GCP에 ASA 가상 Auto Scale 솔루션 구축

GCP의 ASA 가상용 Auto Scale 솔루션

다음 섹션에서는 GCP의 자동 확장 솔루션 구성 요소가 ASA 가상에서 작동하는 방식을 설명합니다.

Auto Scale 솔루션

ASA 가상 Auto Scale for GCP는 GCP에서 제공하는 서버리스 인프라(Cloud Functions, 로드 밸런서, Pub/Sub, 인스턴스 그룹 등)를 사용하는 완벽한 서버리스 구현입니다.

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

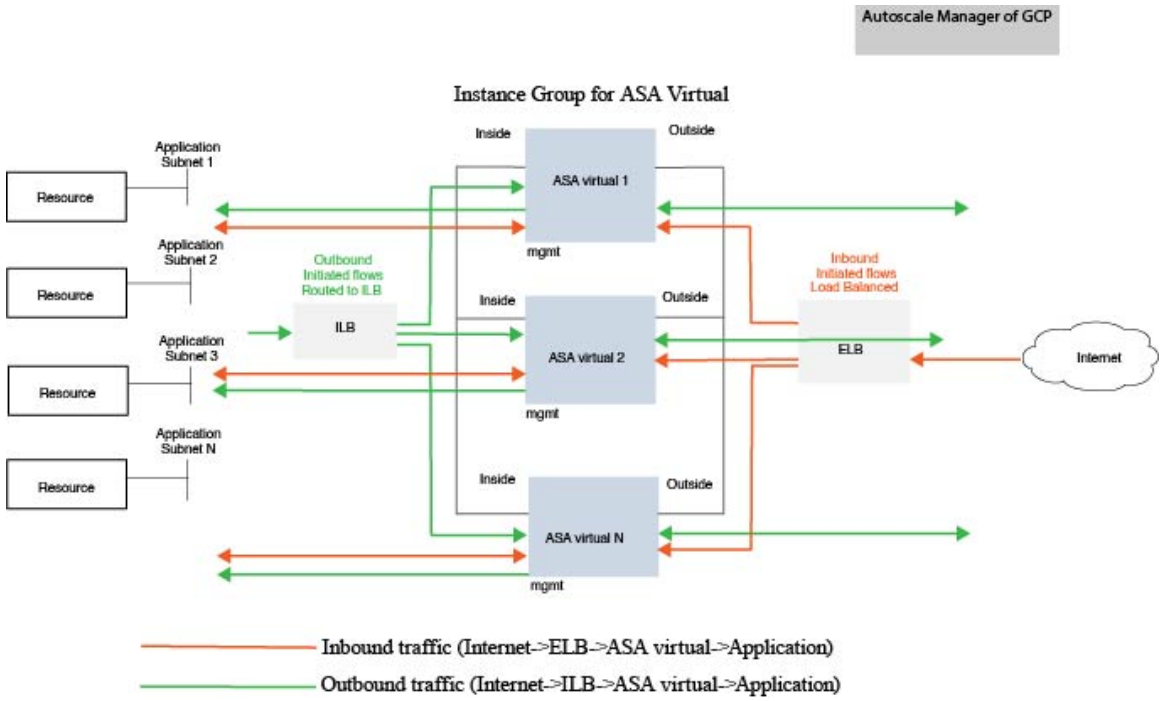
- GCP Deployment Manager 템플릿 기반 구축.
- CPU 기반의 메트릭 확장 지원.
- ASA 가상 구축 및 다중 가용성 영역 지원
- 확장된 ASA 가상 인스턴스에 자동으로 적용되는 완전히 자동화된 컨피그레이션.
- 로드 밸런서 및 다중 가용성 영역 지원
- Cisco에서는 구축을 쉽게 수행할 수 있도록 Auto Scale for GCP 구축 패키지를 제공합니다.

Auto Scale 사용 사례

ASA 가상 Auto Scale for GCP는 GCP Azure 내부 로드 밸런서(ILB)와 GCP 외부 로드 밸런서(ELB) 사이에 ASA 가상 인스턴스 그룹을 배치하는 자동화된 수평 확장 솔루션입니다.

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

그림 ***1: ASA*** 가상 ***Auto Scale*** 사용 사례



범위

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

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

구축 패키지 다운로드

ASA 가상 Auto Scale for GCP 솔루션은 GCP에서 제공하는 서버리스 인프라(Cloud Functions, 로드 밸런서, Pub/Sub, 인스턴스 그룹 등)를 사용하는 GCP 구축 관리자 템플릿 기반 구축입니다.

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

주의 Cisco에서 제공하는 자동 확장용 구축 스크립트 및 템플릿은 오픈 소스 예시로 제공되며 일반적인 Cisco TAC 지원 범위에서는 다루지 않습니다.

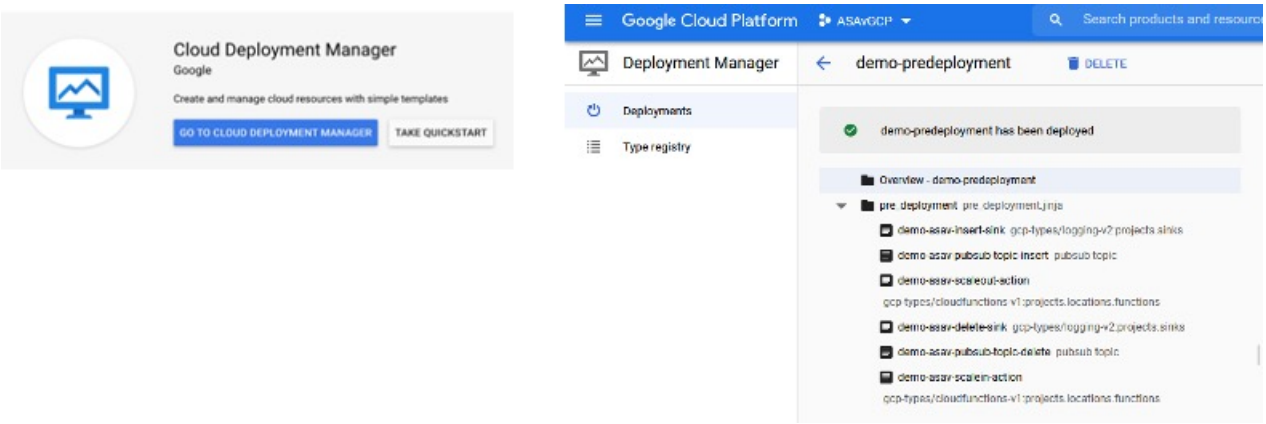
Auto Scale 솔루션 구성 요소

다음 구성 요소가 ASA 가상 Auto Scale for GCP 솔루션을 구성합니다.

구축 관리자

- 컨피그레이션을 코드로서 처리하고 반복 가능한 구축을 수행합니다. Google Cloud Deployment Manager를 사용하면 YAML을 이용해 애플리케이션에 필요한 모든 리소스를 선언적 형식으로 지정할 수 있습니다. 또한 Python 또는 Jinja2 템플릿을 사용하여 구성을 매개변수화하고 공통 구축 패러다임을 재사용할 수 있습니다.
- 리소스를 정의하는 구성 파일을 생성합니다. 이러한 리소스를 생성하는 프로세스를 반복하여 일관된 결과를 얻을 수 있습니다. 자세한 내용은 https://cloud.google.com/deployment-manager/docs 를 참조하십시오.

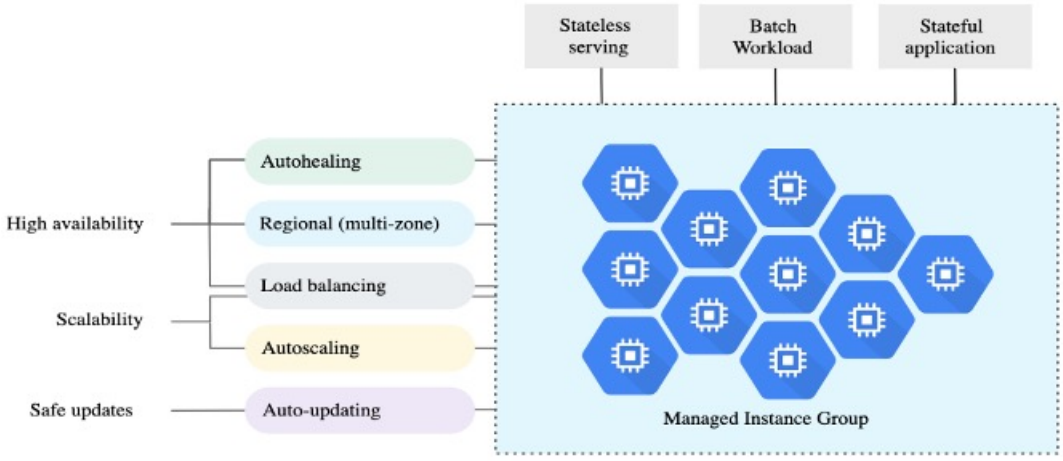
그림 *2: Deployment Manager* 보기



GCP의 Managed Instance Group

MIG(Managed Instance Group)는 사용자가 지정하는 인스턴스 템플릿 및 선택적 스테이트풀 구성을 기반으로 각 매니지드 인스턴스를 생성합니다. 자세한 내용은 https://cloud.google.com/compute/docs/instance-groups를 참조하십시오.

그림 *3:* 인스턴스 그룹 기능

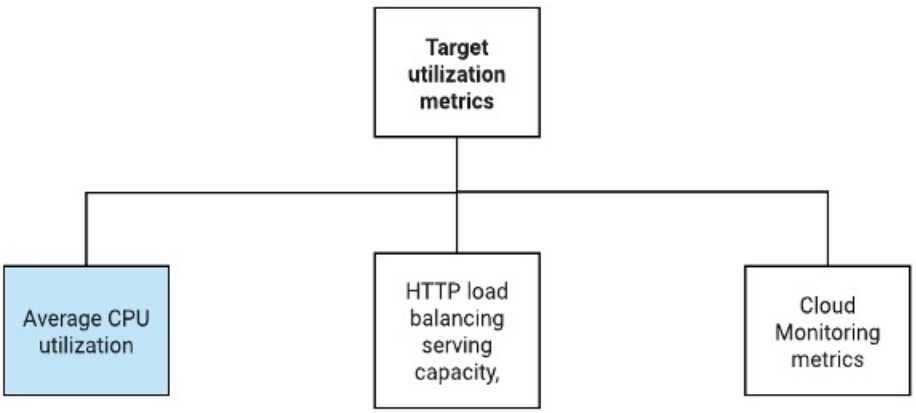


목표 사용률 메트릭

- 다음 다이어그램은 목표 사용률 메트릭을 보여줍니다. 자동 확장 결정에는 평균 CPU 사용률 메트릭만 사용합니다.
- Autoscaler는 선택한 사용률 메트릭을 기반으로 사용량 정보를 지속적으로 수집하고, 실제 사용률을 원하는 목표 사용률과 비교하며, 이 정보를 사용하여 그룹에서 인스턴스를 제거해야 하는지(축소) 추가해야 하는지(확장)를 결정합니다.

- 목표 사용률 수준은 VM(가상 머신) 인스턴스를 유지 관리할 수준입니다. 예를 들어 CPU 사용률을 기준으로 확장하는 경우 목표 사용률 레벨을 75%로 설정할 수 있으며, 이 경우 Autoscaler는 지정된 인스턴스 그룹의 CPU 사용률을 75% 이하로 유지합니다. 각 메트릭의 사용률 수준은 자동 확장 정책에 따라 다르게 해석됩니다. 자세한 내용은 https://cloud.google.com/compute/docs/autoscaler를 참조하십시오.

그림 *4:* 목표 사용률 메트릭



서버리스 클라우드 기능

Instance Group Manager에 인스턴스가 나타나면, 사용자는 SSH 비밀번호를 설정하고, 비밀번호를 활성화하고, 호스트 이름을 변경하기 위한 Google Cloud 기능을 사용합니다.

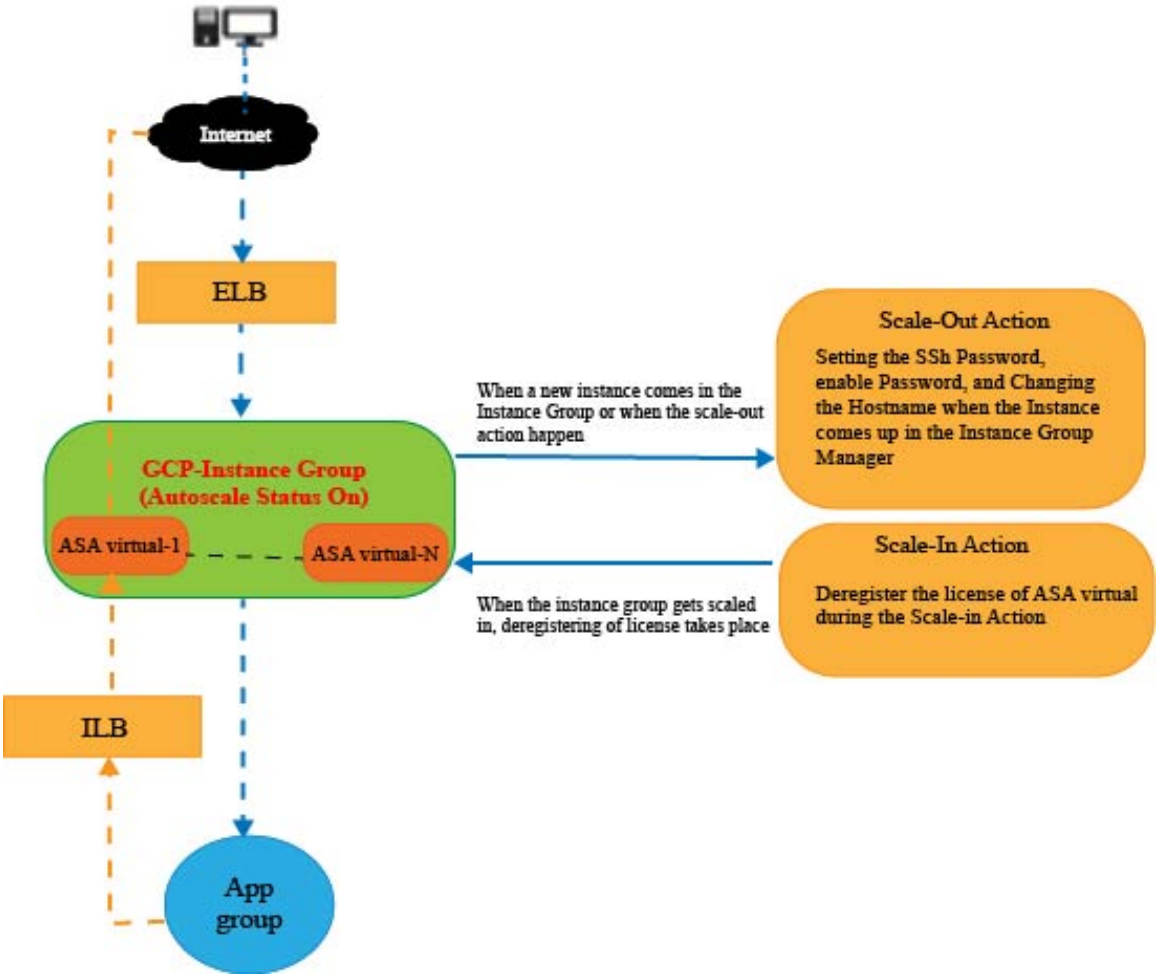
- 확장 중에 인스턴스 그룹에 새 ASA 가상 인스턴스가 나타나면, 확장 프로세스를 항상 모니터링할 수는 없으므로 SSH 비밀번호를 설정하고 비밀번호를 활성화한 다음 호스트 이름을 변경해야 합니다.
- 클라우드 기능은 확장 프로세스 중에 Cloud Pub/Sub 주제를 통해 트리거됩니다. 확장하는 동안 인스턴스를 추가할 때만 사용 가능한 필터가 있는 Log Sink도 있습니다.

Cloud Functions를 사용하여 서버리스 라이선스 등록 해제

- 축소 중에 인스턴스가 삭제되는 동안, ASA 가상 인스턴스에서 라이선스를 등록 해제해야 합니다.
- 클라우드 기능은 Cloud Pub/Sub 주제를 통해 트리거됩니다. 특히 삭제 프로세스의 경우, 축소하는 동안 인스턴스 삭제에만 사용 가능한 필터가 있는 Log Sink가 있습니다.
- Cloud Function은 트리거되면 SSH를 통해 삭제 중인 ASA 가상 인스턴스에 SSH를 수행하고 라이선스 등록 취소를 위한 명령을 실행합니다.

Autoscale 솔루션의 대략적인 개요

그림 ***5: Autoscale*** 솔루션 개요



Auto Scale 솔루션 사전 요건

GCP 리소스

GCP 프로젝트

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

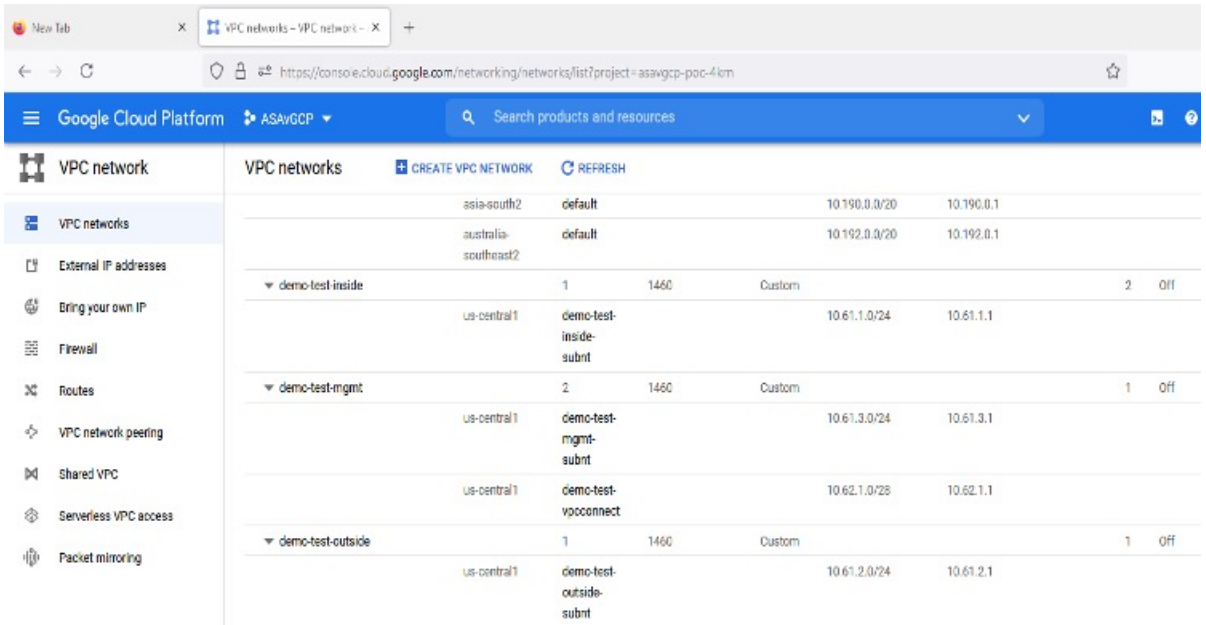
네트워킹

VPC 3개가 사용 가능한지/생성되었는지 확인합니다. Auto Scale 구축에서는 네트워킹 리소스를 생성, 변경 또는 관리하지 않습니다.

ASA 가상에는 네트워크 인터페이스 3개가 필요하므로 가상 네트워크에는 다음을 위한 서브넷 3개가 필요합니다.

- 관리 트래픽
- 내부 트래픽
- 외부 트래픽

그림 ***6: VPC*** 네트워크 보기



방화벽

VPC 간 통신을 허용하고 상태 프로브도 허용하는 방화벽 규칙을 만들어야 합니다. 나중에 구축 관리자 템플릿에서 사용할 방화벽 태그를 기록해 두어야 합니다.

서브넷이 연결된 네트워크 보안 그룹에서 다음 포트를 열어야 합니다.

- SSH(TCP/22) - 로드 밸런서와 ASA 가상 사이의 상태 프로브에 필요합니다. 서버리스 함수와 ASA 가상 간의 통신에 필요합니다.
- 애플리케이션 전용 프로토콜/포트 - 모든 사용자 애플리케이션(예: TCP/80)에 필요합니다.

ASA 컨피그레이션 파일 준비

deployment manager jinja 컨피그레이션 파일에 넣을 ASA 가상 컨피그레이션 파일을 준비합니다. 이 컨피그레이션은 ASA 가상 프로젝트의 인스턴스 템플릿에서 시작 스크립트로 사용합니다.

컨피그레이션 파일에는 최소한 다음이 포함되어야 합니다.

- 모든 인터페이스에 DHCP IP 할당을 설정합니다.

- GCP 로드 밸런서는 nic0으로만 트래픽을 전달하므로 Nic0은 'outside(외부)'로 표시되어야 합니다.
- Nic0은 IP 전달만 지원하므로 ASA 가상에 대한 SSH에 사용됩니다.
- ASA 컨피그레이션의 외부 인터페이스에서 SSH를 활성화합니다.
- 외부에서 내부 인터페이스로 트래픽을 전달하도록 NAT 컨피그레이션을 생성합니다.
- 원하는 트래픽을 허용하는 액세스 정책을 생성합니다.
- 리소스 상태의 경우, 적절한 NAT 규칙을 사용하여 관련 상태 프로브를 메타데이터 서버로 리디렉션해야 합니다.

다음은 참조용으로만 사용할 수 있는 샘플 ASA 컨피그레이션 파일입니다.

```
!ASA Version 9.15.1.10
!Interface Config
interface G0/0
nameif inside
security-level 100
ip address dhcp setroute
no shutdown


interface G0/1
nameif management
security-level 50
ip address dhcp setroute
no shutdown

interface M0/0
no management-only
nameif outside
security-level 0
ip address dhcp setroute
no shutdown
!
same-security-traffic permit inter-interface
!
!Due to some constraints in GCP,
!"GigabitEthernet0/0" will be used as a Management interface
!"Management0/0" will be used as a data interface
crypto key generate rsa modulus 2048
ssh 0.0.0.0 0.0.0.0 management
ssh version 2
ssh timeout 60
aaa authentication ssh console LOCAL
ssh authentication publickey {{ properties["publicKey"] }}
username admin privilege 15
username admin attributes
service-type admin

! required config end
dns domain-lookup management
dns server-group DefaultDNS
name-server 8.8.8.8
!
access-list all extended permit ip any any
access-list out standard permit any4
access-group all global
```

```
! Objects
object network metadata
host 169.254.169.254
object network ilb
host $(ref.{{ properties["resourceNamePrefix"] }}-ilb-ip.address)
object network hc1
subnet 35.191.0.0 255.255.0.0
object network hc2
subnet 130.211.0.0 255.255.63.0
object network elb
host $(ref.{{ properties["resourceNamePrefix"] }}-elb-ip.address)
object network appServer
host 10.61.2.3
object network defaultGateway
subnet 0.0.0.0 0.0.0.0
! Nat Rules
nat (inside,outside) source dynamic hc1 ilb destination static ilb metadata
nat (inside,outside) source dynamic hc2 ilb destination static ilb metadata
nat (inside,outside) source dynamic defaultGateway interface
 !
 object network appServer
nat (inside,outside) static $(ref.{{ properties["resourceNamePrefix"] }}-elb-ip.address)
object network defaultGateway
nat (outside,inside) dynamic interface
! Route Add
route inside 0.0.0.0 0.0.0.0 10.61.1.1 2
route management 0.0.0.0 0.0.0.0 10.61.3.1 3
license smart register idtoken <licenseIDToken>
```

GCP 클라우드 기능 패키지 구축

ASA 가상 GCP Auto Scale 솔루션에서는 클라우드 기능을 압축된 ZIP 패키지 형식으로 전달하는 아카이브 파일 2개를 빌드해야 합니다.

- `scalein-action.zip`
- `scaleout-action.zip`

`scalein-action.zip` 및 `scaleout-action.zip` 패키지를 구축하는 자세한 방법은 Auto Scale 구축 지침을 참조하십시오.

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

입력 매개변수

다음 표에서는 템플릿 매개 변수를 정의하고 일 예를 제공합니다. 이러한 값을 결정하고 나면 GCP 프로젝트에 GCP 템플릿을 구축할 때 이러한 매개변수를 사용하여 ASA 가상 디바이스를 생성할 수 있습니다.

표 *1:* 템플릿 매개변수

매개 변수 이름	허용되는 값 / 유형	설명	리소스 생성 유형
resourceNamePrefix	문자열	모든 리소스는 이 접두사를 포함하는 이름으로 생성됩니다. 예: demo-test	신규
region	GCP에서 지원하는 유효한 지역 [문자열]	프로젝트가 구축될 지역의 이름입니다. 예: us-central1	
serviceAccountMailId	문자열 [이메일 ID]	서비스 계정을 식별하는 이메일 주소입니다.	
vpcConnectorName	문자열	서버리스 환경과 VPC 네트워크 간의 트래픽을 처리하는 커넥터의 이름입니다. 예: demo-test-vpc-connector	
bucketName	문자열	클라우드 함수 ZIP 패키지가 업로드될 GCP 스토리지 버킷의 이름입니다. 예: demo-test-bkt	
cpuUtilizationTarget	10진수 (0,1]	Autoscaler가 유지해야 하는 인스턴스 그룹에 있는 VM의 평균 CPU 사용률입니다. 예: 0.5	
healthCheckFirewallRuleName	문자열	상태 검사 프로브 IP 범위의 패킷을 허용하는 방화벽 규칙의 태그입니다. 예: demo-test-healthallowall	기존

매개 변수 이름	허용되는 값 / 유형	설명	리소스 생성 유형
insideFirewallRuleName	문자열	내부 VPC에서 통신을 허용하는 방화벽 규칙의 태그입니다. 예: demo-test-inside-allowall	기존
insideVPCName	문자열	내부 VPC의 이름입니다. 예: demo-test-inside	기존
insideVPCSubnet	문자열	내부 서브넷의 이름입니다. 예: demo-test-inside-subnt	기존
머신 유형	문자열	ASA 가상 VM의 머신 유형입니다. 예: e2-standard-4	
maxASACount	정수	인스턴스 그룹에서 허용되는 최대 ASA 가상 인스턴스 수입니다. 예: 3	
mgmtFirewallRuleName	문자열	관리 VPC에서의 통신을 허용하는 방화벽 규칙의 태그입니다. 예: demo-test-mgmt-allowall	
mgmtVPCName	문자열	관리 VPC의 이름입니다. 예: demo-test-mgmt	
mgmtVPCSubnet	문자열	관리 서브넷의 이름입니다. 예: demo-test-mgmt-subnt	

매개 변수 이름	허용되는 값 / 유형	설명	리소스 생성 유형
minASACount	정수	지정된 시간에 인스턴스 그룹에서 사용 가능한 최소 ASA 가상 인스턴스 수입니다. 예: 1	
outsideFirewallRuleName	문자열	외부 VPC에서의 통신을 허용하는 방화벽 규칙의 태그입니다. 예: demo-test-outside-allowall	
outsideVPCName	문자열	외부 VPC의 이름입니다. 예: demo-test-outside	
outsideVPCSubnet	문자열	외부 서브넷의 이름입니다. 예: demo-test-outside-subnt	
publicKey	문자열	ASA 가상 VM의 SSH 키입니다.	
sourceImageURL	문자열	프로젝트에서 사용할 ASA 가상의 이미지입니다. 예: https://www.googleapis.com/compute/v1/projects/cisco-public/global/images/cisco-asav-9-15-1-15	
애플리케이션 서버 IP 주소	문자열	내부 Linux 머신의 내부 IP 주소입니다. 예: 10.61.1.2	
내부 VPC 게이트웨이 IP 주소	문자열	내부 VPC의 게이트웨이입니다. 예: 10.61.1.1	

매개 변수 이름	허용되는 값 / 유형	설명	리소스 생성 유형
관리 VPC 게이트웨이 IP 주소	문자열	관리 VPC의 게이트웨이입니다. 예: 10.61.3.1	

Auto Scale 솔루션 구축

단계 **1** 로컬 폴더에 Git 리포지토리를 복제합니다.

```
git clone git_url -b branch_name
```

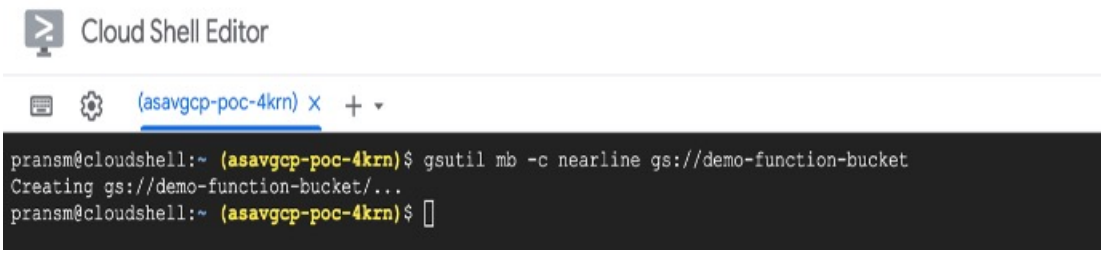
예제:

```
Last login: Thu Jun  3 13:01:32 on ttys002
(base) pransm@PRANSM-M-F9KA ~ % git clone https://bitbucket-eng-bgl1.cisco.com/bitbucket/scm/vcb/cloud_autoscale.git -b saaarwar_asa_autoscale_public_key
Cloning into 'cloud_autoscale'...
remote: Enumerating objects: 1604, done.
remote: Counting objects: 100% (1604/1604), done.
remote: Compressing objects: 100% (1507/1507), done.
remote: Total 1604 (delta 759), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (1604/1604), 58.35 MiB | 8.54 MiB/s, done.
Resolving deltas: 100% (759/759), done.
(base) pransm@PRANSM-M-F9KA ~ %
```

단계 **2** gcloud CLI에서 버킷을 생성합니다.

```
gsutil mb -c nearline gs://bucket_name
```

예제:



단계 **3** 압축된 Zip 패키지를 빌드합니다.

a) `scalein_action` 및 `scaleout_action` 폴더에서 다음 파일로 구성된 압축된 Zip 패키지를 생성합니다.

- main.py
- basic_functions.py

  requirements.txt

b) 압축된 Zip 패키지의 이름을 `scaleout-action.zip` 및 `scalein-action.zip`으로 변경합니다.

참고 폴더 내부로 이동하여 파일을 선택하고 마우스 오른쪽 버튼으로 클릭한 다음 'compress | archive'를 선택하여, GCP가 읽을 수 있는 .zip 파일을 만듭니다.

단계 **4** 압축된 Zip 패키지(`scaleout-action.zip` 및 `scalein-action.zip`)를 Cloud Editor 작업 공간에 업로드합니다.

단계 **5** 구축 관리자 템플릿의 파일을 Cloud Editor 작업 공간에 업로드합니다.

- asav_autoscale.jinja
- asav_autoscale_params.yaml
- pre_deployment.jinja
- pre_deployment.yaml

단계 **6** 압축된 Zip 패키지를 버킷 스토리지에 복사합니다.

- `gsutil cp scaleout-action.zip gs://bucket_name`
- `gsutil cp scalein-action.zip gs://bucket_name`

예제:

```
pransm@cloudshell:~ (asavgcp-poc-4krn)$ gsutil cp scaleout-action.zip gs://demo-function-bucket
Copying file://scaleout-action.zip [Content-Type=application/zip]...
/ [1 files][  3.3 KiB/  3.3 KiB]
Operation completed over 1 objects/3.3 KiB.
pransm@cloudshell:~ (asavgcp-poc-4krn)$ gsutil cp scalein-action.zip gs://demo-function-bucket
Copying file://scalein-action.zip [Content-Type=application/zip]...
/ [1 files][  3.3 KiB/  3.3 KiB]
Operation completed over 1 objects/3.3 KiB.
pransm@cloudshell:~ (asavgcp-poc-4krn)$
```

단계 **7** 내부, 외부 및 관리 인터페이스용 VPC와 서브넷을 생성합니다.

관리 VPC에는 /28 서브넷이 있어야 합니다(예: 10.8.2.0/28).

단계 **8** 내부, 외부 및 관리 인터페이스를 위한 방화벽 규칙 3개가 필요합니다. 상태 검사 프로브를 허용하는 방화벽 규칙도 있어야 합니다.

단계 **9** 사전 구축 및 ASA 가상 자동 확장 구축을 위한 Jinja 및 YAML 파일에서 매개변수를 업데이트합니다.

a) `asav_autoscale_params.yaml` 파일을 열고 다음 매개변수를 업데이트합니다.

- **resourceNamePrefix**: <resourceNamePrefix>
- **region**: <region>
- **serviceAccountMailId**: <serviceAccountMailId>
- **publicKey**: <publicKey>
- **insideVPCName**: <Inside-VPC-Name>
- **insideVPCSubnet**: <Inside-VPC-Subnet>
- **outsideVPCName**: <Outside-VPC-Name>
- **outsideVPCSubnet**: <Outside-VPC-Subnet>
- **mgmtVPCName**: <Mgmt-VPC-Name>

- **mgmtVPCSubnet**: <Mgmt-VPC-Subnet>
- **insideFirewallRuleName**: <Inside-Network-Firewall-Tag>
- **outsideFirewallRuleName**: <Outside-Network-Firewall-Tag>
- **mgmtFirewallRuleName**: <Mgmt-Network-Firewall-Tag>
- **healthCheckFirewallRuleName**: <HealthCheck-IP-Firewall-Tag>
- **machineType**: <machineType>

참고 ASA 가상 Auto Scale의 경우 **cpuUtilizationTarget: 0.5** 매개변수가 설정되며, 요구 사항에 맞게 수정할 수 있습니다.

이 값은 모든 ASA 가상 인스턴스 그룹의 CPU 사용량이 50%임을 나타냅니다.

b) `asav_autoscale.jinja` 파일을 열고 다음 매개변수를 업데이트합니다.

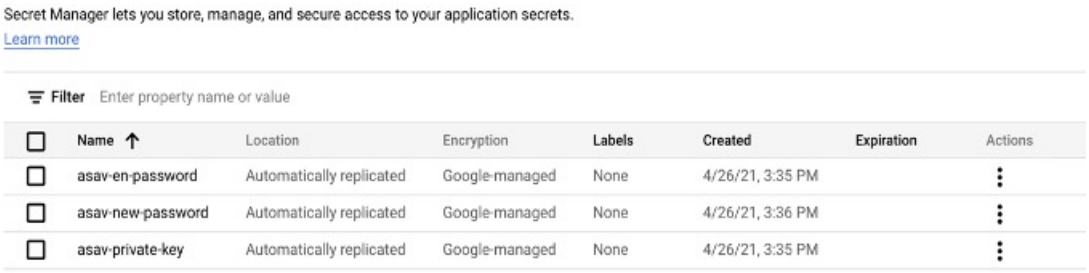
- **host**: <Application server IP address>
- **route inside 0.0.0.0 0.0.0.0**: <Inside VPC Gateway IP address> 2
- **route management 0.0.0.0 0.0.0.0**: <Management VPC Gateway IP address> 3
- **license smart register idtoken**: <licenseIDToken>

c) `pre_deployment.yaml` 파일을 열고 다음 매개변수를 업데이트합니다.

- **resourceNamePrefix**: <resourceNamePrefix>
- **region**: <region>
- **serviceAccountMailId**: <serviceAccountMailId>
- **vpcConnectorName**: <VPC-Connector-Name>
- **bucketName**: <bucketName>

단계 **10** Secret Manager GUI를 사용하여 다음을 위한 3가지 암호를 생성합니다. https://console.cloud.google.com/security/secret-manager의 내용을 참조하십시오.

- asav-en-password
- asav-new-password
- asav-private-key


Secret Manager lets you store, manage, and secure access to your application secrets.
Learn more

Filter Enter property name or value

Name	Location	Encryption	Labels	Created	Expiration	Actions
asav-en-password	Automatically replicated	Google-managed	None	4/26/21, 3:35 PM		
asav-new-password	Automatically replicated	Google-managed	None	4/26/21, 3:36 PM		
asav-private-key	Automatically replicated	Google-managed	None	4/26/21, 3:35 PM		

단계 **11** VPC 커넥터를 만듭니다.

```
gcloud beta compute networks vpc-access connectors create <vpc-connector-name>
--region <region> --subnet=</28 subnet name>
```

예제:

```
gcloud beta compute networks vpc-access connectors create demo-vpc-connector
--region us-central1 --subnet=outside-connect-28
Create request issued for: [demo-vpc-connector]
Waiting for operation [projects/asavgcp-poc-4krn/locations/us-central1/operations/
10595de7-837f-4c19-9396-0c22943ecf15] to complete...done.
Created connector [demo-vpc-connector].
```

단계 **12** 사전 구축 YAML 컨피그레이션을 구축합니다.

```
gcloud deployment-manager deployments create <pre-deployment-name>
--config pre_deployment.yaml
```

예제:

```
gcloud deployment-manager deployments create demo-predeployment
--config pre_deployment.yaml
The fingerprint of the deployment is b'9NOy0gsTPgg16SqUEVsBjA=='
Waiting for create [operation-1624383045917-5c55e266e596d-4979c5b6-66d1025c]...done.
Create operation operation-1624383045917-5c55e266e596d-4979c5b6-66d1025c
completed successfully
```

NAME	TYPE	STATE
demo-asav-delete-sink	gcp-types/logging-v2:projects.sinks	COMPLETED
demo-asav-insert-sink	gcp-types/logging-v2:projects.sinks	COMPLETED
demo-asav-pubsub-topic-delete	pubsub.v1.topic	COMPLETED
demo-asav-pubsub-topic-insert	pubsub.v1.topic	COMPLETED
demo-asav-scalein-action	gcp-types/cloudfunctions-v1:projects.locations.functions	COMPLETED
demo-asav-scaleout-action	gcp-types/cloudfunctions-v1:projects.locations.functions	COMPLETED

단계 **13** ASA 가상 Auto Scale 구축을 만듭니다.

```
gcloud deployment-manager deployments create <deployment-name>
--config asav_autoscale_params.yaml
```

예제:

```
gcloud deployment-manager deployments create demo-asav-autoscale
--config asav_autoscale_params.yaml
The fingerprint of the deployment is b'1JCQi7Il-laWOY7vOLza0g=='
Waiting for create [operation-1624383774235-5c55e51d79d01-1a3acf92-4f3daf16]...done.
Create operation operation-1624383774235-5c55e51d79d01-1a3acf92-4f3daf16
completed successfully.
```

NAME	TYPE	STATE
demo-asav-autoscaler	compute.v1.regionAutoscaler	COMPLETED
demo-asav-backend-service-elb	compute.v1.regionBackendService	COMPLETED
demo-asav-backend-service-ilb	compute.v1.regionBackendService	COMPLETED
demo-asav-fr-elb	compute.v1.forwardingRule	COMPLETED
demo-asav-fr-ilb	compute.v1.forwardingRule	COMPLETED
demo-asav-hc-elb	compute.v1.regionHealthChecks	COMPLETED
demo-asav-hc-ilb	compute.v1.healthCheck	COMPLETED
demo-asav-health-check	compute.v1.healthCheck	COMPLETED
demo-asav-instance-group	compute.v1.regionInstanceGroupManager	COMPLETED
demo-asav-instance-template	compute.v1.instanceTemplate	COMPLETED
demo-elb-ip	compute.v1.address	COMPLETED

단계 **14** ILB가 내부 애플리케이션에서 인터넷으로 패킷을 전달할 경로를 만듭니다.

```
gcloud beta compute routes create <ilb-route-name>
--network=<inside-vpc-name> --priority=1000 --destination-range=0.0.0.0/0
--next-hop-ilb=<ilb-forwarding-rule-name> --next-hop-ilb-region=<region>
```

예제:

```
gcloud beta compute routes create demo-ilb --network=sdt-test-asav-inside
--priority=1000 --destination-range=0.0.0.0/0 --next-hop-ilb=demo-asav-fr-ilb
--next-hop-ilb-region=us-central1
Created [https://www.googleapis.com/compute/beta/projects/asavgcp-poc-4krn/global
/routes/demo-ilb].
```

NAME	NETWORK	DEST_RANGE	NEXT_HOP	PRIORITY
demo-ilb	sdt-test-asav-inside	0.0.0.0/0	10.7.1.60	1000

단계 **15** Cloud Router 및 Cloud NAT를 만듭니다.

```
gcloud compute routers create <cloud-router-name>
--project=<project-name> --region <region> --network=<outside-vpc-name>
--advertisement-mode=custom
```

```
gcloud compute routers nats create <cloud-nat-name>
--router=<cloud-router-name> --nat-all-subnet-ip-ranges --auto-allocate-nat-external-ips
--region=<region>
```

예제:

```
gcloud compute routers create demo-cloud-router --project=asavgcp-poc-4krn
--region us-central1 --network=sdt-test-asav-outside --advertisement-mode=custom
Creating router [demo-cloud-router]...done.
```

NAME	REGION	NETWORK
demo-cloud-router	us-central1	sdt-test-asav-outside

```
gcloud compute routers nats create demo-cloud-nat
--router=demo-cloud-router --nat-all-subnet-ip-ranges
--auto-allocate nat-external-ips --region=us-central1
Creating NAT [demo-cloud-nat] in router [demo-cloud-router]...done.
```

Auto Scale 논리

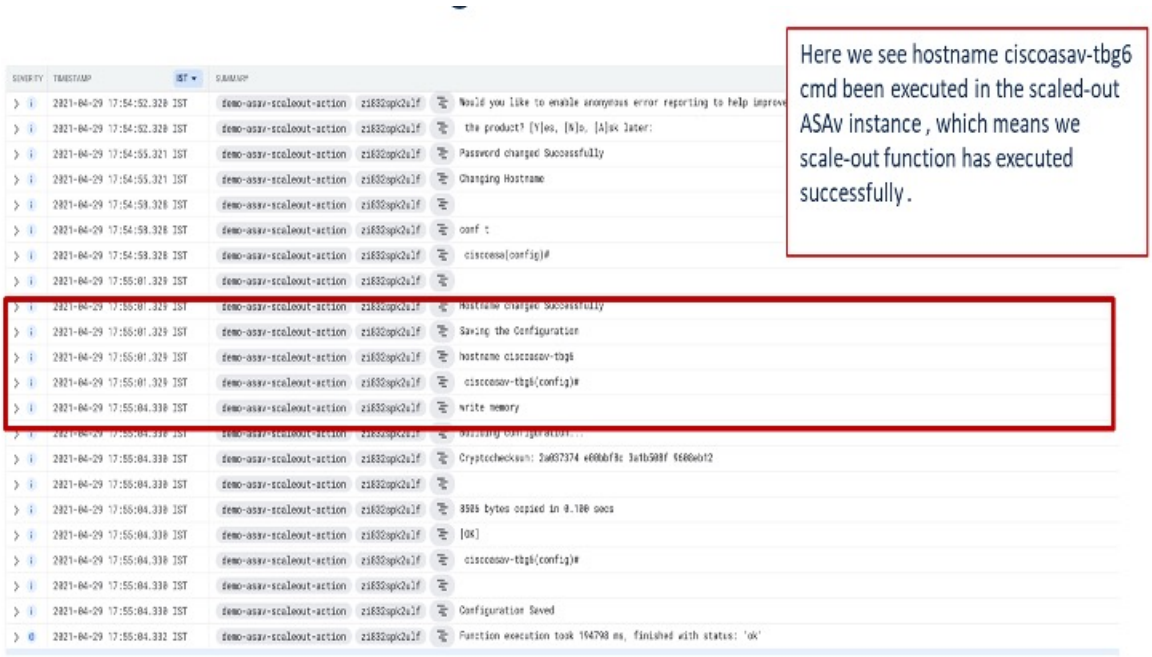
- 자동 확장기는 목표 CPU 사용률을 인스턴스 그룹에서 시간 경과에 따른 모든 vCPU의 평균 사용량의 일부로 취급합니다.
- 총 vCPU의 평균 사용률이 목표 사용률을 초과하면, 자동 확장기는 VM 인스턴스를 추가합니다. 총 vCPU의 평균 사용률이 목표 사용률 미만이라면, 자동 확장기는 인스턴스를 제거합니다.
- 예를 들어 목표 사용률을 0.75로 설정하면 자동 확장기는 인스턴스 그룹 내 모든 vCPU의 평균 사용률을 75%로 유지합니다.
- 확장 결정에는 CPU 사용률 메트릭만 사용합니다.
- 위의 논리는 로드 밸런서가 모든 ASAv에 연결을 동일하게 분산하려 한다는 가정을 기반으로 하며, 평균적으로 모든 ASAv를 동일하게 로드해야 합니다.

Auto Scale 로깅 및 디버깅

클라우드 기능의 로그는 다음과 같이 볼 수 있습니다.

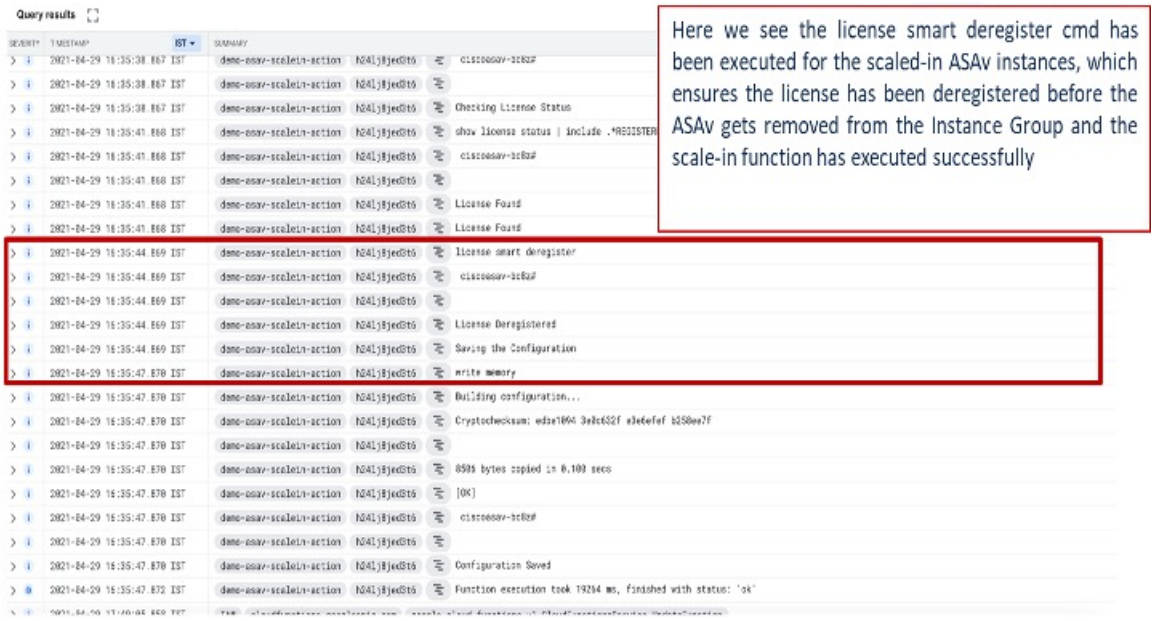
- 확장 기능 로그

  그림 ***7:*** 확장 기능 로그



- 축소 기능 로그

그림 *8:* 축소 기능 로그



Auto Scale 지침 및 제한 사항

- IPv4만 지원됩니다.
- 지원되는 라이선싱은 BYOL 전용입니다. PAYG는 GCP의 ASA 가상에서는 사용할 수 없습니다.
- 외부 로드 밸런서는 템플릿에 의해 생성되므로, 로드 밸런서의 공용 IP에 대한 특정 DNS 요구 사항은 범위를 벗어납니다.
- 애플리케이션이 사용자가 생성한 로드 밸런서 뒤에 있으며, ASA 가상은 모든 트래픽을 (트래픽을 특정 애플리케이션 IP로 직접 전송하는 대신) 이 로드 밸런서로 라우팅한다고 가정합니다.
- 태그, 리던던시 및 로드 밸런서 선호도 구성의 필요성에 대한 세부 정보는 고려되지 않습니다.
- ASA 가상 자격 증명은 다음과 같이 표시됩니다.
  - 서버리스 코드의 텍스트를 지웁니다.
  - 인스턴스 그룹의 모든 인스턴스에 표시됩니다.
  - 공유 GCP 계정을 사용하는 경우 인스턴스 템플릿에 표시됩니다.

민감한 데이터는 GCP의 공개 키 서비스를 사용하여 보호할 수 있습니다.

중요 Cisco에서는 라이선싱 서버를 이용한 ASA 가상 등록을 주기적으로 추적하여 확장 ASA가 라이선싱 서버에 정상적으로 등록되고 있는지, 그리고 축소 ASA 가상 인스턴스가 라이선스 서버에서 제거되는지 확인하는 것을 권장합니다.

Auto Scale 문제 해결

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

- `main.py`를 찾을 수 없음 - Zip 패키지가 파일을 통해서만 생성되었는지 확인합니다. 클라우드 기능으로 이동하여 파일 트리를 확인할 수 있습니다. 어떤 폴더도 없어야 합니다.
- 템플릿을 구축하는 동안 오류 발생 - "<>" 안에 jimja 및 .yaml을 포함한 모든 매개변수 값이 입력되어 있으며, 동일한 이름의 구축이 이미 존재하는지 확인하십시오.
- Google Function에서 ASA 가상에 연결할 수 없음 - VPC 커넥터가 생성되었으며 YAML 매개변수 파일에 동일한 이름이 언급되는지 확인합니다.
- ASA 가상 SSH 과정에서 인증 실패 - 공개 및 개인 키 쌍이 올바른지 확인합니다.
- 라이선스 등록 실패 - 라이선스 ID 토큰이 올바른지 확인합니다. 또한 클라우드 NAT가 생성되었으며 ASA 가상에서 tools.cisco.com에 연결할 수 있는지 확인합니다.