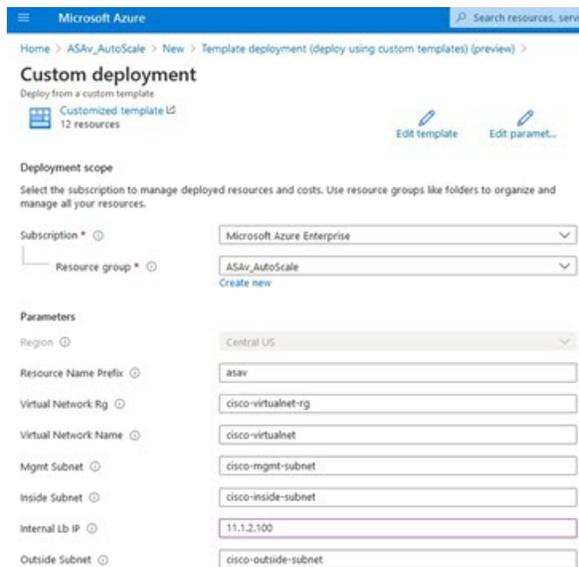
단계 9 **Edit template**(템플릿 편집) 창에서 모든 기본 콘텐츠를 삭제하고 업데이트된 *azure\_asav\_autoscale.json*에서 콘텐츠를 복사하고 **Save**(저장)를 클릭합니다.

그림 5: 템플릿 수정



단계 10 다음 섹션에서 모든 매개변수를 입력합니다. 각 매개변수에 대한 자세한 내용은 [입력 매개변수](#), 7 페이지를 참조한 다음 **Purchase**(구매)를 클릭하십시오.

그림 6: ARM 템플릿 매개변수

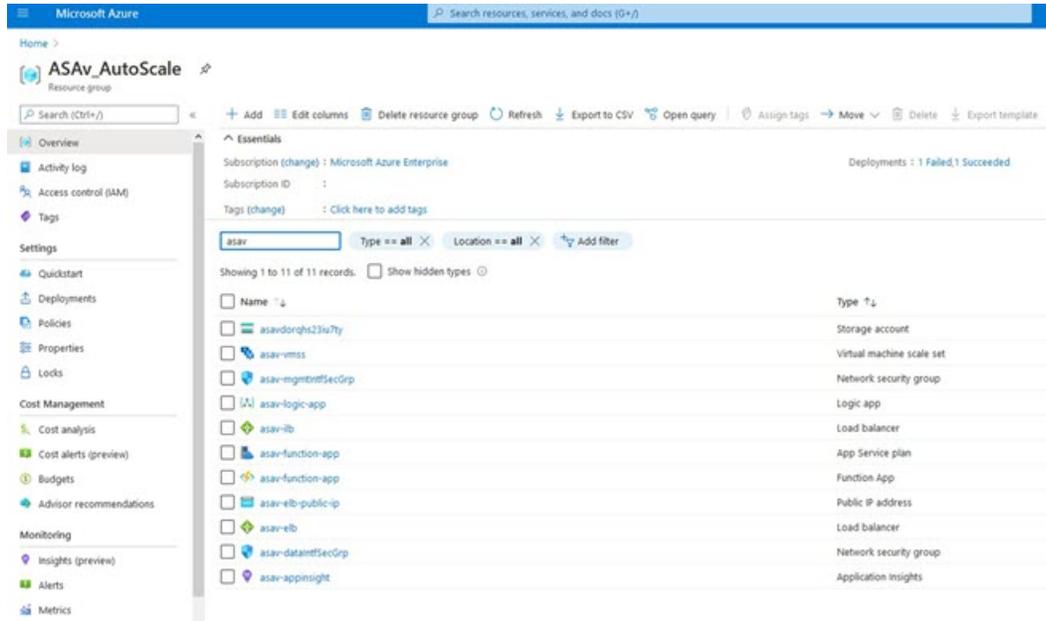


참고 **Edit Parameters**(매개변수 편집)를 클릭하고 JSON 파일을 편집하거나 미리 채워진 내용을 업로드할 수도 있습니다.

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

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

그림 7: ASA Virtual Auto Scale Template 구축



## Azure Function 앱 구축

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

시작하기 전에

- *ASM\_Function.zip* 패키지를 빌드합니다. [소스 코드로 Azure 기능 빌드, 28 페이지](#)의 내용을 참조하십시오.

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

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

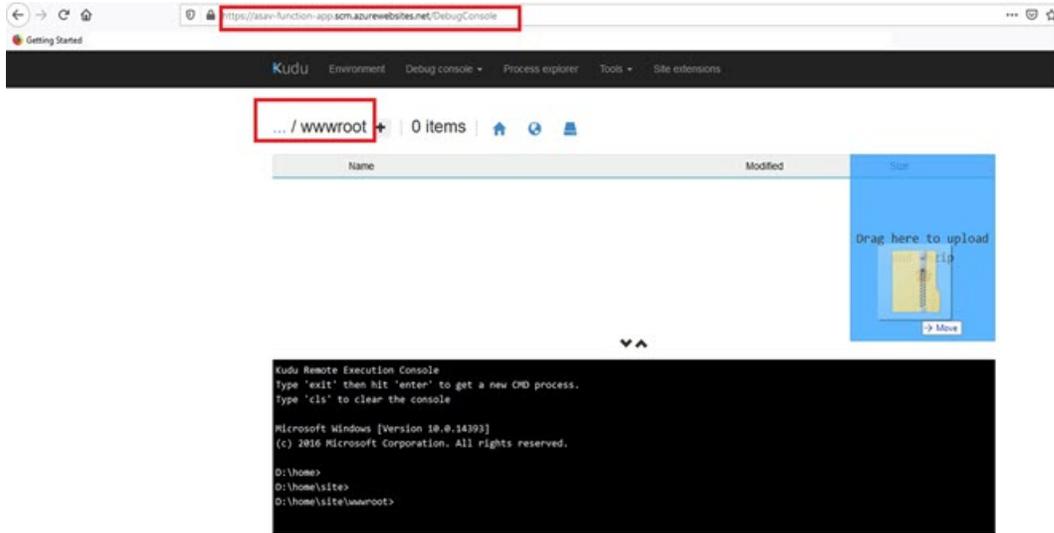
[Auto Scale ARM 템플릿 구축, 11 페이지](#)의 예:

<https://asav-function-app.scm.azurewebsites.net/DebugConsole>

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

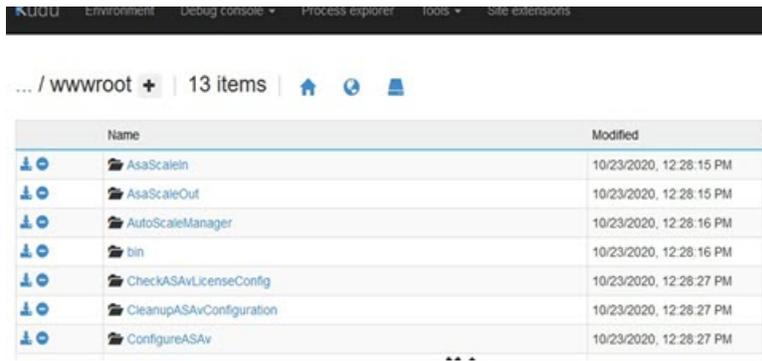
**단계 3** **ASM\_Function.zip**을 파일 탐색기의 오른쪽 모서리로 끌어다 놓습니다.

그림 8: ASA Virtual Auto Scale 기능 업로드



단계 4 업로드에 성공하면 모든 서버리스 함수가 표시됩니다.

그림 9: ASA 가상 서버리스 기능

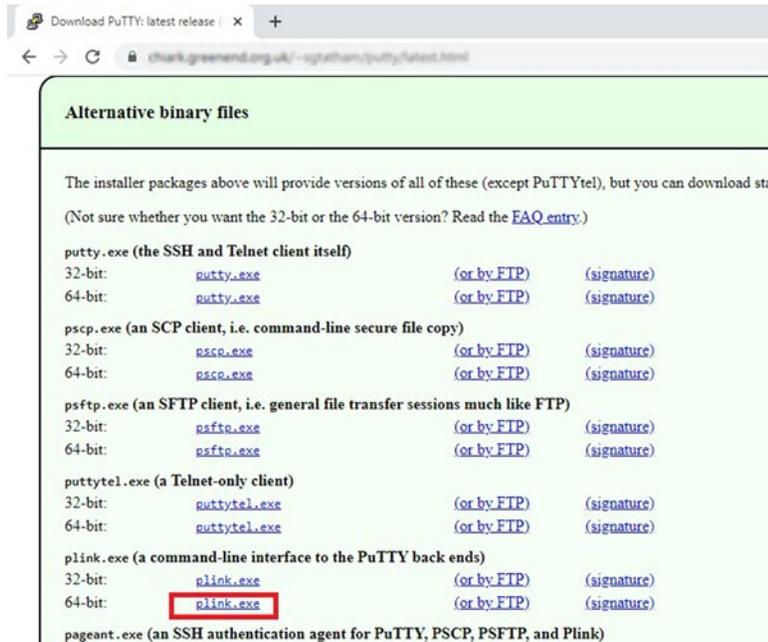


단계 5 PuTTY SSH 클라이언트를 다운로드합니다.

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

[www.putty.org](http://www.putty.org)에서 PuTTY 명령줄 인터페이스를 PuTTY 백엔드(*plink.exe*)에 다운로드합니다.

그림 10: PuTTY 다운로드



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

단계 7 파일 탐색기의 오른쪽 모서리, 즉 이전 단계에서 **ASM\_Function.zip**이 업로드된 위치에 **asassh.exe**를 끌어다 놓습니다.

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

## 컨피그레이션 조정

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

시작하기 전에

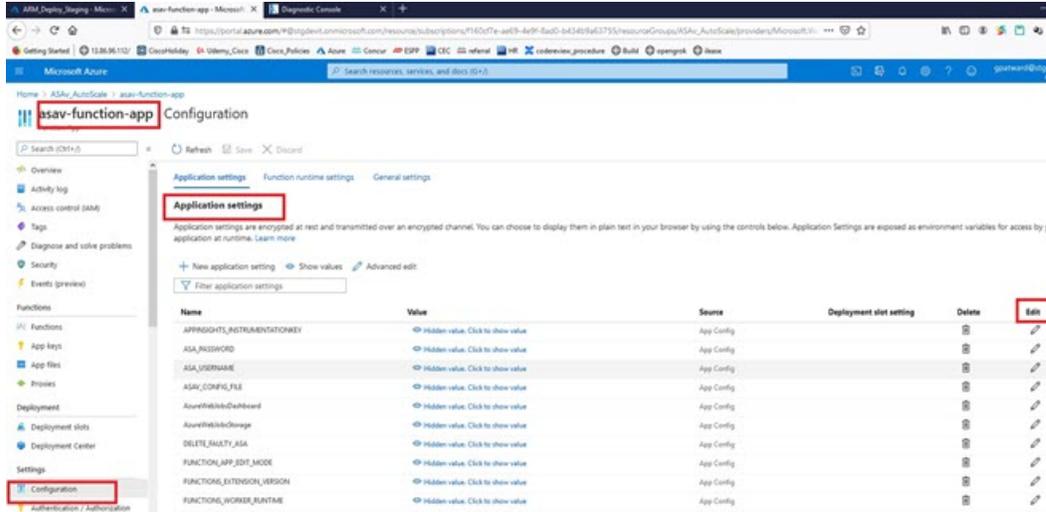


참고 이젠 언제든지 수정할 수 있습니다. 컨피그레이션을 수정하려면 이 순서를 따릅니다.

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

단계 1 Azure Portal에서 ASA 가상 함수 애플리케이션을 검색하여 선택합니다.

그림 11: ASA 가상 기능 애플리케이션



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

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

- 컨피그레이션 이름: “DELETE\_FAULTY\_ASA”(기본값: YES)

확장 중에 새 ASA 가상 인스턴스가 시작되고 컨피그레이션 파일을 통해 구성됩니다. 컨피그레이션이 실패할 경우 이 옵션을 기반으로 Auto Scale Manager는 해당 ASA 가상 인스턴스를 유지하거나 삭제할지 결정합니다. (예: 결합 ASA 가상 삭제 / 아니요: 컨피그레이션이 실패하더라도 ASA 가상 인스턴스를 유지합니다.)

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

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

참고 Azure 문서를 검색하여 애플리케이션 데이터를 보호하는 모범 사례를 찾습니다.

## 가상 시스템 확장 집합의 IAM 역할 구성

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

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

단계 1 Azure 포털에서 VMSS로 이동합니다.

단계 2 액세스 제어(IAM)를 클릭합니다.

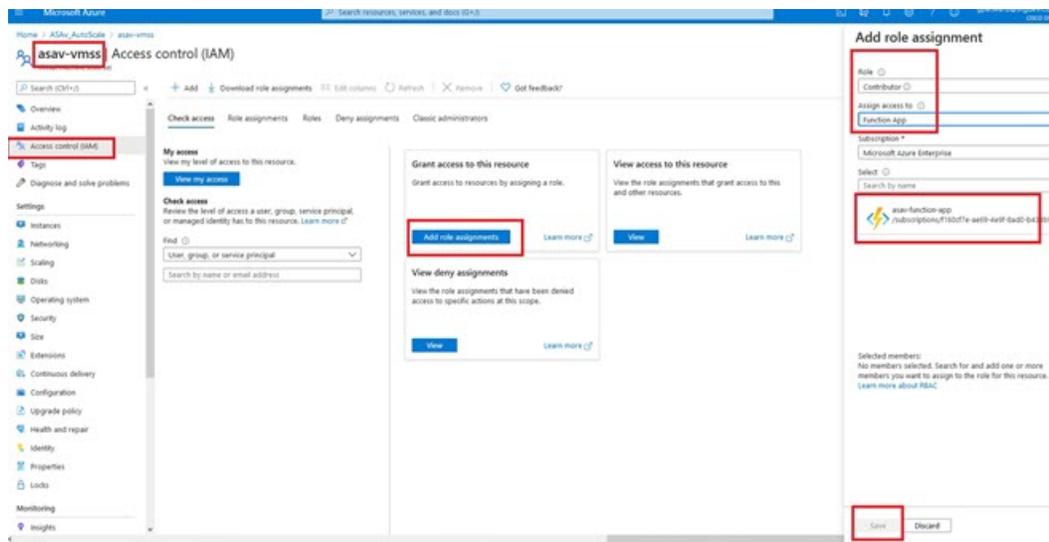
단계 3 Add(추가)를 클릭하여 역할 할당을 추가합니다.

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

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

단계 6 ASA 가상 함수 애플리케이션을 선택합니다.

그림 12: AIM 역할 할당



단계 7 Save(저장)를 클릭합니다.

참고 또한 아직 시작된 ASA 가상 인스턴스가 없는지 확인해야 합니다.

## 보안 그룹 업데이트

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

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

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

## Azure Logic 앱 업데이트

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

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

중요 계속하기 전에 이 단계를 모두 읽고 숙지하십시오.

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

- 필수: "SUBSCRIPTION\_ID"의 모든 어커런스를 찾아서 구독 ID 정보로 교체합니다.
- 필수: "RG\_NAME" 어커런스를 모두 찾아서 리소스 그룹 이름으로 바꿉니다.
- 필수: "FUNCTIONAPPNAME" 어커런스를 모두 찾아서 함수 앱 이름으로 바꿉니다.

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

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

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

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

- e) (선택 사항) 드레인 시간을 수정하거나 기본값(5)을 유지합니다. 이는 축소(Scale-In) 작업 중에 디바이스를 삭제하기 전에 ASA 가상에서 기존 연결을 드레인하는 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"
          }
        }
      }
    }
  }
}
```

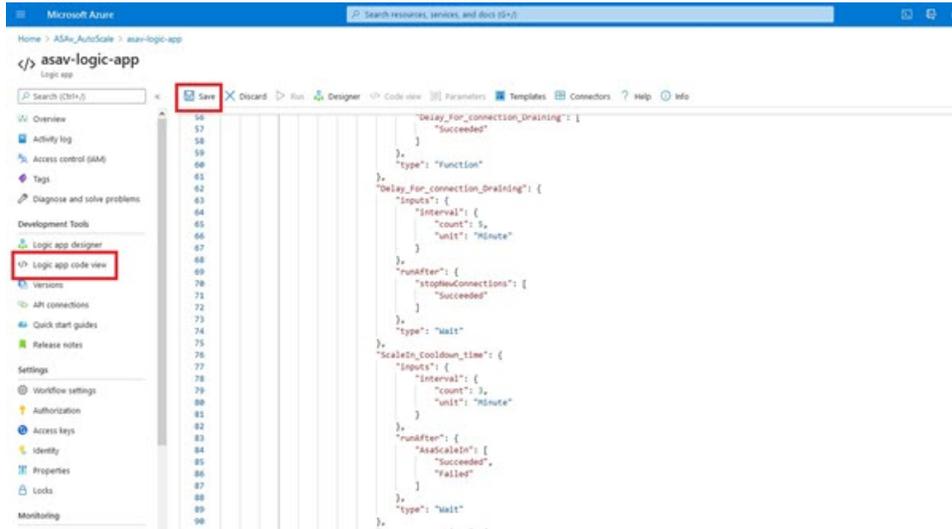
- f) (선택 사항) 냉각 시간을 수정하거나 기본값(10)을 유지합니다. 이 시간은 확장(Scale-Out)이 완료된 후 작업 없음을 유지하는 시간입니다. 다음 예는 *LogicApp.txt* 파일에서 이러한 행을 보여줍니다.

```
"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.

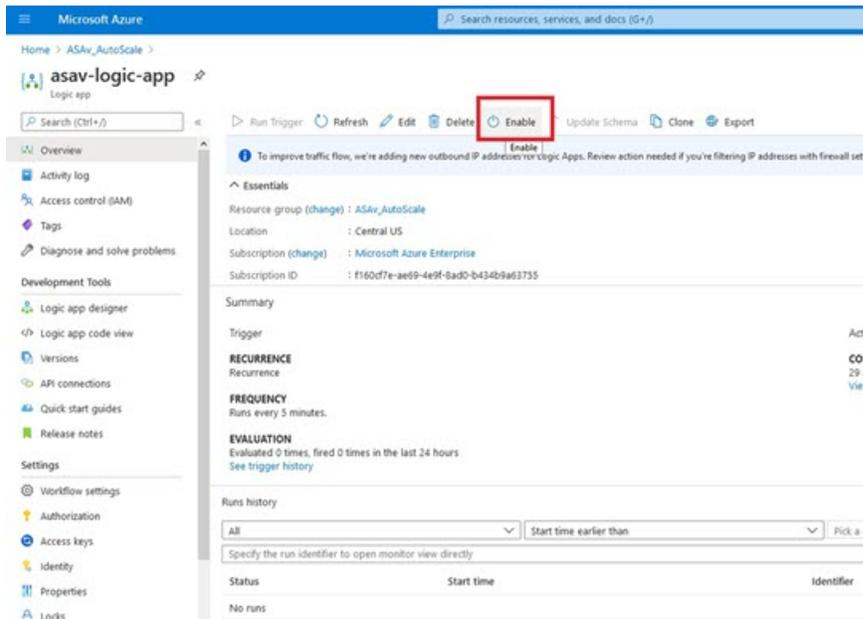
단계 2 Logic 앱 코드 보기로 이동하여 기본 콘텐츠를 삭제하고 수정된 *LogicApp.txt* 파일에서 콘텐츠를 붙여넣고 **Save**(저장)을 클릭합니다.

그림 13: Logic 앱 코드 보기



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

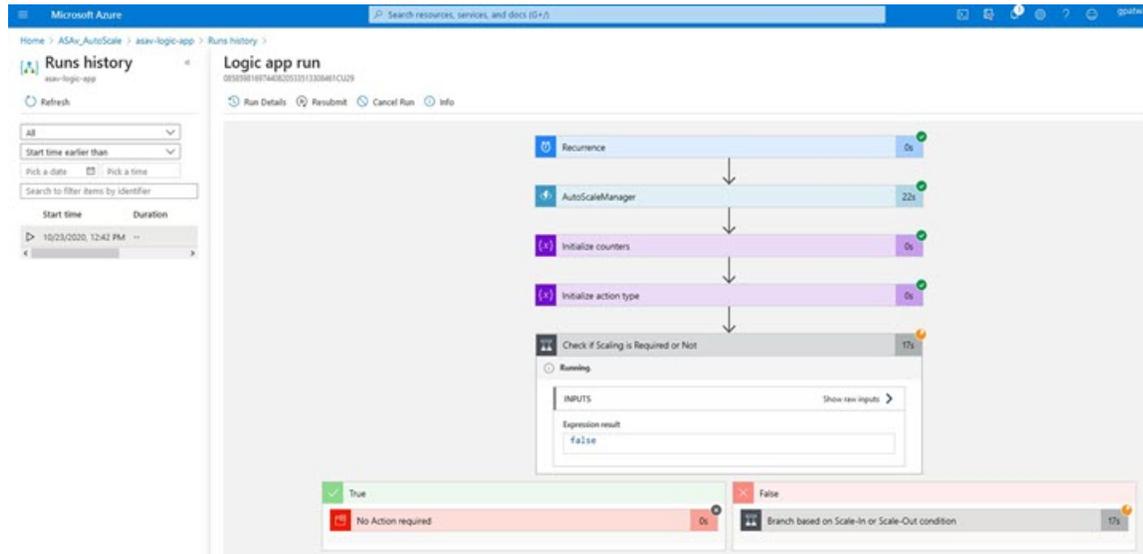
그림 14: Logic 앱 활성화



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

## Threat Defense VirtualASA 가상 업그레이드

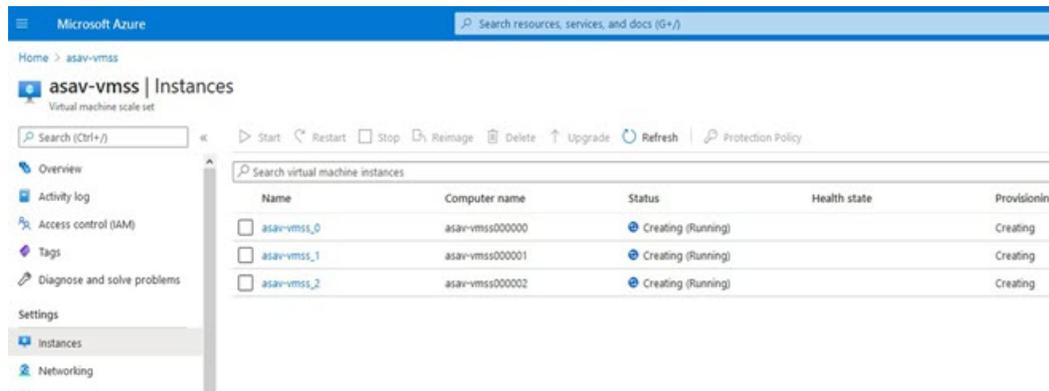
그림 15: Logic 앱 실행 상태



단계 5 Logic 앱이 시작되면 모든 구축 관련 단계가 완료됩니다.

단계 6 VMSS에서 ASA 가상 인스턴스가 생성되고 있는지 확인합니다.

그림 16: 실행 중인 ASA 가상 인스턴스



이 예에서는 ARM 템플릿 구축에서 'minAsaCount'가 '3'으로, 'initDeploymentMode'가 'BULK'로 설정되었으므로 3개의 ASA 가상 인스턴스가 시작됩니다.

## Threat Defense VirtualASA 가상 업그레이드

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



참고 모든 REST 클라이언트를 사용하여 ASA 가상을 업그레이드할 수 있습니다.

시작하기 전에

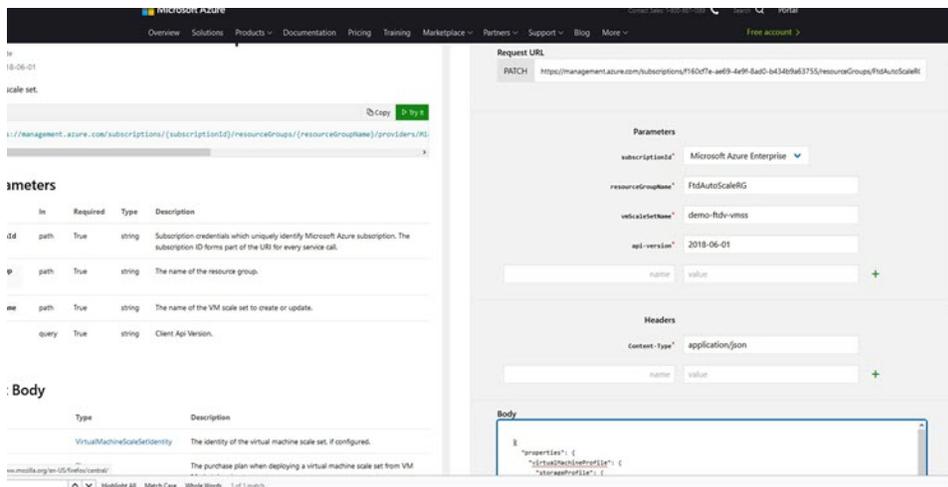
- 마켓플레이스에서 사용 가능한 새 ASA 가상 이미지 버전을 가져옵니다(예: 914.001).
- 원래 스케일 세트를 구축하는 데 사용된 SKU를 가져옵니다(예: asav-azure-byol).
- 리소스 그룹 및 가상 시스템 확장 집합 이름을 가져옵니다.

단계 1 브라우저에서 다음 URL로 이동합니다.

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

단계 2 매개변수 섹션에 세부 사항을 입력합니다.

그림 17: Threat Defense VirtualASA 가상 업그레이드



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

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

단계 4 Azure의 성공적인 응답은 VMSS가 변경 사항을 수락했음을 의미합니다.

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

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

## Auto Scale 논리

### 확장 논리

- **POLICY-1:** 어떤 경우든 ASA 가상 평균로드가 구성된 기간 동안 확장 임계값을 초과하면 확장(Scale-Out)이 트리거됩니다.
- **POLICY-2:** 구성된 기간 동안 모든 ASA 가상 디바이스의 평균로드가 확장 임계값을 초과하면 확장(Scale-Out)이 트리거됩니다.

### 축소 논리

- 모든 ASA 가상 디바이스의 CPU 사용률이 구성된 기간 동안 구성된 축소 임계값 미만인 경우.

### Notes(참고)

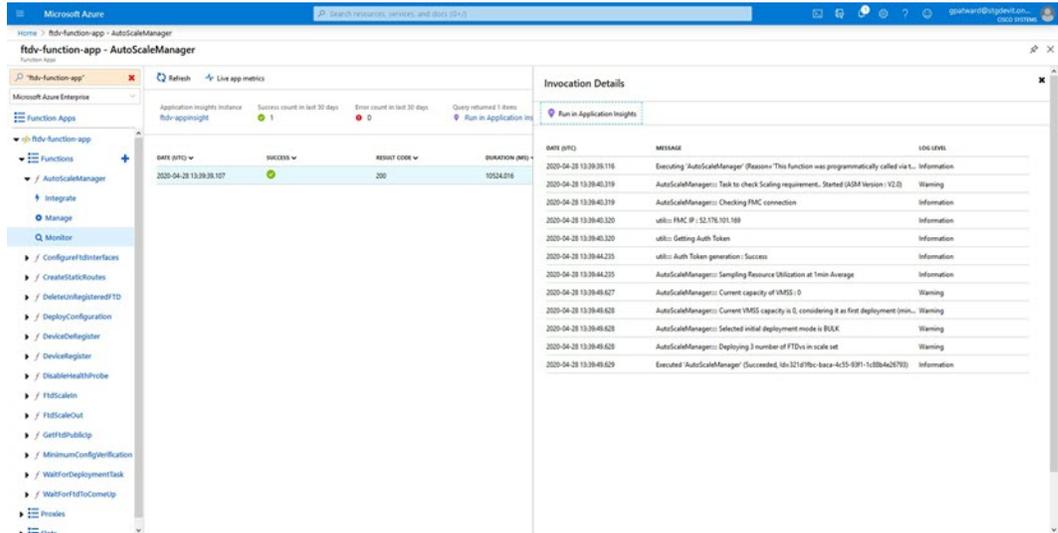
- 축소(Scale-In)/확장(Scale-Out)은 1단계로 수행됩니다(즉, 한 번에 1개 ASA 가상만 축소/확장).
- 위의 논리는 로드 밸런서가 모든 ASA 가상 디바이스에 연결을 동일하게 분산하려고 시도한다는 가정을 기반으로 하며 평균적으로 모든 ASA 가상 디바이스를 동일하게 로드해야 합니다.

## Auto Scale 로깅 및 디버깅

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

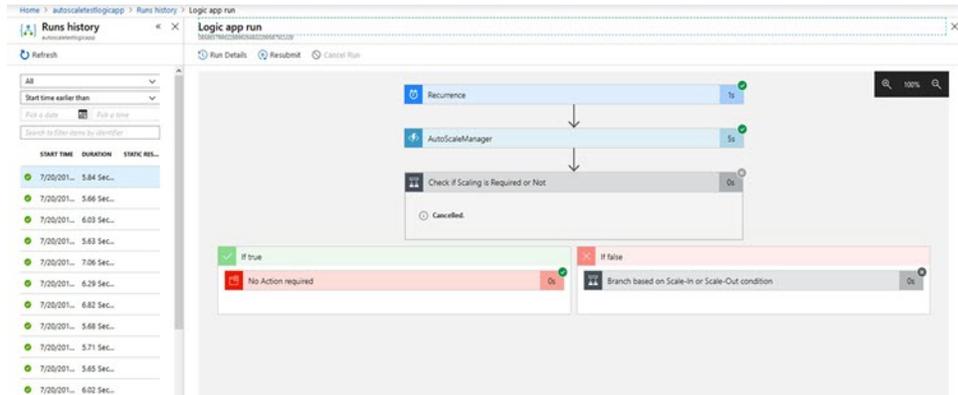
- 개별 Azure 함수의 로그를 볼 수 있습니다.

그림 18: Azure Function 로그



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

그림 19: Logic 앱 실행 로그



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

## Auto Scale 지침 및 제한 사항

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

- 확장 결정은 CPU 사용률을 기반으로 합니다.

- ASA 가상 Management 인터페이스가 공용 IP 주소를 갖도록 구성되었습니다.
- IPv4만 지원됩니다.
- ARM 템플릿은 입력 검증 기능이 제한되어 있으므로 올바른 입력 검증을 제공하는 것은 사용자의 책임입니다.
- Azure 관리자는 Function 앱 환경 내에서 민감한 데이터(예: 관리자 로그인 자격 증명 및 비밀번호)를 일반 텍스트 형식으로 볼 수 있습니다. Azure Key Vault 서비스를 사용하여 민감한 데이터를 보호할 수 있습니다.

## Auto Scale 문제 해결

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

- ASA 가상로 SSH할 수 없음: 템플릿을 통해 복잡한 비밀번호가 ASA 가상에 전달되는지 확인합니다. 보안 그룹에서 SSH 연결을 허용하는지 확인하십시오.
- 로드 밸런서 상태 확인 실패: ASA 가상에서 데이터 인터페이스의 SSH에 응답하는지 확인합니다. 보안 그룹 설정을 확인합니다.
- 트래픽 문제: 로드 밸런서 규칙, NAT 규칙 / ASA 가상에 구성된 고정 경로를 확인합니다. 템플릿 및 보안 그룹 규칙에 제공된 Azure 가상 네트워크 / 서브넷 / 게이트웨이 세부 정보를 확인합니다.
- Logic 앱이 VMSS에 액세스하지 못함: VMSS의 IAM 역할 컨피그레이션이 올바른지 확인하십시오.
- Logic 앱이 매우 오랫동안 실행 됨: 확장된 ASA 가상 디바이스에서 SSH 액세스를 확인합니다. Azure VMSS에서 ASA 가상 디바이스의 상태를 확인합니다.
- 구독 ID와 관련된 오류 발생 Azure Function: 계정에서 기본 구독이 선택되었는지 확인하십시오.
- 축소(Scale-In) 작업 실패: 경우에 따라 Azure에서 인스턴스를 삭제하는 데 시간이 오래 걸리는 경우가 있습니다. 이러한 상황에서는 축소 작업이 시간 초과되고 오류를 보고할 수 있지만 결국엔 인스턴스가 삭제됩니다.
- 컨피그레이션 변경을 수행하기 전에 논리 애플리케이션을 비활성화하고 실행 중인 모든 작업이 완료될 때까지 기다리십시오.

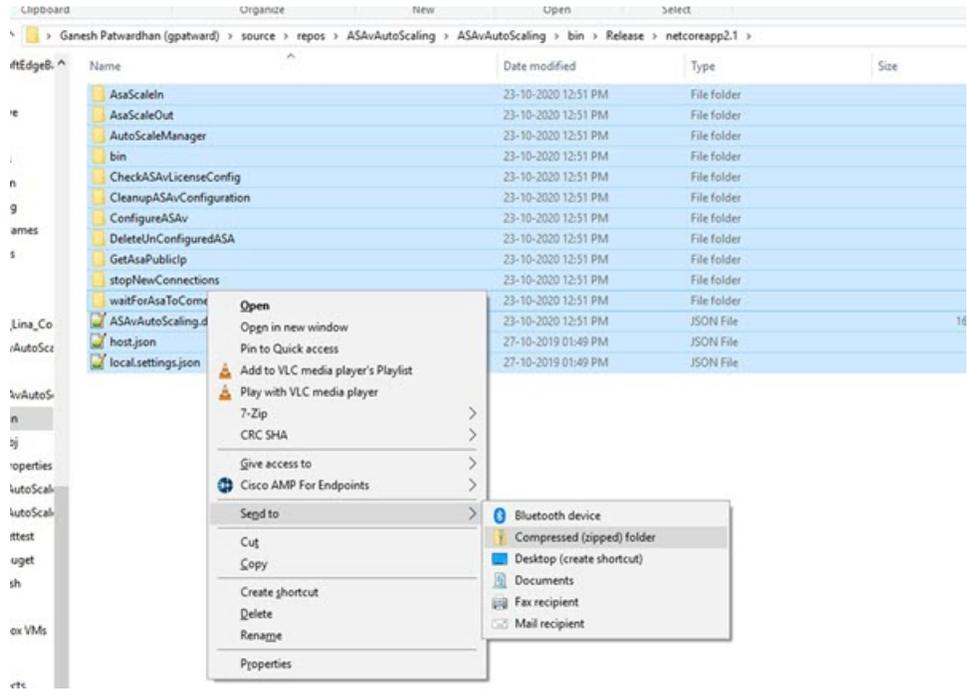
## 소스 코드로 Azure 기능 빌드

시스템 요구 사항

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



그림 21: Build ASM\_Function.zip



## 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.