

# AWS上での監視サーバー(Zabbix)構築【5.AWSとCMLのVPN接続】

AWS



## AWS上での監視サーバー(Zabbix)構築【5.AWSとCMLのVPN接続】

2021.09.11 2021.09.05

監視サーバーをAWS上で構築し、CML上のネットワーク機器/サーバーを監視します。監視ソフトウェアはZabbixを利用します。

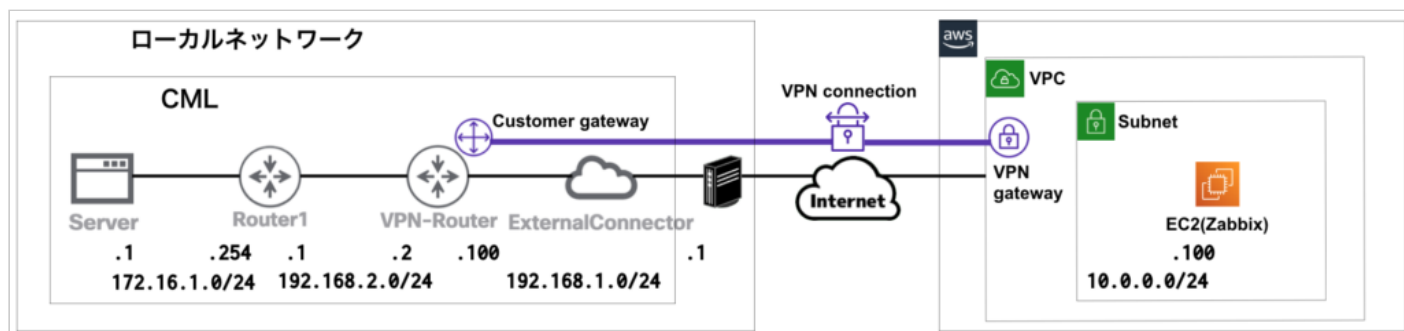
[【前回】AWS上での監視サーバー\(Zabbix\)構築【4.Zabbixのインストールと設定】](#)

[【次回】AWS上での監視サーバー\(Zabbix\)構築【6.監視登録\(ICMPノード監視\)】](#)

## ネットワーク構成

下記のネットワーク環境を構築し、AWS上のEC2(Zabbixサーバー)から、CML上のネットワーク機器/サーバーを監視できるようにしていきます。

[【参考】AWSサイト間VPNの構築（1.AWSの基本設定）](#)



# AWSのVPN構築

## カスタマーゲートウェイの作成

ローカルネットワーク側のVPNの起点となるカスタマーゲートウェイを作成します。まず、自身が利用しているグローバルIPアドレスを確認します。ここでは[CMAN](http://www.cman.jp/network/)のIPアドレス確認ページで確認しています。



VPC→仮想プライベートネットワーク(VPN)の画面から操作します。

カスタマーゲートウェイ → 「カスタマーゲートウェイの作成」をクリックします。



「zabbix-test-cgw」という名前で、ルーティングは「静的」を選択し、IPアドレスは確認した「グローバルIPアドレス」を設定します。

カスタマーゲートウェイ > カスタマーゲートウェイの作成

### カスタマーゲートウェイの作成

ゲートウェイの外部インターフェイスのインターネットでルーティング可能な IP アドレスを指定します。このアドレスは静的である必要があります。また、ネットワークアドレス変換 (NAT) を実行するデバイスの背後のアドレスを使用できます。動的なルーティングでは、ゲートウェイのボーダーゲートウェイプロトコル (BGP) 自律システム番号 (ASN) も指定します。これはパブリックまたはプライベート ASN (64512~65534 の範囲内のものなど) とすることができます。

名前

ルーティング ☐ 動的 ☒ 静的

IP アドレス

Certificate ARN

Device

\* 必須

キャンセル [カスタマーゲートウェイの作成](#)

「使用可能」となれば、作成完了です。

New VPC Experience  
Tell us what you think

カスタマーゲートウェイの作成 アクション

状態: available フィルターの追加

Name	ID	状態	タイプ	IP アドレス	BGP ASN	Certificate ARN
zabbix-test-cgw	cgw-	使用可能	ipsec.1		65000	

カスタマーゲートウェイ: cgw-

詳細 タグ

ID	タイプ	BGP ASN	Device	状態	IP アドレス	Certificate ARN
cgw-	ipsec.1	65000	-	使用可能		

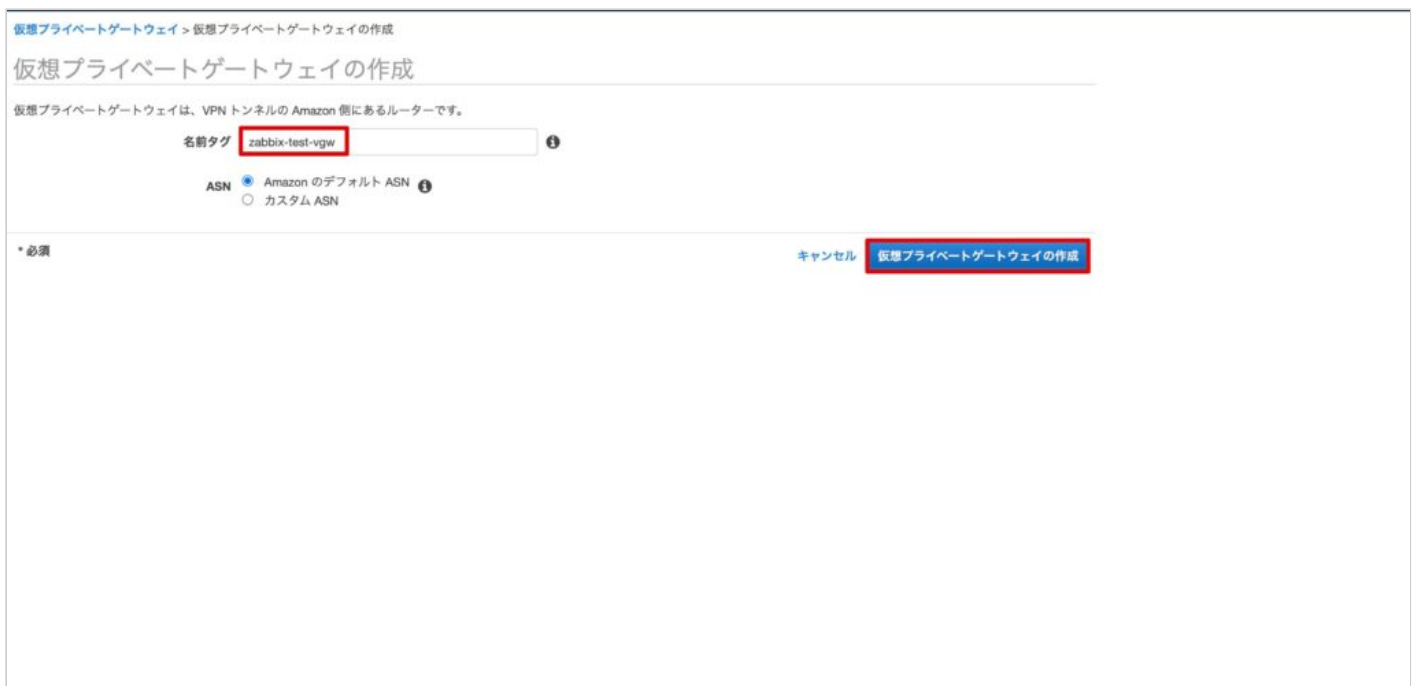
## 仮想プライゲートゲートウェイの作成

AWS側のVPNの起点となる仮想プライベートゲートウェイを作成します。

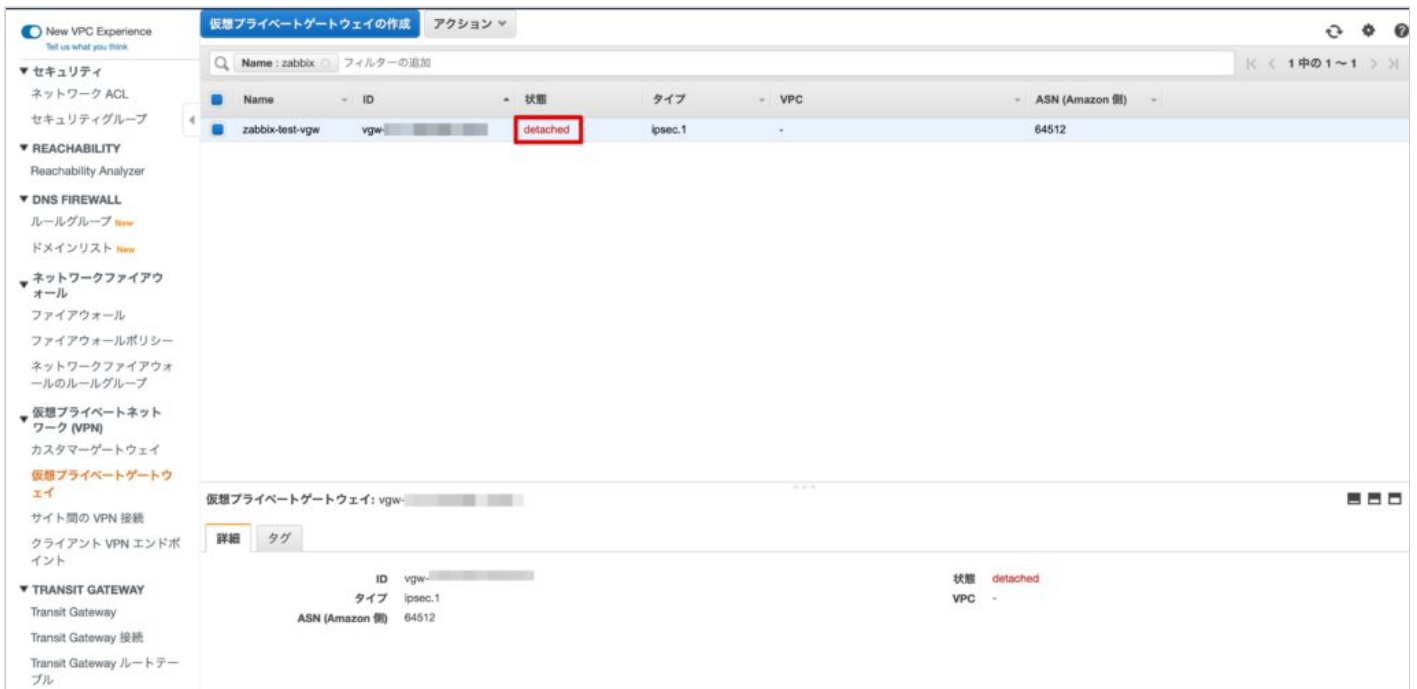
仮想プライベートゲートウェイ → 「仮想プライベートゲートウェイの作成」をクリックします。



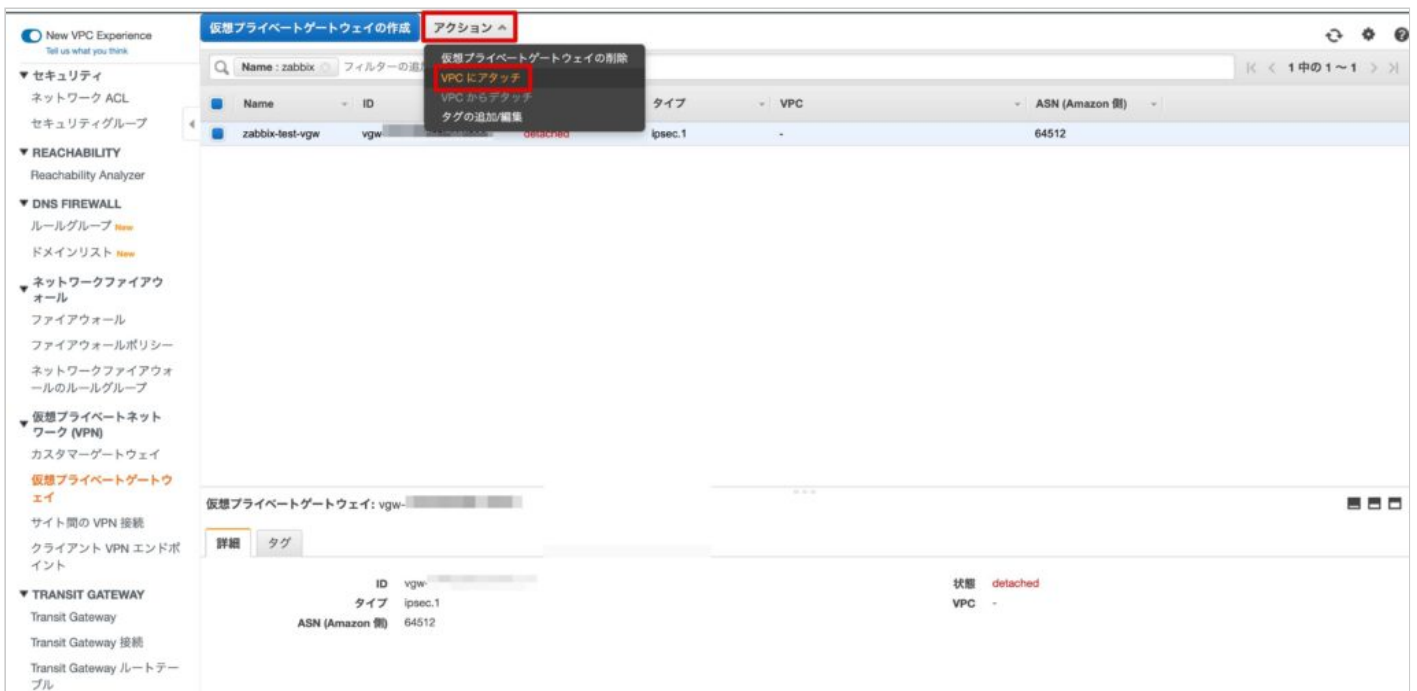
「zabbix-test-vgw」という名前で作成しています。



作成後は状態が「detached」となっているため、VPCにアタッチします。



「アクション」 → 「VPCにアタッチ」を選択します。



VPC(aws-zabbix-test)を選択しアタッチします。

仮想プライベートゲートウェイ &gt; VPC にアタッチ

## VPC にアタッチ

仮想プライベートゲートウェイにアタッチする VPC を選択します。

仮想プライベートゲートウェイ ID vgw-

VPC\* vpc-

キャンセル

はい、アタッチします

「attaching」の状態を経由し、

Name	ID	状態	タイプ	VPC	ASN (Amazon 側)
zabbix-test-vgw	vgw-	attaching	ipsec.1	vpc-   aws-zabbix-test	64512

「attached」となれば、作成完了です。

Name	ID	状態	タイプ	VPC	ASN (Amazon 側)
zabbix-test-vgw	vgw-	attached	ipsec.1	vpc-   aws-zabbix-test	64512

## ルートテーブルの設定

仮想プライベートゲートウェイからVPC内にルートを伝播するための設定を行います。

ルートテーブルの「ルート伝播」タブを選択し、「ルート伝播の編集」をクリックします。

このスクリーンショットは、AWS Management Consoleの「ルートテーブル」詳細ページを示しています。左側のナビゲーションメニューには「VIRTUAL PRIVATE CLOUD」の下に「ルートテーブル」が選択されています。中央の「ルート伝播」セクションで、1つの伝播がリストアップされています。この伝播は「仮想プライベートゲートウェイ」に関連付けられており、そのステータスは「いいえ」です。右側の「ルート伝播の編集」リンクが赤い枠で囲まれています。

作成した仮想プライベートゲートウェイの伝播の「有効化」にチェックを入れます。

このスクリーンショットは、AWS Management Consoleの「ルート伝播の編集」ページを示しています。左側のナビゲーションメニューには「VIRTUAL PRIVATE CLOUD」の下に「ルートテーブル」が選択されています。中央の「ルート伝播の編集」セクションで、1つの伝播がリストアップされています。この伝播は「仮想プライベートゲートウェイ」に関連付けられており、そのステータスは「有効化」です。右側の「有効化」チェックボックスが赤い枠で囲まれています。

伝播が「はい」となれば、設定完了です。





## サイト間のVPN接続の作成

カスタマーゲートウェイと仮想プライベートゲートウェイの間でVPNを構築するため、サイト間のVPN接続を作成します。

サイト間のVPN接続 → 「VPN接続の作成」をクリックします。

※VPN接続を作成すると利用料金が発生します※ 利用料金の説明 [→こちら](#)



下記を設定します。

名前タグ : zabbix-test-vpn ※任意の名前

ターゲットゲートウェイタイプ : 仮想プライベートゲートウェイ



仮想プライベートゲートウェイ：作成した仮想プライベートゲートウェイを選択

カスタマーゲートウェイ：既存

カスタマーゲートウェイID：作成したカスタマーゲートウェイを選択

ルーティングオプション：静的

静的IPプレフィックス：下記を追加

– 192. 168. 1. 0/24 (ローカル

ネットワーク)

– 192. 168. 2. 0/24 (CMLのル

ータ間セグメント)

– 172. 16. 1. 0/24 (サーバー

接続セグメント)

VPN 接続 > VPN 接続の作成

### VPN 接続の作成

VPN 接続を使用して接続するターゲットゲートウェイとカスタマーゲートウェイを選択します。ターゲットゲートウェイの情報を入力済みである必要があります。

名前タグ

ターゲットゲートウェイタイプ ☒ 仮想プライベートゲートウェイ  
☐ Transit Gateway

仮想プライベートゲートウェイ

カスタマーゲートウェイ ☒ 既存  
☐ 新規

カスタマーゲートウェイ ID

ルーティングオプション ☐ 動的 (BGP が必要)  
☒ 静的

静的 IP プレフィックス

IP プレフィックス	ソース	状態
192.168.1.0/24	-	-
192.168.2.0/24	-	-
172.16.1.0/24	-	-

別のルールの追加

トンネル内部 IP バージョン ☒ IPv4  
☐ IPv6

トンネルオプションは設定変更不要です。

### トンネルオプション

CIDR 内のトンネルと VPN トンネルの事前共有キーをカスタマイズします。未指定のトンネルオプションは、Amazon によってランダムに生成されます。

トンネル 1 の内部 IPv4 CIDR

トンネル 1 の事前共有キー

トンネル 2 の内部 IPv4 CIDR

トンネル 2 の事前共有キー

トンネル 1 の詳細オプション ☒ デフォルトオプションを使用  
☐ トンネル 1 オプションを編集

トンネル 2 の詳細オプション ☒ デフォルトオプションを使用  
☐ トンネル 2 オプションを編集

このステップを完了すると、VPN 接続料金が発生します。 [料金を表示](#)

\* 必須

[キャンセル](#) [VPN 接続の作成](#)

作成したVPN接続を確認します。

「詳細」タブで、「仮想プライベートゲートウェイ」と「カスタマーゲートウェイ」と「VPC」が正しく

設定されていることを確認します。

The screenshot shows the AWS VPC console interface. On the left is a navigation menu with options like 'VPC ダッシュボード', 'サブネット', 'ルートテーブル', etc. The main area is titled 'VPN 接続の作成' (VPN Connections). Below this, there's a table listing VPN connections. The connection 'vpn-' is highlighted. Below the table, the '詳細' (Details) tab is active, showing various attributes of the VPN connection. Red boxes highlight specific values: 'vpn-' for the VPN ID, 'vgw-' for the virtual gateway ID, 'aws-zabbix-test' for the VPC ID, and 'cgw-' for the customer gateway ID.

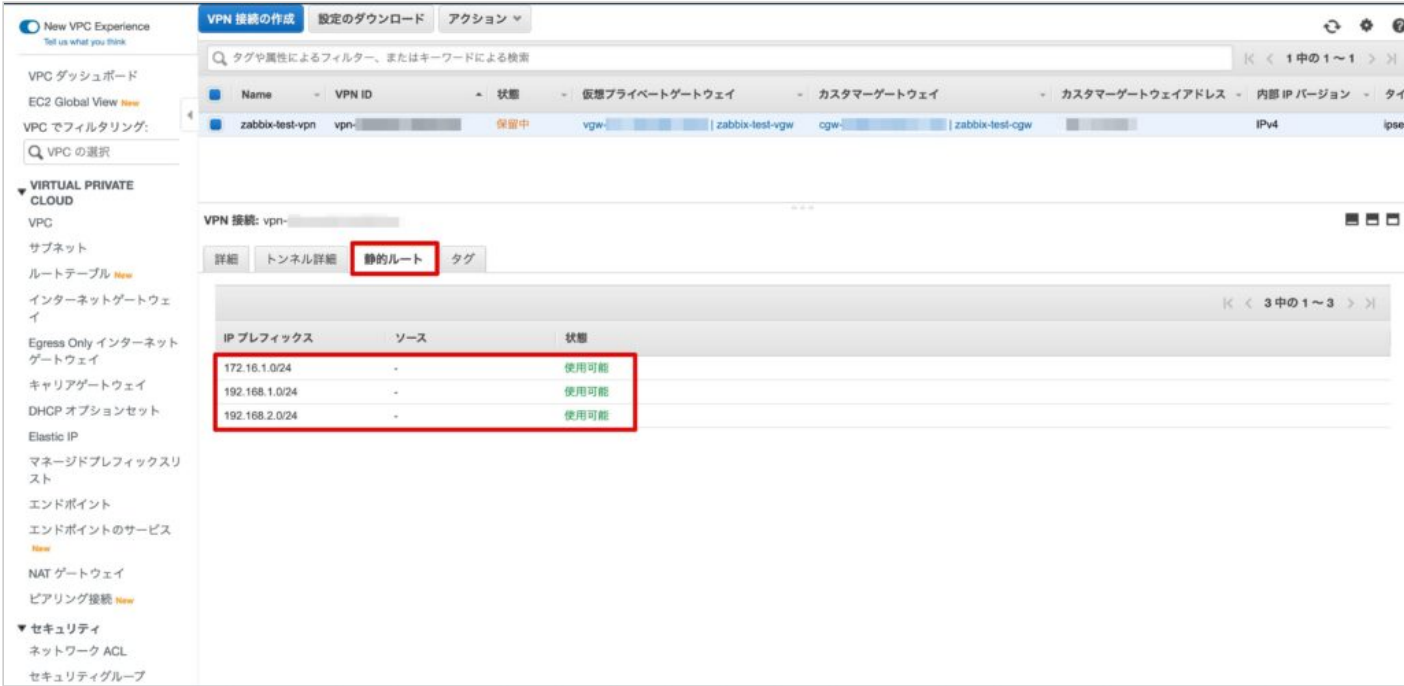
「トンネル詳細」タブで、トンネルの設定を確認します。

AWSのVPN接続では、冗長化のためにデフォルトで2つのトンネルが作成されます。

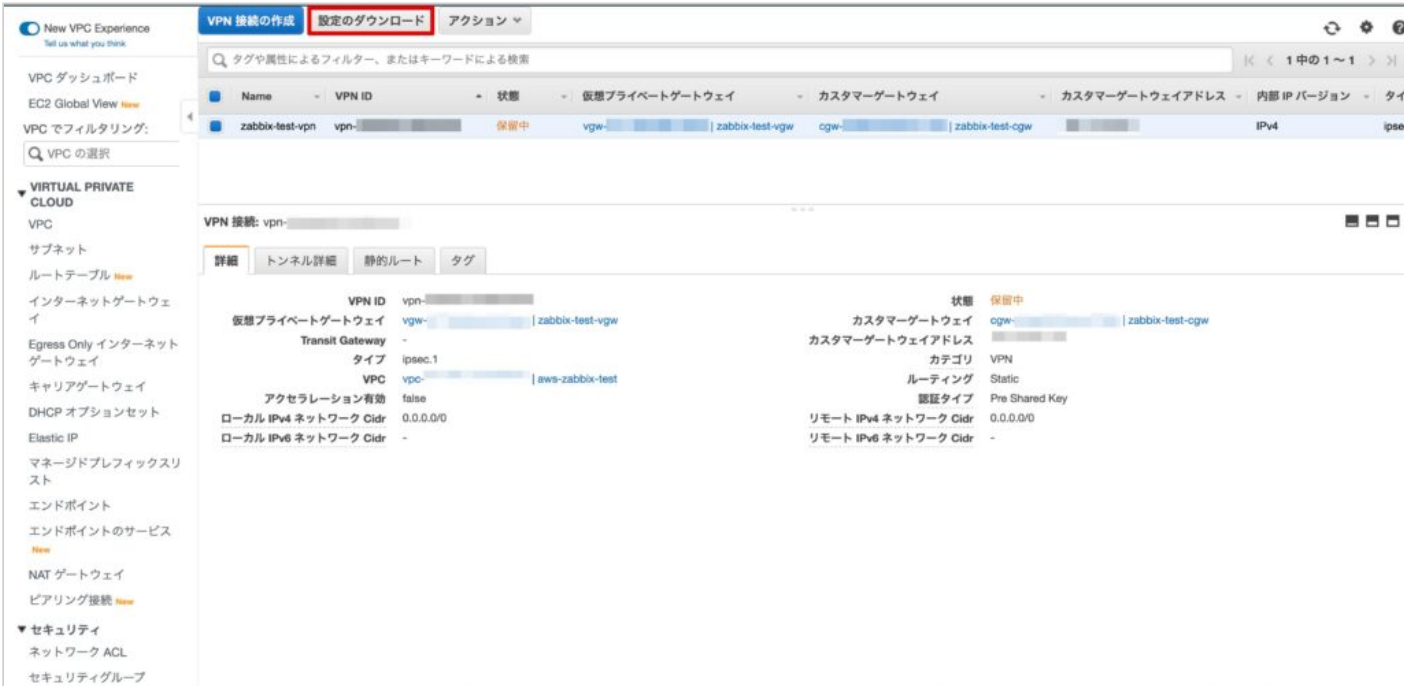
外部IPアドレスは「グローバルIPアドレス」、トンネルインターフェースのIPアドレスは「リンクローカルアドレス」が利用されます。

This screenshot shows the 'トンネル詳細' (Tunnel Details) tab in the AWS VPC console. It displays a table of tunnels. Two tunnels are listed: 'Tunnel 1' and 'Tunnel 2'. Both have a status of 'ダウン' (Down). The 'Tunnel 1' and 'Tunnel 2' rows are highlighted with red boxes. Below the table, there's a section for 'トンネル 1 のオプション' (Tunnel 1 Options) with various configuration parameters like 'フェーズ 1 暗号化アルゴリズム' and 'フェーズ 2 暗号化アルゴリズム'.

「静的ルート」タブで、設定したIPプレフィックスが「使用可能」となっていることを確認します。



ネットワーク機器の設定サンプルをダウンロードします。  
「設定のダウンロード」をクリックします。

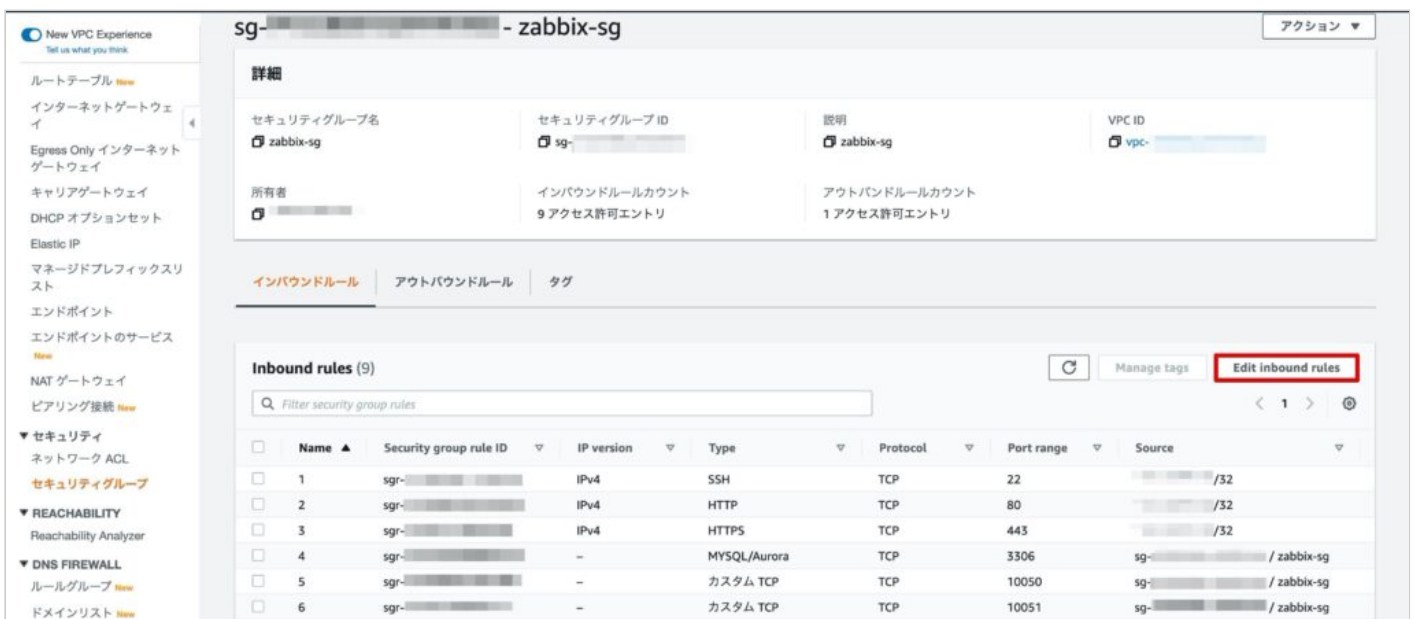


ベンダーは「Cisco Systems, Inc.」、プラットフォームは「Cisco ASR 1000」を選択し、ダウンロードをクリックします。

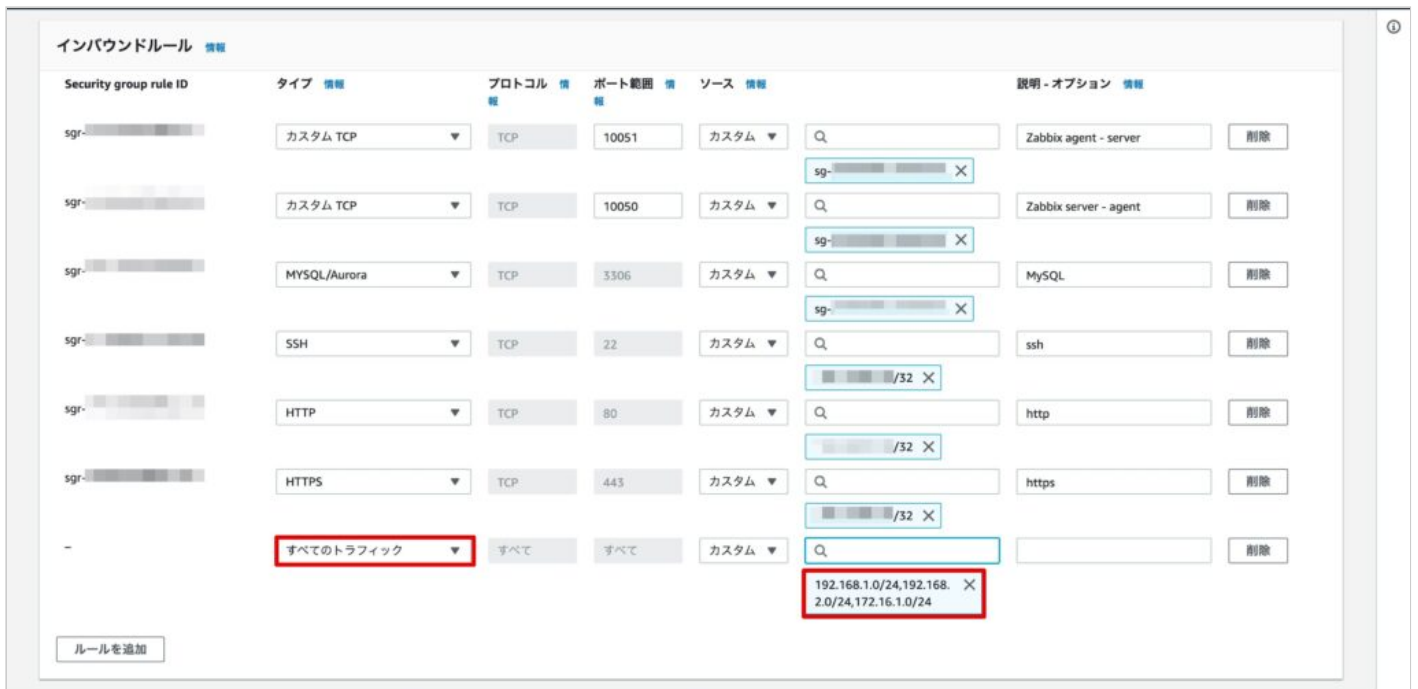


## セキュリティグループの設定

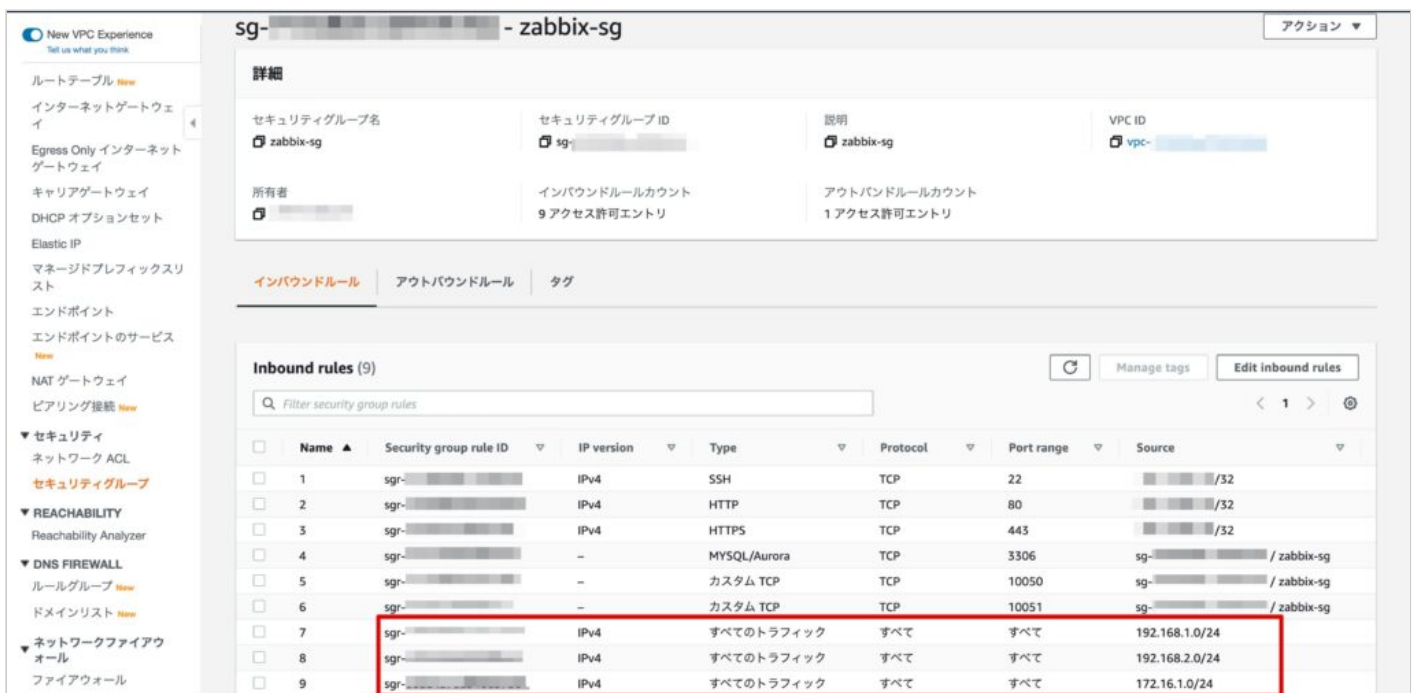
EC2に適用したセキュリティグループ(zabbix-sg)の「Edit inbound rules」をクリックします。



タイプは「全てのトラフィック」、ソースは「192.168.1.0/24,192.168.2.0/24,172.16.1.0/24」のルールを追加します。



ルールが追加されていることを確認します。



## CMLの構築

### CMLの設定

Server、IOSv(VPN-Router, Router1)、ExternalConnectorを配置し、下記の通りリンクを接続します。

The screenshot shows the Cisco Modeling Labs (CML) Workbench interface. At the top, there's a navigation bar with 'DASHBOARD', 'TOOLS', and 'ADMIN' tabs. Below it, a network topology diagram is displayed, showing a 'Server' connected to 'Router1' via 'Eth0' and 'G0/1'. 'Router1' is connected to 'VPN-Router' via 'G0/0' and 'G0/1'. 'VPN-Router' is connected to an 'ExternalConnector' via 'G0/0'. A vertical 'ADD NODES' button is on the right side of the diagram.

Below the diagram, there's a 'LAB INFO' section with tabs for 'PERMISSIONS', 'SIMULATE', 'NODES', 'DESIGN', 'LOGS', and 'LAB NOTES'. The 'NODES' tab is active, showing a table of nodes:

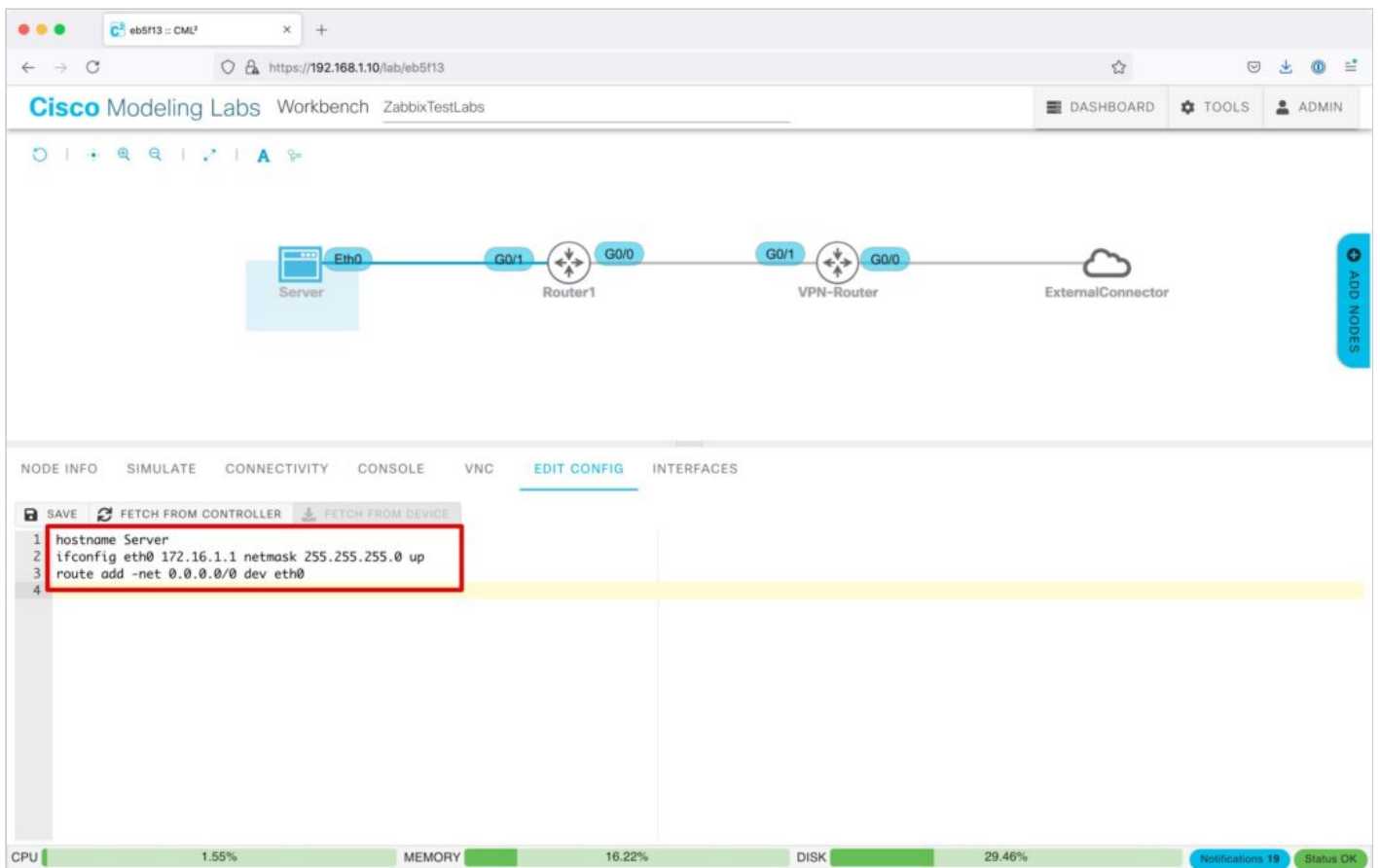
Node	State	Uptime	CPU
VPN-ROUTER	CREATED	00:00:00	0.00%
EXTERNALCONNECTOR	CREATED	00:00:00	0.00%
ROUTER1	CREATED	00:00:00	0.00%
SERVER	CREATED	00:00:00	0.00%

At the bottom, there's a status bar showing 'CPU' at 2.10%, 'MEMORY' at 16.22%, and 'DISK' at 29.46%. There are also buttons for 'Notifications 19' and 'Status OK'.

Serverは、「EDIT CONFIG」で下記の設定を行い起動します。

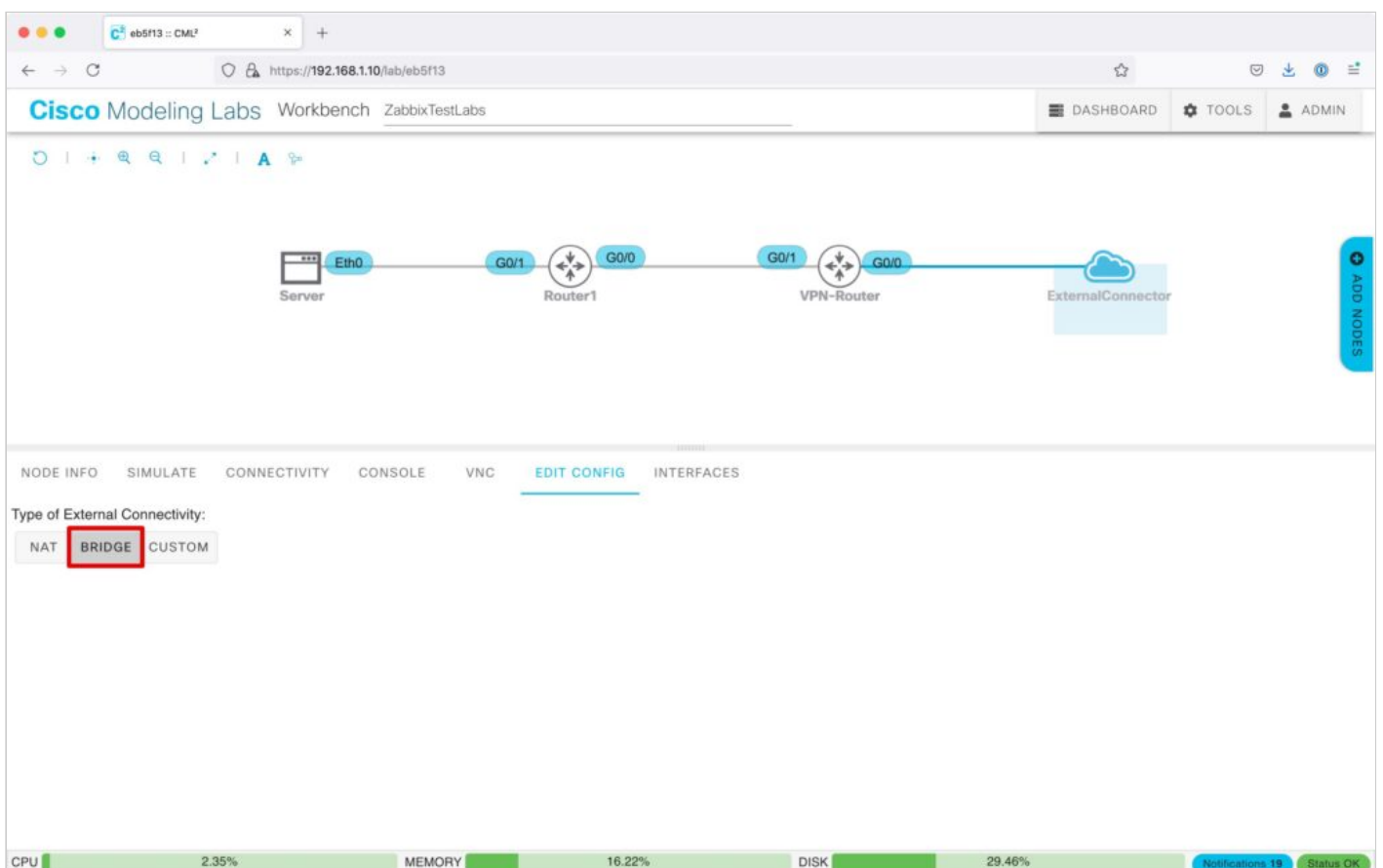
```
hostname Server
ifconfig eth0 172.16.1.1 netmask 255.255.255.0 up
route add -net 0.0.0.0/0 dev eth0
```





ExternalConnectorは、「EDIT CONFIG」で「BRIDGE」を選択します。

※CMLの外部ネットワーク接続の詳細は[こちら](#)で説明しています。





VPN-Routerを設定します。

\*の部分はダウンロードしたテンプレート通りです。

[VPN-Router]

```
crypto keyring keyring-vpn-*****
```

```
local-address 192.168.1.100
```

```
pre-shared-key address ***, ***, ***, *** key *****
```

```
crypto isakmp policy 200
```

```
encr aes
```

```
authentication pre-share
```

```
group 2
```

```
lifetime 28800
```

```
exit
```

```
crypto isakmp profile isakmp-vpn-*****
```

```
keyring keyring-vpn-*****
```

```
match identity address ***, ***, ***, *** 255.255.255.255
```

```
local-address XXX.XXX.XXX.XXX ※ここは自身のグローバルアドレスを指定
```

```
exit
```

```
crypto ipsec transform-set ipsec-prop-vpn-***** esp-aes 128 esp-sha-  
hmac
```

```
mode tunnel
```

```
exit
```

```
crypto ipsec profile ipsec-vpn-*****
```

```
set pfs group2
```

```
set security-association lifetime seconds 3600
```

```
set transform-set ipsec-prop-vpn-*****
```

```
exit
```

```
interface Tunnel1
```

```
ip address 169.254.173.150 255.255.255.252
```

```
ip virtual-reassembly
```

```
tunnel source 192.168.1.100
```

```
tunnel destination ***, ***, ***, ***
```

```
tunnel mode ipsec ipv4
```

```
tunnel protection ipsec profile ipsec-vpn-*****
```

```
ip tcp adjust-mss 1379
no shutdown
exit

interface GigabitEthernet0/0
ip address 192.168.1.100 255.255.255.0
no shutdown
exit

interface GigabitEthernet0/1
ip address 192.168.2.2 255.255.255.0
no shutdown
exit

ip route 0.0.0.0 0.0.0.0 192.168.1.1
ip route 10.0.0.0 255.255.255.0 Tunnel1
ip route 172.16.1.0 255.255.255.0 192.168.2.1
```

Tunnel1が作成され、リンクアップしていることを確認します。

```
show ip int brief
```

```
Router#show ip int brief
```

Interface	IP-Address	OK?	Method
GigabitEthernet0/0	192.168.1.100	YES	manual up
GigabitEthernet0/1	192.168.2.2	YES	manual up
Tunnel1	169.254.173.150	YES	manual up

AWSのVPN接続の「トンネル詳細」で、Tunnel1のステータスが「アップ」していることを確認します。

VPN 接続の作成 設定のダウンロード アクション

タグや属性によるフィルター, またはキーワードによる検索

Name	VPN ID	状態	仮想プライベートゲートウェイ	カスタマーゲートウェイ	カスタマーゲートウェイアドレス	内部 IP バージョン	タイプ
zabbix-test-vpn	vpn-	使用可能	vgw-	zabbix-test-vgw	cgw-	zabbix-test-cgw	IPv4

VPN 接続: vpn-

詳細 トネル詳細 静的ルート タグ

トンネルの状態

トンネル番号	外部 IP アドレス	内部 IPv4 CIDR	内部 IPv6 CIDR	ステータス	ステータスの最終変更日	詳細	証明
Tunnel 1		/30	-	アップ	UTC+9	-	-
Tunnel 2		/30	-	ダウン	UTC+9	-	-

利用可能なトンネルオプションとデフォルト値の詳細については、[こちら](#)を参照してください。

トンネル 1 のオプション

オプション	値	オプション	値
フェーズ 1 暗号化アルゴリズム	<default>	フェーズ 2 暗号化アルゴリズム	<default>
フェーズ 1 整合性アルゴリズム	<default>	フェーズ 2 整合性アルゴリズム	<default>
フェーズ 1 DH グループ番号	<default>	フェーズ 2 DH グループ番号	<default>
フェーズ 1 の有効期間	<default>	フェーズ 2 の有効期間	<default>
IKE バージョン	<default>	キー再生成マージン時間	<default>
キー再生成 Fuzz	<default>	再生ウィンドウサイズ	<default>
DPD タイムアウト	<default>	DPD タイムアウトアクション	<default>
スタートアップアクション	<default>		

## 疎通確認

ServerからEC2に向けてPingを実施し、疎通可能であることを確認します。

```
ping 10.0.0.100
```

Cisco Modeling Labs Workbench ZabbixTestLabs

DASHBOARD TOOLS ADMIN

Server Router1 VPN-Router ExternalConnector

NODE INFO SIMULATE CONNECTIVITY CONSOLE VNC EDIT CONFIG INTERFACES DOWNLOAD HISTORY SETTINGS

```

cisco@Server:~$ ping 10.0.0.100
PING 10.0.0.100 (10.0.0.100): 56 data bytes
64 bytes from 10.0.0.100: seq=0 ttl=252 time=20.336 ms
64 bytes from 10.0.0.100: seq=1 ttl=252 time=24.062 ms
64 bytes from 10.0.0.100: seq=2 ttl=252 time=31.690 ms
64 bytes from 10.0.0.100: seq=3 ttl=252 time=19.299 ms
64 bytes from 10.0.0.100: seq=4 ttl=252 time=25.597 ms
64 bytes from 10.0.0.100: seq=5 ttl=252 time=17.043 ms
64 bytes from 10.0.0.100: seq=6 ttl=252 time=20.525 ms
^C
--- 10.0.0.100 ping statistics ---
7 packets transmitted, 7 packets received, 0% packet loss
round-trip min/avg/max = 17.043/22.650/31.690 ms
cisco@Server:~$
cisco@Server:~$
cisco@Server:~$
cisco@Server:~$
cisco@Server:~$
cisco@Server:~$
cisco@Server:~$

```

CPU 21.88% MEMORY 26.74% DISK 29.63% Notifications 00 Status OK

EC2からServerに向けてPingを実施し、疎通可能であることを確認します。

```
ping 172.16.1.1
```

```
[[ec2-user@ip-10-0-0-100 ~]$  
[[ec2-user@ip-10-0-0-100 ~]$ ping 172.16.1.1  
PING 172.16.1.1 (172.16.1.1) 56(84) bytes of data.  
64 bytes from 172.16.1.1: icmp_seq=1 ttl=62 time=17.8 ms  
64 bytes from 172.16.1.1: icmp_seq=2 ttl=62 time=34.1 ms  
64 bytes from 172.16.1.1: icmp_seq=3 ttl=62 time=59.9 ms  
64 bytes from 172.16.1.1: icmp_seq=4 ttl=62 time=16.3 ms  
64 bytes from 172.16.1.1: icmp_seq=5 ttl=62 time=16.5 ms  
^C  
--- 172.16.1.1 ping statistics ---  
5 packets transmitted, 5 received, 0% packet loss, time 4005ms  
rtt min/avg/max/mdev = 16.338/28.986/59.963/16.879 ms  
[[ec2-user@ip-10-0-0-100 ~]$
```

これで、AWS上での監視サーバー(Zabbix)構築【5.AWSとCMLのVPN接続】の説明は完了です！