演習 - ネットワーク セキュリティ グループを使用して VM へのネットワーク アクセスを構成する

100 XP

10 分

このモジュールでは、サンドボックスを完了する必要があります。 **サンドボックス**で無料リソースにアクセスできます。 お客様の個人のサブスクリプションに対する課金は行われません。 サンドボックスを使用できるのは、Microsoft Learn のトレーニングを完了するためだけです。 その他の目的で利用することは禁止されており、サンドボックスに永久にアクセスできなくなる可能性があります。

サインインしてサンドボックスをアクティブにする

この演習では、Azure で実行される仮想マシン (VM) へのネットワーク アクセスを構成します。

Linux VM を作成し、その VM に Nginx という一般的な Web サーバーをインストールすることから始めます。 次に、Web サーバーにアクセスできるように、ポート 80 (HTTP) で受信アクセスを許可するネットワーク セキュリティ グループ (NSG) 規則を作成します。

ネットワーク設定を含めて、VM を作成して管理するための方法はたくさんあります。 たとえば、Azure portal、Azure CLI、Azure PowerShell、Azure Resource Manager (ARM) テンプレートを使用できます。

ここでは、Azure CLI を使用します。 Azure CLI を使用して、Azure に接続し、Azure リソースに対して管理コマンドを実行できます。 他のコマンドライン インターフェイスと同様に、ターミナルからコマンドを直接実行することも、Bash スクリプトや PowerShell スクリプトにコマンドを追加することもできます。 Azure CLI は、Windows、macOS、または Linux で実行されます。

ここでは、Azure Cloud Shell から Azure CLI にアクセスします。 Cloud Shell は、Azure リソースの開発と管理を行うために使用するブラウザーベースのシェル エクスペリエンスです。 Cloud Shell は、クラウド上で動作する対話型コンソールと考えてください。

Azure CLI や Cloud Shell を初めて使用する場合でも、単純に手順に従ってください。

Linux 仮想マシンを作成し、Nginx をインストールする

次の Azure CLI コマンドを使用して Linux VM を作成し、Nginx をインストールします。 VM が作成されたら、カスタム スクリプト拡張機能を使用して Nginx をインストールします。 カスタム

スクリプト拡張機能は、Azure VM でスクリプトをダウンロードして実行する簡単な方法です。 これは、VM が稼働してからシステムを構成できるさまざまな方法の 1 つに過ぎません。

1. Cloud Shell から次の `az vm create` コマンドを実行して、Linux VM を作成します。

   Azure CLI コピー

   ```
   az vm create \
     --resource-group [sandbox resource group name] \
     --name my-vm \
     --image UbuntuLTS \
     --admin-username azureuser \
     --generate-ssh-keys
   ```

   VM が作成されるまで、しばらく時間がかかります。

   VM に **my-vm** という名前を付けます。 この名前を使用して、後の手順でこの VM を参照します。

2. 次の `az vm extension set` コマンドを実行して、VM 上に Nginx を構成します。

   Azure CLI コピー

   ```
   az vm extension set \
     --resource-group [sandbox resource group name] \
     --vm-name my-vm \
     --name customScript \
     --publisher Microsoft.Azure.Extensions \
     --version 2.1 \
     --settings '{"fileUris":
   ["https://raw.githubusercontent.com/MicrosoftDocs/mslearn-welcome-to-
   azure/master/configure-nginx.sh"]}' \
     --protected-settings '{"commandToExecute": "./configure-nginx.sh"}'
   ```

   このコマンドでは、カスタム スクリプト拡張機能を使用して、VM 上で Bash スクリプトを実行します。 このスクリプトは GitHub に格納されています。

   コマンドを実行しながら、別のブラウザー タブで Bash スクリプトを調べることができます。

   まとめると、スクリプトでは次のことが行われます。

   a. `apt-get update` を実行して、インターネットから最新のパッケージ情報をダウンロードします。 この手順によって、次のコマンドで Nginx パッケージの最新バージョンを確実に見つけることができます。
   b. Nginx をインストールします。
   c. ホームページ */var/www/html/index.html* を設定して、VM のホスト名を含むウェルカム メッセージを出力します。

Web サーバーにアクセスする

この手順では、VM の IP アドレスを取得し、Web サーバーのホーム ページへのアクセスを試します。

1. 次の `az vm list-ip-addresses` コマンドを実行して、VM の IP アドレスを取得し、その結果を Bash 変数として格納します。

   Azure CLI コピー

   ```
   IPADDRESS="$(az vm list-ip-addresses \
     --resource-group [sandbox resource group name] \
     --name my-vm \
     --query "[].virtualMachine.network.publicIpAddresses[*].ipAddress" \
     --output tsv)"
   ```

2. 次の `curl` コマンドを実行して、ホームページをダウンロードします。

   Bash コピー

   ```
   curl --connect-timeout 5 http://$IPADDRESS
   ```

   `--connect-timeout` 引数で、接続が発生するまで最大 5 秒の時間を許可することを指定します。

   5 秒後に、接続がタイムアウトしたことを示すエラー メッセージが表示されます。

   出力 コピー

   ```
   curl: (28) Connection timed out after 5001 milliseconds
   ```

   このメッセージは、タイムアウト期間内に VM にアクセスできなかったことを意味します。

3. 省略可能な手順として、ブラウザーから Web サーバーへのアクセスを試してみます。

   a. 次を実行して、VM の IP アドレスをコンソールに出力します。

      Bash コピー

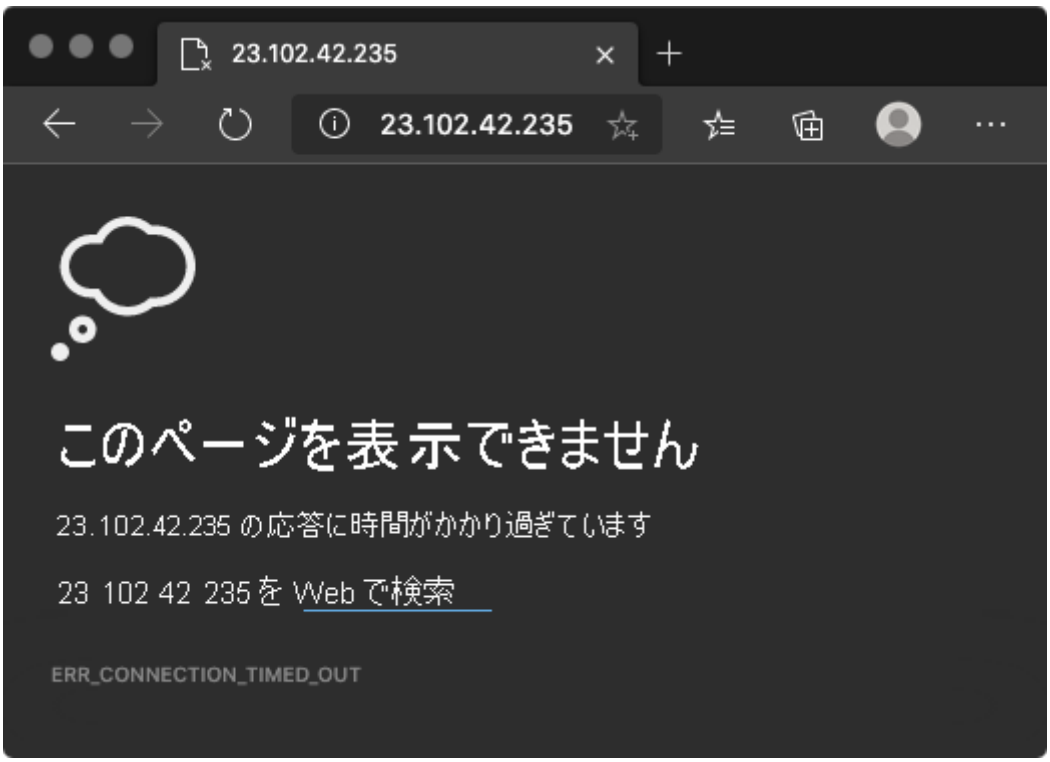
      ```
      echo $IPADDRESS
      ```

      たとえば *23.102.42.235* のような IP アドレスが表示されます。

   b. 表示された IP アドレスをクリップボードにコピーします。

   c. 新しいブラウザー タブを開き、Web サーバーに移動します。

しばらくすると、接続が行われていないことがわかります。 ブラウザーがタイムアウトするまで待つと、次のように表示されます。



このブラウザー タブは後で使用するため、開いたままにしておきます。

現在のネットワーク セキュリティ グループ規則を一覧表示する

Web サーバーにアクセスすることができませんでした。 理由を明らかにするために、現在の NSG 規則を調べてみましょう。

1. 次の `az network nsg list` コマンドを実行して、VM に関連付けられているネットワーク セキュリティ グループを一覧表示します。

Azure CLI コピー

```
az network nsg list \
  --resource-group [sandbox resource group name] \
  --query '[].name' \
  --output tsv
```

次のように表示されます。

出力 コピー

```
my-vmNSG
```

Azure 上のすべての VM は、少なくとも 1 つのネットワーク セキュリティ グループに関連付けられます。 この例では、Azure によって、*my-vmNSG* という名前の NSG が作成されました。

2. 次の `az network nsg rule list` コマンドを実行して、*my-vmNSG* という名前の NSG に関連付けられている規則を一覧表示します。

Azure CLI コピー

```
az network nsg rule list \
  --resource-group [sandbox resource group name] \
  --nsg-name my-vmNSG
```

JSON 形式の大きなテキスト ブロックが出力として表示されます。 次の手順で、この出力を読みやすくするための似たようなコマンドを実行します。

3. `az network nsg rule list` コマンドをもう一度実行します。

今回は、 `--query` 引数を使用して、各規則の名前、優先度、影響を受けるポート、およびアクセス (**許可** または **拒否**) のみを取得します。

`--output` 引数によって、読みやすくするために出力が表として書式設定されます。

Azure CLI コピー

```
az network nsg rule list \
  --resource-group [sandbox resource group name] \
  --nsg-name my-vmNSG \
  --query '[].{Name:name, Priority:priority, Port:destinationPortRange,
Access:access}' \
  --output table
```

次のように表示されます。

出力 コピー

```
Name                Priority    Port    Access
------------------  ----------  ------  --------
default-allow-ssh   1000        22      Allow
```

既定の規則 *default-allow-ssh* が表示されます。 この規則によって、ポート 22 (SSH) 経由の受信接続が許可されます。 SSH (Secure Shell) は、管理者がシステムにリモートでアクセスできるようにするために、Linux で使用されるプロトコルです。

この規則の優先度は 1000 です。 規則は優先度順に処理され、小さい数値を持つ規則が大きな数値のものよりも前に処理されます。

既定では、Linux VM の NSG では、ポート 22 でのネットワーク アクセスのみが許可されます。 これにより、管理者はシステムにアクセスできます。 さらに、HTTP 経由のアクセスを許可するポート 80 での受信接続も許可する必要があります。

ネットワーク セキュリティ ルールを作成する

ここでは、ポート 80 (HTTP) での受信アクセスを許可するネットワーク セキュリティ規則を作成します。

1. 次の `az network nsg rule create` コマンドを実行して、ポート 80 での受信アクセスを許可する *allow-http* という名前の規則を作成します。

   Azure CLI コピー

   ```
   az network nsg rule create \
     --resource-group [sandbox resource group name] \
     --nsg-name my-vmNSG \
     --name allow-http \
     --protocol tcp \
     --priority 100 \
     --destination-port-range 80 \
     --access Allow
   ```

   学習目的のために、ここでは優先度を 100 に設定します。 この例では優先度は重要ではありません。 ポート範囲が重複している場合は、優先度を考慮する必要があります。

2. 構成を検証するために、 `az network nsg rule list` を実行して、更新された規則の一覧を表示します。

   Azure CLI コピー

   ```
   az network nsg rule list \
     --resource-group [sandbox resource group name] \
     --nsg-name my-vmNSG \
     --query '[].{Name:name, Priority:priority, Port:destinationPortRange,
   Access:access}' \
     --output table
   ```

   default-allow-ssh 規則と、新しい規則である *allow-http* の両方が表示されます。

   出力 コピー

   ```
   Name               Priority    Port    Access
   -----------------  ----------  ------  --------
   default-allow-ssh  1000        22      Allow
   allow-http         100         80      Allow
   ```

Web サーバーにもう一度アクセスする

これでポート 80 へのネットワーク アクセスを構成したので、Web サーバーにもう一度アクセスしてみましょう。

1. 先ほど実行したのと同じ `curl` コマンドを実行します。

   Bash コピー

   ```
   curl --connect-timeout 5 http://$IPADDRESS
   ```

   次のように表示されます。

   HTML コピー

   ```
   <html><body><h2>Welcome to Azure! My name is my-vm.</h2></body></html>
   ```

2. 省略可能な手順として、Web サーバーを指すようにブラウザー タブを更新します。

   次のように表示されます。

クリーンアップ

このモジュールを完了したら、サンド ボックスは、リソースを自動的にクリーンアップします。

独自のサブスクリプションを使用している場合は、プロジェクトの最後に、作成したリソースがまだ必要かどうかを確認してください。 リソースを実行したままにすると、お金がかかる場合が

あります。 リソースを個別に削除するか、リソース グループを削除してリソースのセット全体を削除することができます。

お疲れ様でした。 実際には、必要な受信と送信のネットワーク アクセス規則を含むスタンドアロン ネットワーク セキュリティ グループを作成できます。 同じ目的で使用される VM が複数ある場合は、作成時に各 VM にその NSG を割り当てることができます。 この手法を使用して、複数の VM へのネットワーク アクセスを、単一の一元的な規則セットで制御できます。