



ゲストシェルの有効化

- [ゲストシェルの有効化 \(1 ページ\)](#)
- [IAM インスタンスロールの作成 \(1 ページ\)](#)
- [Cisco Catalyst 8000V インスタンスへの IAM インスタンスロールの割り当て \(3 ページ\)](#)
- [新しいインスタンスへの IAM インスタンスロールの割り当て \(4 ページ\)](#)
- [ゲストシェルの例 \(5 ページ\)](#)

ゲストシェルの有効化

AWS で実行中の Cisco Catalyst 8000V でゲストシェルを有効にするには、IAM インスタンスロールを作成し、EC2 サービスとの信頼関係を確立します。IAM インスタンスロールを既存の Cisco Catalyst 8000V インスタンスに割り当てるか、それとも IAM インスタンスロールを新しい Cisco Catalyst 8000V インスタンスに割り当てるかを選択できます。

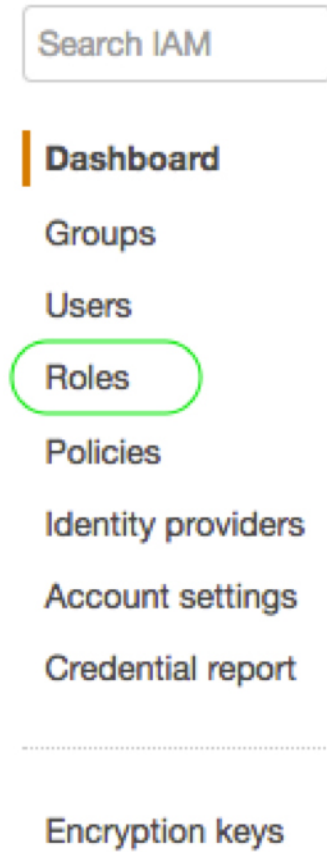
これらのタスクを実行する方法の詳細については、「Cisco Catalyst 8000V への IAM インスタンスロールの割り当て」と「新しい Cisco Catalyst 8000V への IAM インスタンスロールの割り当て」を参照してください。

次に、Cisco Catalyst 8000V でその後の設定手順を実行し、ゲストシェルを開きます。

IAM インスタンスロールの作成

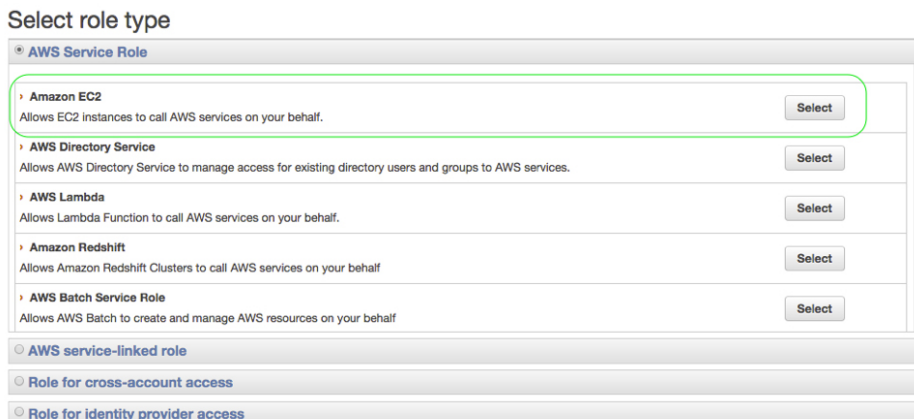
1. IAM ロールを作成する権限を持つ管理者として、AWS にサインインします。
2. [EC2] をクリックして、EC2 コンソールを開始します。
3. [IAM] をクリックして、IAM コンソールを開始します。
4. [Roles] をクリックします。

図 1: IAM インスタンスロール



5. [Create New Role] をクリックします。
6. アプリのロールの名前を入力します。
7. [Continue] をクリックします。
8. ロールタイプを選択します。

図 2: IAM インスタンスロールタイプ



366960

9. Amazon EC2 ロールタイプに関して、[Select] をクリックします。
この操作により、EC2 サービスとの信頼関係が確立されます。
10. [Set Permissions] で、[Select Policy Template] をクリックします。
11. [Select] をクリックして、テンプレート ([Amazon S3 Full Access] など) を選択します。
複数のサービスを選択できます。アクセスをさらに詳細に指定するには、このオプションを使用します。たとえば、IAM インスタンスロールに S3 バケットからの読み取りを許可する一方で、S3 バケットへの書き込みは許可しない設定にすることができます。
12. ロール名を入力します。
13. [Create Role] をクリックします。

Cisco Catalyst 8000V インスタンスへの IAM インスタンスロールの割り当て

IAM インスタンスロールの指定は、ゲストシェルへのアクセスに必須ではありません。とはいえ、指定しておくことで、キーまたはパスワードを使用して AWS アカウントの特定のエンティティにアクセスできるようになるため、Cisco Catalyst 8000V インスタンスのアカウント情報を保存する必要がなくなります。

- ステップ 1 [EC2] をクリックして、EC2 ダッシュボードを開きます。
- ステップ 2 一覧表示された Cisco Catalyst 8000V インスタンスのいずれかを選択し、右クリックして [Instance Setup] を選択します。次に、[Attach/Replace IAM Role] を選択します。
- ステップ 3 ドロップダウンリストから、以前に作成した IAM インスタンスロールを選択します。
- ステップ 4 Cisco Catalyst 8000V で次の CLI 設定コマンドを入力し、Cisco Catalyst 8000V を再起動します。

```
Router(config)# interface GigabitEthernet1
Router(config-if)# ip address dhcp
Router(config-if)# ip nat outside
Router(config-if)# exit
Router(config)# interface VirtualPortGroup0
Router(config-if)# ip address 192.168.35.1 255.255.255.0
Router(config-if)# ip nat inside
Router(config-if)# exit
Router(config)# ip nat inside source list GS_NAT_ACL interface GigabitEthernet1 overload
Router(config)# ip access-list standard GS_NAT_ACL
Router(config)# permit 192.168.0.0 0.0.255.255
Router(config)# app-hosting appid guestshell
Router(config-app-hosting)# vnic gateway1 virtualportgroup 0 guest-interface 0 guest-ipaddress
192.168.35.2 netmask 255.255.255.0 gateway 192.168.35.1 name-server 8.8.8.8 default
Router(config-app-hosting)# resource profile custom cpu 1500 memory 512
Router(config-app-hosting)# exit
Router(config)# exit
Router# guestshell enable
Router# guestshell run python
```

新しいインスタンスへの IAM インスタンスロールの割り当て

次の手順は、新しい Cisco Catalyst 8000V インスタンスの作成中に IAM インスタンスロールを Cisco Catalyst 8000V に割り当てる方法を示しています。

ステップ 1 EC2 インスタンスとして新しい Cisco Catalyst 8000V を起動し、インスタンスタイプを選択します。

ステップ 2 [Next: Configure Instance Details] をクリックします。

ステップ 3 次の 2 つのうちいずれかの手順を実行します。

- [IAM role] テキストボックスをクリックして、ドロップダウンリストから既存の IAM インスタンスロールを選択します。
- [Create new IAM role] をクリックして、新しい IAM インスタンスロールを作成します。

ステップ 4 Cisco Catalyst 8000V インスタンスで次の CLI 設定コマンドを入力し、インスタンスを再起動します。

```
Router(config)# interface GigabitEthernet1
Router(config-if)# ip address dhcp
Router(config-if)# ip nat outside
Router(config-if)# exit
Router(config)# interface VirtualPortGroup0
Router(config-if)# ip address 192.168.35.1 255.255.255.0
Router(config-if)# ip nat inside
Router(config-if)# exit
Router(config)# ip nat inside source list GS_NAT_ACL interface GigabitEthernet1 overload
Router(config)# ip access-list standard GS_NAT_ACL
Router(config)# permit 192.168.0.0 0.0.255.255
Router(config)# app-hosting appid guestshell
Router(config-app-hosting)# vnic gateway1 virtualportgroup 0 guest-interface 0 guest-ipaddress
192.168.35.2 netmask 255.255.255.0 gateway 192.168.35.1 name-server 8.8.8.8 default
Router(config-app-hosting)# resource profile custom cpu 1500 memory 512
```

```
Router(config-app-hosting)# exit
Router(config)# exit
Router# guestshell enable
Router# guestshell run python
```

ゲストシェルの例

次の例は、Cisco Catalyst 8000V インスタンスのゲストシェルでパッケージをダウンロードする方法と、他の便利なゲストシェルコマンドのいくつかを示しています。

1. yum コマンドまたは pip3 コマンドを使用してパッケージをインストールします。たとえば、[guestshell@guestshell ~] sudo pip3 install awscli コマンドを入力して、AWS CLI と Amazon SDK をインストールします。

```
[guestshell@guestshell ~]$ sudo pip3 install awscli
WARNING: Running pip install with root privileges is generally not a good idea. Try
`pip3 install --user` instead.
Collecting awscli
  Downloading
https://files.pythonhosted.org/packages/ce/38/6f2b06f0f00e6b381ac4741d0bf97eb2e3fa232382dfe31577154e207c/awscli-1.18.157-py2.py3-none-any.whl
(3.4MB)
  100% |#####| 3.4MB 369kB/s
Collecting colorama<0.4.4,>=0.2.5; python_version != "3.4" (from awscli)
  Downloading
https://files.pythonhosted.org/packages/c9/dc/45cdef1b4d119e9631663117e6c570e0802999b2f5e2c143c7a0a5cc5/colorama-0.4.3-py2.py3-none-any.whl
Collecting s3transfer<0.4.0,>=0.3.0 (from awscli)
  Downloading
https://files.pythonhosted.org/packages/69/79/e6af338b04e96cefd690f741d7db24547ff1f94240c997a26fa808b/s3transfer-0.3.3-py2.py3-none-any.whl
(69kB)
  100% |#####| 71kB 7.3MB/s
Collecting docutils<0.16,>=0.10 (from awscli)
  Downloading
https://files.pythonhosted.org/packages/22/cd/a6ae9591a619918cd55023bd151949c643d5f5b3f4fd7ee0c6e8/docutils-0.15.2-py3-none-any.whl
(547kB)
  100% |#####| 552kB 2.1MB/s
Collecting PyYAML<5.4,>=3.10; python_version != "3.4" (from awscli)
  Downloading
https://files.pythonhosted.org/packages/64/c2/b80047c7ac2478f950167c988a5411ed5572f35d1beff9cae07d321512c/PyYAML-5.3.1.tar.gz
(269kB)
  100% |#####| 276kB 3.6MB/s
Collecting rsa<=4.5.0,>=3.1.2; python_version != "3.4" (from awscli)
  Downloading
https://files.pythonhosted.org/packages/26/eb/8127f1bb0294f044121d20aac7785fbb810e159098447967a6103bdf96/rsa-4.5-py2.py3-none-any.whl
Collecting boto3==1.18.16 (from awscli)
  Downloading
https://files.pythonhosted.org/packages/2d/9e/afa41d0c91186808b7839d021e67e23c87b317aa4662d3f3cf/boto3-1.18.16-py2.py3-none-any.whl
(6.7MB)
  100% |#####| 6.7MB 173kB/s
Collecting pyasn1>=0.1.3 (from rsa<=4.5.0,>=3.1.2; python_version != "3.4"->awscli)
  Downloading
https://files.pythonhosted.org/packages/62/1e/a94e8635fa3e4cfc75060035481a2447ae76ff5ca5392970fa3053f/pyasn1-0.4.8-py2.py3-none-any.whl
(77kB)
  100% |#####| 81kB 7.5MB/s
Collecting urllib3<1.26,>=1.20; python_version != "3.4" (from
boto3==1.18.16->awscli)
  Downloading
https://files.pythonhosted.org/packages/9f/f0/a391d1463d1b1233795ca0f0ef383bb442339e68894702619e6987/urllib3-1.25.10-py2.py3-none-any.whl
(127kB)
```

```

100% |#####| 133kB 6.1MB/s
Collecting python-dateutil<3.0.0,>=2.1 (from boto3==1.18.16->awscli)
Downloading
https://files.pythonhosted.org/packages/41/70/3045c31148f87586247ae807091e1b306af21e5134d7615b/python_dateutil-2.8.1-py2.py3-none-any.whl
(227kB)
100% |#####| 235kB 4.0MB/s
Collecting jmespath<1.0.0,>=0.7.1 (from boto3==1.18.16->awscli)
Downloading
https://files.pythonhosted.org/packages/07/d5/5f01272b5fab231c9e0acc0448aa5c8621770920e3469c6e0139/jmespath-0.10.0-py2.py3-none-any.whl
Collecting six>=1.5 (from python-dateutil<3.0.0,>=2.1->boto3==1.18.16->awscli)
Downloading
https://files.pythonhosted.org/packages/ee/ff/43de5c0f13094d729fe40316a2a247463ff1c528248a4b04078a/six-1.15.0-py2.py3-none-any.whl
Installing collected packages: colorama, urllib3, six, python-dateutil, jmespath,
boto3, s3transfer, docutils, PyYAML, pyasn1, rsa, awscli
Running setup.py install for PyYAML ... done
Successfully installed PyYAML-5.3.1 awscli-1.18.157 boto3-1.18.16 colorama-0.4.3
docutils-0.15.2 jmespath-0.10.0 pyasn1-0.4.8 python-dateutil-2.8.1 rsa-4.5
s3transfer-0.3.3 six-1.15.0 urllib3-1.25.10
[guestshell@guestshell ~]$ aws s3 ls c8kv
Unable to locate credentials. You can configure credentials by running "aws configure"

```

2. AWS CLI をインストールしたら、aws s3 ls などの aws s3 コマンドを入力します。

```

[guestshell@guestshell ~]$ aws s3 ls c8kv
2020-10-14 19:44:08 433546509 upgrade.bin
[guestshell@guestshell ~]$

```

3. sudo pip3 install csr_aws_guestshell コマンドを使用して、サンプルスクリプトを含む Cisco Catalyst 8000V AWS パッケージをダウンロードできます。

例：

```

[guestshell@guestshell ~]$ sudo pip3 install csr_aws_guestshell
WARNING: Running pip install with root privileges is generally not a good idea. Try
`pip3 install --user` instead.
Collecting csr_aws_guestshell
Downloading
https://files.pythonhosted.org/packages/42/a7/c72726166f80988223ef448f5d7fa2cf8809525a1199161281cd080a/csr_aws_guestshell-0.0.17.dev.tar.gz
Collecting awscli (from csr_aws_guestshell)
Downloading
https://files.pythonhosted.org/packages/23/1b/265d14e18a8b2341375991cc22021233f3c3f1d10990170ce/awscli-1.18.162-py2.py3-none-any.whl
(3.4MB)
100% |#####| 3.4MB 352kB/s
Collecting boto (from csr_aws_guestshell)
Downloading
https://files.pythonhosted.org/packages/23/10/c0578c27298029e445e472a19191e20b1821b1662ec7f2ca1d0c523/boto-2.49.0-py2.py3-none-any.whl
(1.4MB)
100% |#####| 1.4MB 794kB/s
Collecting boto3 (from csr_aws_guestshell)
Downloading
https://files.pythonhosted.org/packages/30/3c/c965c3981e689c933c65a27a69afcd758850a7994ad3ac6599e8a/boto3-1.16.2-py2.py3-none-any.whl
(129kB)
100% |#####| 133kB 7.2MB/s
Collecting rsa<=4.5.0,>=3.1.2; python_version != "3.4" (from
awscli->csr_aws_guestshell)
Downloading
https://files.pythonhosted.org/packages/26/f8/8127f1ba0294f044121d20aac7785feb810e159098447967a6103edf096/rsa-4.5-py2.py3-none-any.whl
Collecting botocore==1.19.2 (from awscli->csr_aws_guestshell)
Downloading
https://files.pythonhosted.org/packages/1f/96/35fa364675cf17e3a190ae08716c4078ca86a2ef071d32c886c52cc/botocore-1.19.2-py2.py3-none-any.whl
(6.7MB)
100% |#####| 6.7MB 164kB/s
Collecting PyYAML<5.4,>=3.10; python_version != "3.4" (from awscli->csr_aws_guestshell)

```

```

Downloading
https://files.pythhosted.org/packages/64/c2/b80047c7ac2478f9501676c988a5411ed5572f35d1beff9cae07d321512c/PyYAML-5.3.1.tar.gz
(269kB)
100% |#####| 276kB 3.6MB/s
Collecting s3transfer<0.4.0,>=0.3.0 (from awscli->csr_aws_guestshell)
Downloading
https://files.pythhosted.org/packages/69/79/e6af38b0de96ef0b60f74d7db2457ff1f9424099a26fa808b/s3transfer-0.3.3-py2.py3-none-any.whl
(69kB)
100% |#####| 71kB 7.6MB/s
Collecting docutils<0.16,>=0.10 (from awscli->csr_aws_guestshell)
Downloading
https://files.pythhosted.org/packages/22/cd/a6a959da619918cd550234db15194c644c5553f4ff7ee0c6e8/docutils-0.15.2-py3-none-any.whl
(547kB)
100% |#####| 552kB 1.9MB/s
Collecting colorama<0.4.4,>=0.2.5; python_version != "3.4" (from
awscli->csr_aws_guestshell)
Downloading
https://files.pythhosted.org/packages/c9/db/45cdef1b4d119e963163117e6c5708a08029992c2fee2c1437a05cc5/colorama-0.4.3-py2.py3-none-any.whl
Collecting jmespath<1.0.0,>=0.7.1 (from boto3->csr_aws_guestshell)
Downloading
https://files.pythhosted.org/packages/07/db/5f00127b6fa231c9e0acc0438eaf5c862170920e346c6a0139/jmespath-0.10.0-py2.py3-none-any.whl
Collecting pyasn1>=0.1.3 (from rsa<=4.5.0,>=3.1.2; python_version !=
"3.4"->awscli->csr_aws_guestshell)
Downloading
https://files.pythhosted.org/packages/62/1e/a4a8635fa3e4cfc7f506035480e2447ae76f6c5a53932970fe3053f/pyasn1-0.4.8-py2.py3-none-any.whl
(77kB)
100% |#####| 81kB 9.4MB/s
Collecting urllib3<1.26,>=1.25.4; python_version != "3.4" (from
botocore==1.19.2->awscli->csr_aws_guestshell)
Downloading
https://files.pythhosted.org/packages/56/aa/4ef5a67a9a62505b1245d526233cd1d153462b8f899fa4ef582/urllib3-1.25.11-py2.py3-none-any.whl
(127kB)
100% |#####| 133kB 6.5MB/s
Collecting python-dateutil<3.0.0,>=2.1 (from
botocore==1.19.2->awscli->csr_aws_guestshell)
Downloading
https://files.pythhosted.org/packages/d4/70/604503b8e875862407ae8970980b3062ae5d13d7861db/python-dateutil-2.8.1-py2.py3-none-any.whl
(227kB)
100% |#####| 235kB 4.6MB/s
Collecting six>=1.5 (from
python-dateutil<3.0.0,>=2.1->botocore==1.19.2->awscli->csr_aws_guestshell)
Downloading
https://files.pythhosted.org/packages/ee/ff/48ae50f013094d729fe40316a2a24774c3ff1c52b92e84d04078a/six-1.15.0-py2.py3-none-any.whl
Installing collected packages: pyasn1, rsa, urllib3, six, python-dateutil, jmespath,
botocore, PyYAML, s3transfer, docutils, colorama, awscli, boto, boto3,
csr-aws-guestshell
Running setup.py install for PyYAML ... done
Running setup.py install for csr-aws-guestshell ... done
Successfully installed PyYAML-5.3.1 awscli-1.18.162 boto-2.49.0 boto3-1.16.2
botocore-1.19.2 colorama-0.4.3 csr-aws-guestshell-0.0.17.dev0 docutils-0.15.2
jmespath-0.10.0 pyasn1-0.4.8 python-dateutil-2.8.1 rsa-4.5 s3transfer-0.3.3 six-1.15.0
urllib3-1.25.11
    
```

次のスクリプトが `csr_aws_guestshell` パッケージに含まれています。

`get-metadata.py` : AWS からインスタンスメタデータを取得して出力します。

`get-route-table.py` : VPC のインスタンスをルート、ルートテーブル、関連付けを含めて取得します。

`save-config-to-s3.py` : Cisco IOS XE CLI コマンドを S3 バケットに保存します。

`save-tech-support-to-s3.py` : テクニカルサポートの出力を S3 バケットに保存します。

load-bin-from-s3.py : Cisco Catalyst 8000V 用の .bin ファイルをダウンロードしてリロードします。

get-stat-drop.py : CLI 統計情報を取得して、CloudWatch にプッシュします。

capture-interface.py : 監視に使用する Cisco IOS XE CLI コマンドを設定して、一定期間パケットをキャプチャした後、ファイルを S3 にアップロードします。

4. 次の例では、load-bin-from-s3.py スクリプトが S3 からバイナリをロードし、Cisco Catalyst 8000V イメージを起動します。

```
[guestshell@guestshell ~]$ load-bin-from-s3.py csr1kv ultra_167.bin  
/bootflash/ultra_167.bin 446866343 / 446866343 (100.00%)  
Download Complete
```



(注) csr_aws_guestshell パッケージは、引き続き Cisco Catalyst 8000V と連動します。

翻訳について

このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。