Déployer la solution d'évolutivité automatique de l'ASA virtuel sur GCP

- Aperçu, à la page 1
- Télécharger le paquet de déploiement, à la page 3
- Composants de la solution d'évolutivité automatique, à la page 3
- Prérequis, à la page 6
- Déployer la solution d'évolutivité automatique, à la page 13
- Logique d'évolutivité automatique, à la page 17
- Journalisation et débogage, à la page 18
- Lignes directrices et limites relatives à la licence, à la page 19
- Dépannage, à la page 19

Aperçu

Les sections suivantes décrivent comment les composants de la solution d'évolutivité automatique fonctionnent pour l'ASA virtuel sur GCP.

À propos de la solution d'évolutivité automatique

L'évolutivité automatique d'ASA virtuel pour GCP est une implémentation complète sans serveur qui utilise l'infrastructure sans serveur fournie par GCP (fonctions dans le nuage, équilibreurs de charges, publication/abonnement, groupes d'instances, etc.).

Certaines des fonctionnalités clés de l'évolutivité automatique d'ASA virtuel pour l'implémentation GCP comprennent :

- Déploiement basé sur le modèle de gestionnaire de déploiement GCP.
- Prise en charge des mesures d'évolutivité en fonction du CPU.
- Prise en charge du déploiement d'ASA virtuel et des zones de multi-disponibilité.
- La configuration entièrement automatisée s'applique automatiquement aux instances ASA virtuel mises à l'échelle.
- Prise en charge des équilibreurs de charges et des zones de multi-disponibilité.

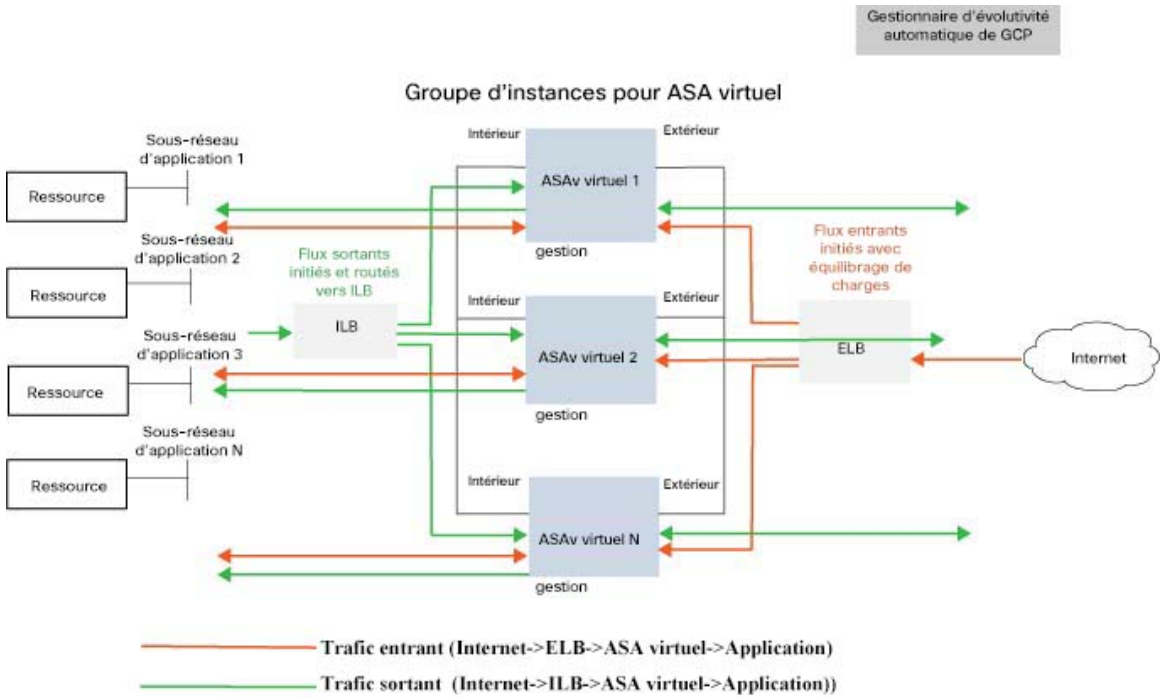
- Cisco fournit un paquet de déploiement de l'évolutivité automatique pour GCP afin de faciliter le déploiement.

Scénario d'évolutivité automatique

L'évolutivité automatique d'ASA virtuel pour GCP est une solution d'évolutivité horizontale automatisée qui positionne un groupe d'instances ASA virtuel entre un équilibreur de charge interne (ILB) et un équilibreur de charge externe (ELB) GCP.

- L'ELB distribue le trafic Internet aux instances ASA virtuel du groupe d'instances; le pare-feu transfère ensuite le trafic à l'application.
- L'ILB distribue le trafic Internet sortant d'une application aux instances ASA virtuel du groupe d'instances; le pare-feu transfère ensuite le trafic à Internet.
- Un paquet réseau ne traversera jamais les deux équilibreurs de charges (interne et externe) en une seule connexion.
- Le nombre d'instances ASA virtuel dans l'ensemble d'évolutivité se verra évoluer et sera configuré automatiquement en fonction des conditions de charge.

Illustration 1 : Scénario d'évolutivité automatique d'ASA virtuel



Champ d'application

Ce document couvre les procédures détaillées pour déployer les composants sans serveur pour la solution ASA virtuel d'évolutivité automatique pour GCP.

Important

- Lisez le document entier avant de commencer le déploiement.
- Assurez-vous que les conditions préalables sont remplies avant de commencer le déploiement.
- Assurez-vous de suivre les étapes et l'ordre d'exécution décrits dans le présent document.

Télécharger le paquet de déploiement

La solution d'évolutivité automatique d'ASA virtuel pour GCP est un déploiement basé sur le modèle de gestionnaire de déploiement GCP qui utilise l'infrastructure sans serveur fournie par GCP (fonctions dans le nuage, équilibreurs de charges, publication/abonnement, groupes d'instances, etc.).

Téléchargez les fichiers requis pour lancer la solution d'évolutivité automatique d'ASA virtuel pour GCP. Les scripts et les modèles de déploiement pour votre version ASA sont disponibles dans le référentiel GitHub.

Attention Remarque : les scripts et les modèles de déploiement fournis par Cisco pour l'évolutivité automatique sont présentés à titre d'exemples de code source libre et ne font pas l'objet de l'assistance technique du TAC dans sa portée normale.

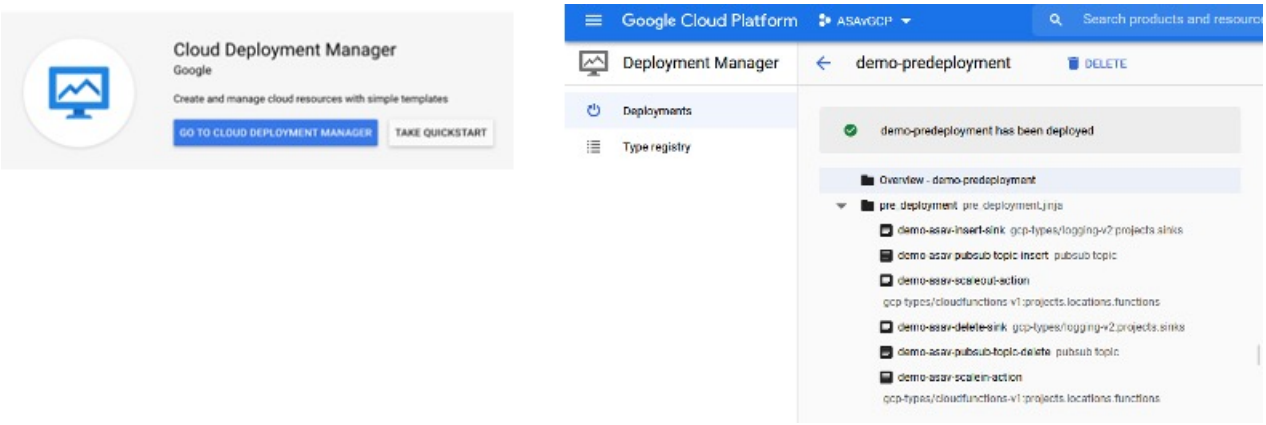
Composants de la solution d'évolutivité automatique

Les composants suivants constituent la solution d'évolutivité automatique ASA virtuel pour GCP.

Gestionnaire de déploiement

- Traitez votre configuration comme du code et effectuez des déploiements reproductibles. Google Cloud Deployment Manager vous permet de spécifier toutes les ressources nécessaires à votre application dans un format explicite à l'aide de YAML. Vous pouvez également utiliser des modèles Python ou Jinja2 pour paramétrer la configuration et permettre la réutilisation des schémas de déploiement courants.
- Créez des fichiers de configuration qui définissent les ressources. Le processus de création de ces ressources peut être répété à plusieurs reprises avec des résultats cohérents. Consultez https://cloud.google.com/deployment-manager/docs pour obtenir de plus amples renseignements.

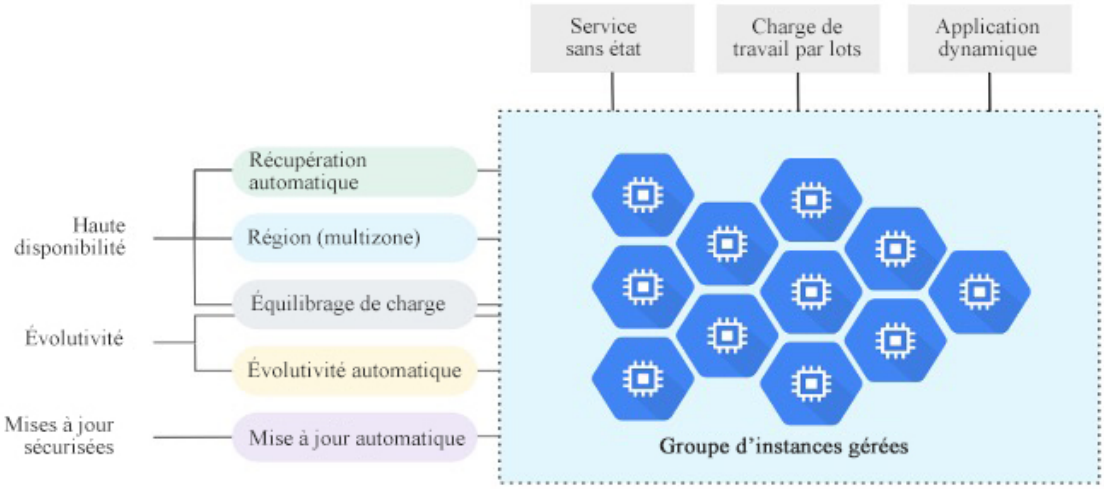
Illustration 2 : Vue du gestionnaire de déploiement



Groupe d'instances gérées dans GCP

Un groupe d'instances gérées (MIG) crée chacune de ses instances gérées en fonction du modèle d'instance et de la configuration dynamique facultative que vous spécifiez. Consultez https://cloud.google.com/compute/docs/instance-groups pour obtenir de plus amples renseignements.

Illustration 3 : Fonctionnalités du groupe d'instances

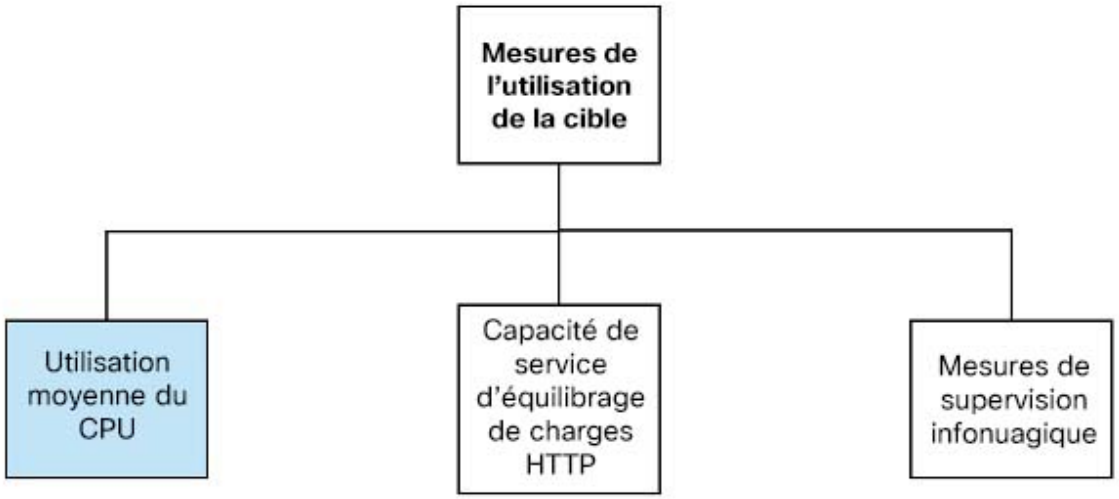


Mesures de l'utilisation de la cible

- Le diagramme suivant présente les mesures d'utilisation de la cible. Seules les mesures d'utilisation moyenne du CPU sont utilisées pour prendre des décisions en matière d'évolutivité automatique.
- Le dispositif d'évolutivité automatique recueille en permanence des renseignements sur l'utilisation en fonction de la mesure d'utilisation sélectionnée, compare l'utilisation réelle à votre utilisation cible souhaitée et utilise ces renseignements pour déterminer si le groupe doit supprimer des instances (évolutivité à la baisse) ou ajouter des instances (évolutivité à la hausse).

- Le niveau d'utilisation cible est le niveau auquel vous souhaitez maintenir vos instances de machine virtuelle (VM). Par exemple, si vous évoluez en fonction de l'utilisation du CPU, vous pouvez définir votre niveau d'utilisation cible à 75 % et le dispositif d'évolutivité automatique maintiendra l'utilisation du CPU du groupe d'instances spécifié à 75 %. Le niveau d'utilisation de chaque mesure est interprété différemment en fonction de la politique d'évolutivité automatique. Consultez https://cloud.google.com/compute/docs/autoscaler pour obtenir de plus amples renseignements.

Illustration 4 : Mesures de l'utilisation de la cible



Fonctions de nuage sans serveur

Vous utilisez les fonctions Google Cloud sans serveur pour définir le mot de passe SSH, activer le mot de passe et modifier le nom de domaine lorsque l'instance se présente dans le gestionnaire de groupe d'instances.

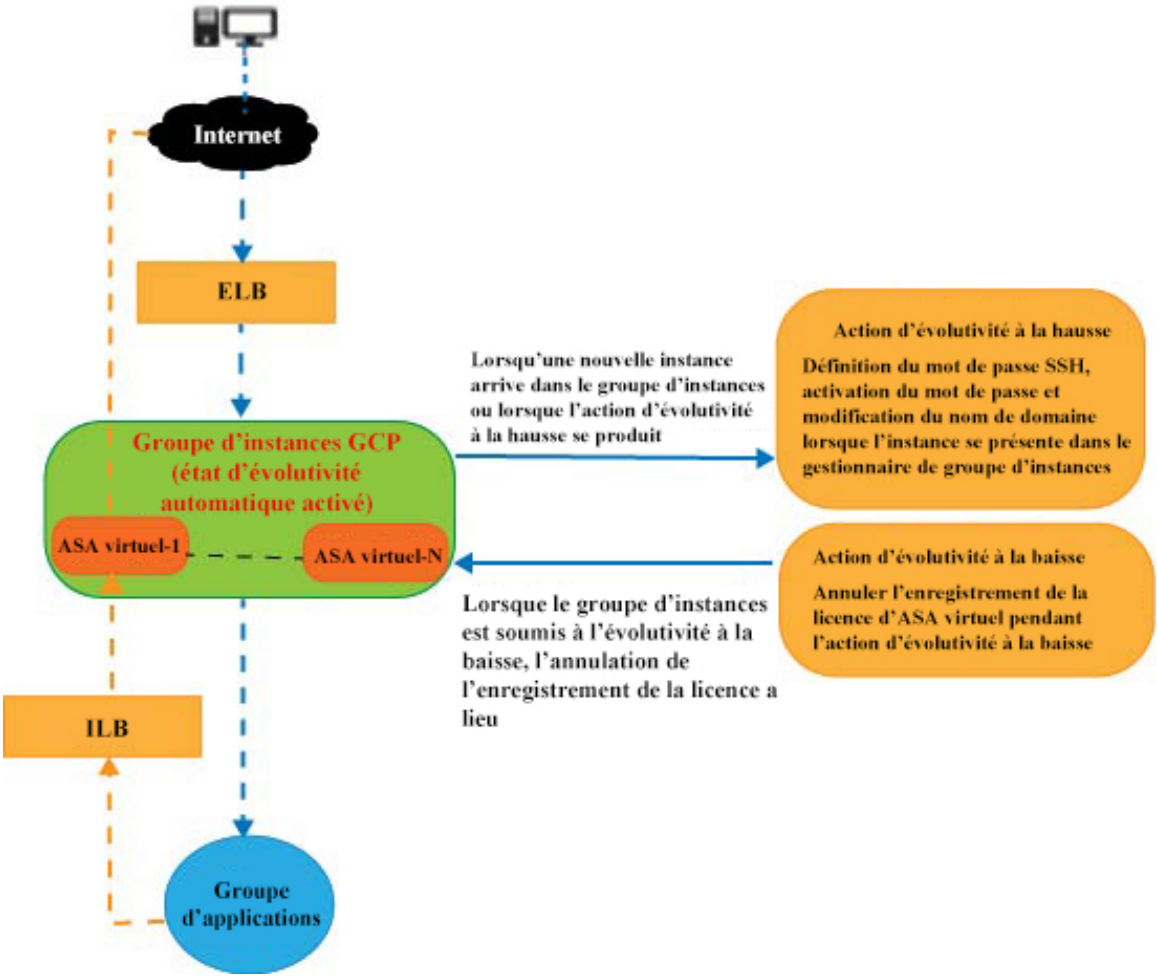
- Lorsqu'une nouvelle instance ASA virtuel apparaît dans le groupe d'instances lors du processus d'évolutivité à la hausse, vous devez définir le mot de passe SSH, activer le mot de passe et changer le nom de domaine, car vous ne pouvez pas toujours surveiller le processus d'évolutivité à la hausse.
- Les fonctions du nuage sont déclenchées par un sujet de publication ou d'abonnement en nuage pendant le processus d'évolutivité à la hausse. Vous disposez également d'un récepteur de journaux avec un filtre exclusif à l'ajout d'instances lors de l'évolutivité à la hausse.

Annulation de l'enregistrement de la licence sans serveur à l'aide des fonctions du nuage

- Pendant que les instances sont supprimées lors de l'évolutivité à la baisse, vous devez annuler l'enregistrement de la licence de l'instance ASA virtuel.
- Les fonctions du nuage sont déclenchées par un sujet de publication ou d'abonnement en nuage. En particulier pour le processus de suppression, vous disposez d'un récepteur de journaux avec un filtre exclusif à la suppression des instances lors de l'évolutivité à la baisse.
- La fonction du nuage, lorsqu'elle est déclenchée, se connecte en SSH à l'instance ASA virtuel à supprimer et exécute la commande pour annuler l'enregistrement de la licence.

Présentation générale de la solution d'évolutivité automatique

Illustration 5 : Présentation de la solution d'évolutivité automatique



Prérequis

Ressources GCP

Projet GCP

Un projet existant ou nouvellement créé est requis pour déployer tous les composants de cette solution.

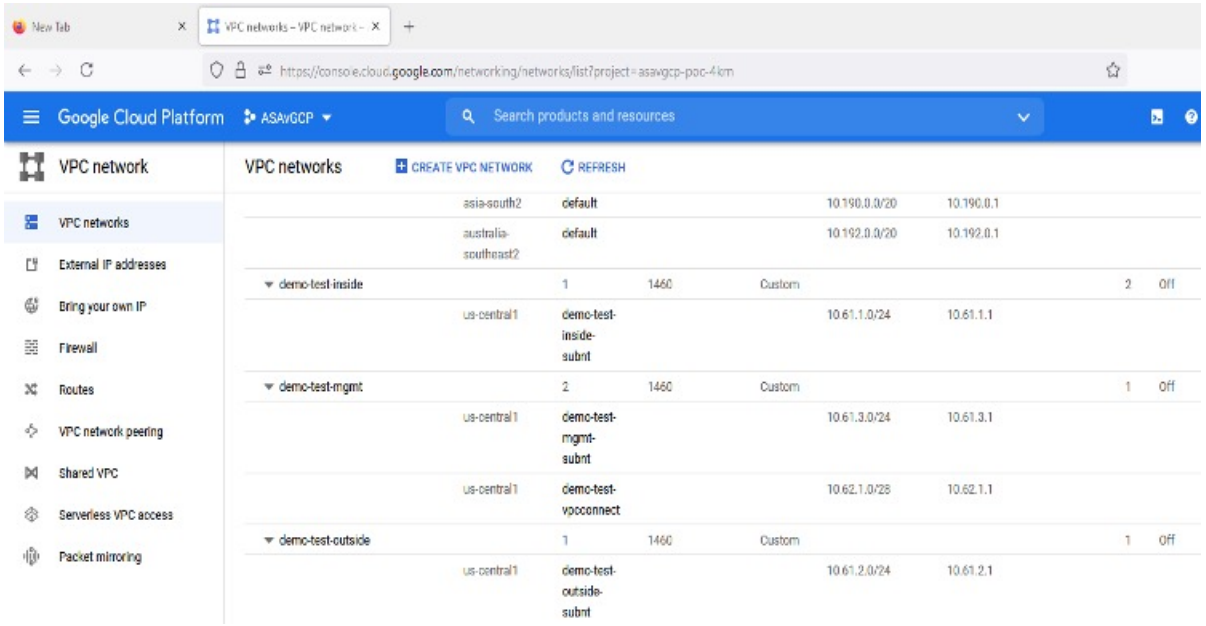
Mise en réseau

Assurez-vous que trois VPC sont disponibles ou créés. Un déploiement de l'évolutivité automatique ne créera, ne modifiera ni ne gérera de ressources réseau.

L'ASA virtuel nécessite trois interfaces réseau, donc votre réseau virtuel nécessite trois sous-réseaux pour :

- Le trafic de gestion
- Le trafic interne
- Le trafic externe

Illustration 6 : Vue du réseau VPC



Pare-feu

Des règles de pare-feu qui permettent la communication inter-VPC et permettent également la création de sondes d'intégrité. Vous devez noter les balises de pare-feu utilisées plus tard dans le modèle de gestionnaire de déploiement.

Les ports suivants doivent être ouverts dans le groupe de sécurité réseau auquel les sous-réseaux sont connectés :

- SSH (TCP/22) : requis pour la sonde d'intégrité entre l'équilibreur de charges et l'ASA virtuel. Requis pour la communication entre les fonctions sans serveur et l'ASA virtuel.
- Protocole/ports spécifiques à l'application : requis pour toutes les applications des utilisateurs (par exemple, TCP/80, etc.).

Préparer le fichier de configuration ASA

Préparez un fichier de configuration ASA virtuel qui sera ajouté dans le fichier de configuration jinja du gestionnaire de déploiement. Cette configuration sera utilisée comme script de démarrage dans le modèle d'instance pour ASA virtuel dans le projet.

Le fichier de configuration doit comporter au moins les éléments suivants :

- Définissez l'attribution d'adresse IP DHCP pour toutes les interfaces.
- Nic0 doit être marqué comme « outside » (externe), car l'équilibreur de charges GCP transfère le trafic uniquement à nic0.

- Nic0 sera utilisé pour SSH vers ASA virtuel, car il prend uniquement en charge le transfert IP.
- Activez SSH sur l'interface externe dans la configuration ASA.
- Créez une configuration NAT pour transférer le trafic de l'interface externe vers l'interface interne.
- Créez un protocole d'accès pour autoriser le trafic souhaité.
- Pour l'état d'intégrité des ressources, leurs sondes d'intégrité doivent être redirigées vers le serveur de métadonnées à l'aide des règles NAT appropriées.

Voici un exemple de fichier de configuration ASA pour référence uniquement.

```
!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>
```

Créer le paquet de fonction en nuage de GCP

La solution d'évolutivité automatique GCP d'ASA virtuel nécessite que vous créiez deux fichiers d'archive qui fournissent les fonctions en nuage sous la forme d'un paquet ZIP compressé.

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

Consultez les instructions de déploiement de l'évolutivité automatique pour savoir comment créer les paquets `scalein-action.zip` et `scaleout-action.zip`.

Ces fonctions sont aussi distinctes que possible pour effectuer des tâches spécifiques et peuvent être mises à niveau au besoin pour améliorer et prendre en charge les nouvelles versions.

Paramètres d'entrée

Le tableau suivant définit les paramètres du modèle et fournit un exemple. Une fois que vous avez choisi ces valeurs, vous pouvez utiliser ces paramètres pour créer le périphérique ASA virtuel lorsque vous déployez le modèle de gestionnaire de déploiement GCP dans votre projet GCP.

Tableau 1 : Paramètres du modèle

Nom du paramètre	Valeurs/Type autorisés	Description	Type de création de ressource
resourceNamePrefix	Chaîne	Toutes les ressources sont créées avec un nom contenant ce préfixe. Exemple : demo-test	New (Nouvelle)

Nom du paramètre	Valeurs/Type autorisés	Description	Type de création de ressource
region	Régions valides prises en charge par GCP [Chaîne]	Nom de la région où le projet sera déployé. Exemple : us-central1	
serviceAccountMailId	Chaîne [ID de courriel]	Adresse courriel qui identifie le compte de service.	
vpcConnectorName	Chaîne	Nom du connecteur qui gère le trafic entre votre environnement sans serveur et votre réseau VPC. Exemple : demo-test-vpc-connector	
bucketName	Chaîne	Nom du compartiment de stockage GCP dans lequel le progiciel ZIP de la fonction en nuage sera chargé. Exemple : demo-test-bkt	
cpuUtilizationTarget	Décimal (0, 1]	L'utilisation moyenne du CPU des machines virtuelles du groupe d'instances que l'évolutivité automatique doit maintenir. Exemple : 0.5	
healthCheckFirewallRuleName	Chaîne	Balise de la règle de pare-feu qui autorise les paquets des plages d'adresses IP de sonde de vérification de l'intégrité. Exemple : demo-test-healthallowall	Existant
insideFirewallRuleName	Chaîne	Balise des règles de pare-feu qui permet la communication dans le VPC interne. Exemple : demo-test-inside- allowall	Existant

Nom du paramètre	Valeurs/Type autorisés	Description	Type de création de ressource
insideVPCName	Chaîne	Nom du VPC interne. Exemple : demo-test-inside	Existant
insideVPCSubnet	Chaîne	Nom du sous-réseau interne. Exemple : demo-test-inside-subnt	Existant
machineType	Chaîne	Type de machine pour la machine virtuelle ASA virtuel. Exemple : e2-standard-4	
maxASAcount	Nombre entier	Le nombre maximal d'instances ASA virtuel autorisées dans le groupe d'instances. Exemple : 3	
mgmtFirewallRuleName	Chaîne	Balise des règles de pare-feu qui permet la communication dans le VPC de gestion. Exemple : demo-test-mgmt- allowall	
mgmtVPCName	Chaîne	Nom du VPC de gestion. Exemple : demo-test-mgmt	
mgmtVPCSubnet	Chaîne	Nom du sous-réseau de gestion. Exemple : demo-test-mgmt-subnt	
minASACount	Nombre entier	Le nombre minimal d'instances ASA virtuel disponibles dans le groupe d'instances à tout moment. Exemple : 1	

Nom du paramètre	Valeurs/Type autorisés	Description	Type de création de ressource
outsideFirewallRuleName	Chaîne	Balise des règles de pare-feu qui permet la communication dans le VPC externe. Exemple : demo-test-outside-allowall	
outsideVPCName	Chaîne	Nom du VPC externe. Exemple : demo-test-outside	
outsideVPCSubnet	Chaîne	Nom du sous-réseau externe. Exemple : demo-test-outside-subnt	
publicKey	Chaîne	Clé SSH de la machine virtuelle ASA virtuel.	
sourceImageURL	Chaîne	Image de l'ASA virtuel qui doit être utilisé dans le projet. Exemple : https://www.googleapis.com/compute/v1/projects/cisco-public/global/images/cisco-asav-9-15-1-15	
Adresse IP du serveur d'applications	Chaîne	Adresse IP interne de la machine Linux interne. Exemple : 10.61.1.2	
Adresse IP de la passerelle du VPC interne	Chaîne	Passerelle du VPC interne. Exemple : 10.61.1.1	
Adresse IP de la passerelle du VPC de gestion	Chaîne	Passerelle du VPC de gestion. Exemple : 10.61.3.1	

Déployer la solution d'évolutivité automatique

Procédure

Étape 1 Copiez le référentiel Git dans un dossier local.

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

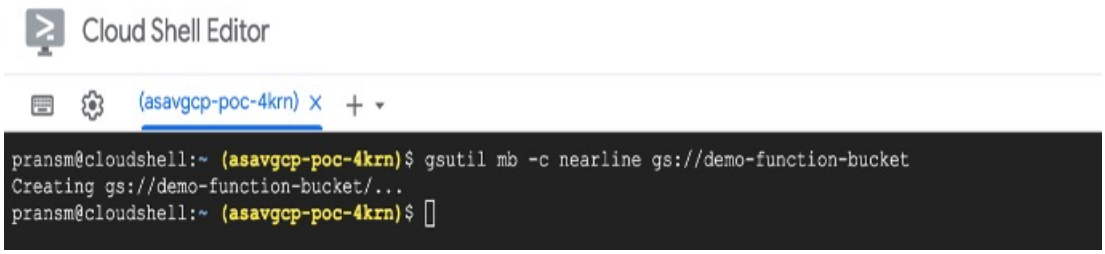
Exemple :

```
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 ~ %
```

Étape 2 Créez le compartiment dans l'interface de ligne de commande gcloud.

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

Exemple :



Étape 3 Créez des paquets zip compressés :

a) Créez des paquets zip compressés contenant les fichiers suivants à partir des dossiers `scalein_action` et `scaleout_action`.

- main.py
- basic_functions.py

requirements.txt

b) Renommez les paquets zip compressés en `scaleout-action.zip` et `scalein-action.zip`.

Remarque
Naviguez dans le dossier, sélectionnez les fichiers, cliquez avec le bouton droit et sélectionnez « compress | archive » (compresser | archiver) pour créer un fichier .zip que GCP peut lire.

Étape 4 Chargez les paquets zip compressés (`scaleout-action.zip` et `scalein-action.zip`) dans l'espace de travail de l'éditeur en nuage.

Étape 5 Chargez les fichiers suivants du modèle de gestionnaire de déploiement dans l'espace de travail de l'éditeur en nuage.

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

Étape 6 Copiez les paquets zip compressés dans le compartiment de stockage.

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

Exemple :

```
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)$
```

Étape 7 Créez un VPC et un sous-réseau pour les interfaces internes, externes et de gestion.

Dans le VPC de gestion, vous devez avoir un sous-réseau/28, par exemple, 10.8.2.0/28.

Étape 8 Vous avez besoin de trois règles de pare-feu pour les interfaces internes, externes et de gestion. En outre, vous devez avoir une règle de pare-feu pour autoriser les sondes de vérification de l'intégrité.

Étape 9 Mettez à jour les paramètres dans les fichiers Jinja et YAML pour le pré-déploiement et le déploiement de l'évolutivité automatique d'ASA virtuel.

a) Ouvrez le fichier `asav_autoscale_params.yaml` et mettez à jour les paramètres suivants :

- **préfixe du nom de la ressource** : <resourceNamePrefix>
- **région** : <region>
- **id de compte de service** : <serviceAccountMailId>
- **Clé publique** : <publicKey>
- **Nom VPC interne** : <Inside-VPC-Name>
- **Sous-réseau VPC interne** : <Inside-VPC-Subnet>
- **Nom VPC externe** :<Outside-VPC-Name>
- **Sous-réseau VPC externe** : <Outside-VPC-Subnet>
- **Nom VPC de gestion** : <Mgmt-VPC-Name>
- **Sous-réseau VPC de gestion** : <Mgmt-VPC-Subnet>
- **Nom de la règle de pare-feu interne** : <Inside-Network-Firewall-Tag>
- **Nom de la règle de pare-feu externe** : <Outside-Network-Firewall-Tag>

- **Nom de la règle de pare-feu de gestion** : <Mgmt-Network-Firewall-Tag>
- **Nom de la règle de pare-feu de vérification de l'intégrité** : <HealthCheck-IP-Firewall-Tag>
- **Type de machine** : <machineType>

Remarque
Pour l'évolutivité automatique d'ASA virtuel, le paramètre **cpuUtilizationTarget: 0.5** est défini et vous pouvez le modifier en fonction de vos besoins.

Cette valeur signifie une utilisation du processeur de 50 % de tout le groupe d'instances ASA virtuel.

b) Ouvrez le fichier `asav_autoscale.jinja` et mettez à jour les paramètres suivants.

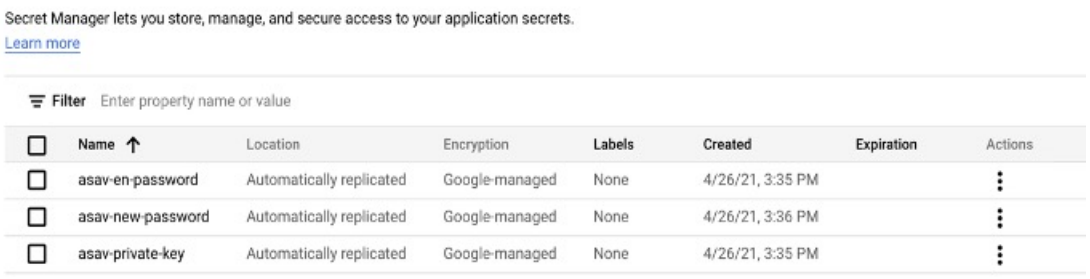
- **hôte** : <Application server IP address>
- **route à l'intérieur de 0.0.0.0 0.0.0.0** : <Inside VPC Gateway IP address> 2
- **gestion de la route 0.0.0.0 0.0.0.0** : <Management VPC Gateway IP address> 3
- **jeton d'ID d'enregistrement des licences smart** : <licenseIDToken>

c) Ouvrez le fichier `pre_deployment.yaml` et mettez à jour les paramètres suivants.

- **préfixe du nom de la ressource** : <resourceNamePrefix>
- **région** : <region>
- **id de compte de service** : <serviceAccountMailId>
- **nom du connecteur VPC** : <VPC-Connector-Name>
- **nom du compartiment** : <bucketName>

Étape 10 Créez trois secrets pour les éléments suivants à l'aide de l'interface graphique utilisateur de Secret Manager. Consultez https://console.cloud.google.com/security/secret-manager.

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



Étape 11 Créez le connecteur VPC.

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

Exemple :

```
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].
```

Étape 12 Déployez la configuration YAML de pré-déploiement.

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

Exemple :

```
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
```

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

Étape 13 Créez le déploiement d'évévolutivité automatique d'ASA virtuel.

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

Exemple :

```
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.
```

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

Étape 14 Créez une route pour qu'ILB transfère les paquets de l'application interne à Internet.

```
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>
```

Exemple :

```
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].
```

NOM	RÉSEAU	PLAGE_DEST	SAUT_SUIVANT	PRIORITÉ
demo-ilb	sdt-test-asav-inside	0.0.0.0/0	10.7.1.60	1000

Étape 15 Créez un routeur et une NAT en nuage.

```
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>
```

Exemple :

```
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.
```

NOM	RÉGION	RÉSEAU
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.
```

Logique d'évolutivité automatique

- Le dispositif d'évolutivité automatique traite le niveau d'utilisation du CPU cible comme une fraction de l'utilisation moyenne de tous les vCPU au fil du temps dans le groupe d'instances.
- Si l'utilisation moyenne de vos vCPU totaux dépasse l'utilisation cible, le dispositif d'évolutivité automatique ajoute d'autres instances de VM. Si l'utilisation moyenne de vos vCPU totaux est inférieure à l'utilisation cible, le dispositif d'évolutivité automatique supprime les instances.
- Par exemple, la définition d'une utilisation cible de 0,75 indique au dispositif d'évolutivité automatique de maintenir une utilisation moyenne de 75 % parmi tous les vCPU du groupe d'instances.
- Seules les mesures d'utilisation du CPU sont utilisées dans les décisions en matière d'évolutivité.
- Cette logique est basée sur l'hypothèse que l'équilibreur de charges tentera de répartir également les connexions sur tous les ASA et qu'en moyenne, tous les ASA doivent être chargés de la même manière.

Journalisation et débogage

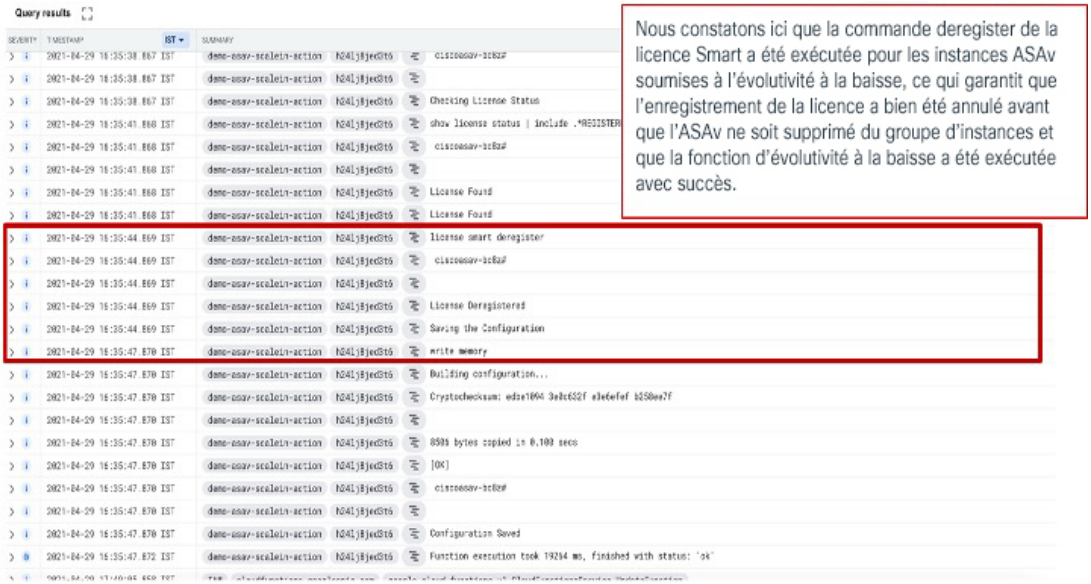
Les journaux des fonctions en nuage peuvent être consultés comme suit.

- Journaux de fonction d'évolutivité à la hausse

Illustration 7 : Journaux de fonction d'évolutivité à la hausse

- Journal de fonction d'évolutivité à la baisse

Illustration 8 : Journal de fonction d'évolutivité à la baisse



Lignes directrices et limites relatives à la licence

- Seul IPv4 est pris en charge.
- Les licences prises en charge sont BYOL uniquement. Le protocole PAYG n'est pas disponible pour l'ASA virtuel sur GCP.
- L'équilibreur de charges externe est créé par le modèle et, par conséquent, toute exigence DNS spécifique pour l'adresse IP publique de l'équilibreur de charges est hors du champ d'application.
- Il est supposé que l'application se trouve derrière un équilibreur de charges créé par l'utilisateur et que l'ASA virtuel acheminera tout le trafic vers cet équilibreur de charges (au lieu d'envoyer directement le trafic à une adresse IP d'application spécifique).
- Les détails sur les besoins en matière de configurations de balises, de redondance et d'équilibreur de charges ne sont pas pris en compte.
- Les informations d'authentification d'ASA virtuel vous sont visibles comme suit :
  - Texte en clair dans le code sans serveur.
  - Dans toutes les instances du groupe d'instances.
  - Dans le modèle d'instance, si vous utilisez un compte GCP partagé.

  Ces données sensibles peuvent être protégées à l'aide du service de clé publique de GCP.

Important Cisco recommande de suivre régulièrement l'enregistrement d'ASA virtuel avec le serveur de licences pour vérifier si les ASA soumis à l'évolutivité à la hausse s'enregistrent auprès du serveur de licences comme prévu et si les instances ASA virtuel soumises à l'évolutivité à la baisse sont supprimées du serveur de licences.

Dépannage

Voici des scénarios d'erreurs courants et des conseils de débogage pour l'évolutivité automatique d'ASA virtuel pour GCP :

- `main.py` introuvable : assurez-vous que le paquet zip est créé uniquement à partir des fichiers. Vous pouvez accéder aux fonctions en nuage et vérifier l'arborescence des fichiers. Il ne devrait y avoir aucun dossier.
- Erreur lors du déploiement du modèle : assurez-vous que toutes les valeurs des paramètres dans « <> » sont renseignées dans .jinja et .yaml également, ou que le déploiement du même nom existe déjà.
- La fonction Google ne peut pas atteindre ASA virtuel : assurez-vous que le connecteur VPC est créé et que le même nom est mentionné dans le fichier de paramètres YAML.
- Échec de l'authentification pendant la connexion par SSH ASA virtuel : vérifiez que la paire de clés publique et privée est correcte.
- Échec de l'enregistrement de la licence : assurez-vous que le jeton d'ID de licence est correct. Assurez-vous également que la NAT en nuage est créée et qu'ASA virtuelpeut accéder à tools.cisco.com.

À propos de la traduction

Cisco peut fournir des traductions du présent contenu dans la langue locale pour certains endroits. Veuillez noter que des traductions sont fournies à titre informatif seulement et, en cas d’incohérence, la version anglaise du présent contenu prévaudra.