トランジットゲートウェイを使用したトランジット VPC の展開

トランジット ゲートウェイ ソリューションに関する情報

Amazon Virtual Private Cloud（Amazon VPC）を使用して、必要な数の仮想ネットワークを作成できます。AWS では、これらのネットワークを相互に接続したり、非 AWS インフラストラクチャ（オンプレミスのデータセンター、離れた場所にある本社、その他のオフィス）に接続したりするためのさまざまなオプションも提供しています。

トランジット VPC ソリューションを使用して Cisco Catalyst 8000V インスタンスを展開すると、Amazon VPC でハブアンドスポークトポロジを構築してエッジ接続を一元化できます。トランジット VPC では、VPC での共有サービスまたはパケットインスペクション/レプリケーションを導入できます。複数のアカウントにわたって機能し、AWS CloudFormation スタックを介して簡単に設定できます。ただし、このソリューションではトランジットゲートウェイではなく VPN ゲートウェイを使用するため、新しいスポークの追加にはある程度複雑な操作が伴います。

この制限を克服するため、トランジット ゲートウェイ ソリューションを使用して Cisco Catalyst 8000V トランジット VPC を展開できるようになりました。トランジットゲートウェイは、AWS クラウドとオンプレミスネットワークで VPC を相互接続するために AWS が提供する地域ネットワーク トランジット ハブ サービスです。トランジットゲートウェイを使用した Cisco Catalyst 8000V トランジット VPC ソリューションでは、スポーク側のトランジットゲートウェイを使用して、同じ地域内の全スポーク VPC 間の接続を可能にします。トランジットゲートウェイは、VPN 接続を使用してトランジット VPC の 2 つの Cisco Catalyst 8000V インスタンスに接続されます。Cisco Catalyst 8000V インスタンスは、さまざまなオンプレミス ブランチ ロケーションへの VPN 接続を提供します。

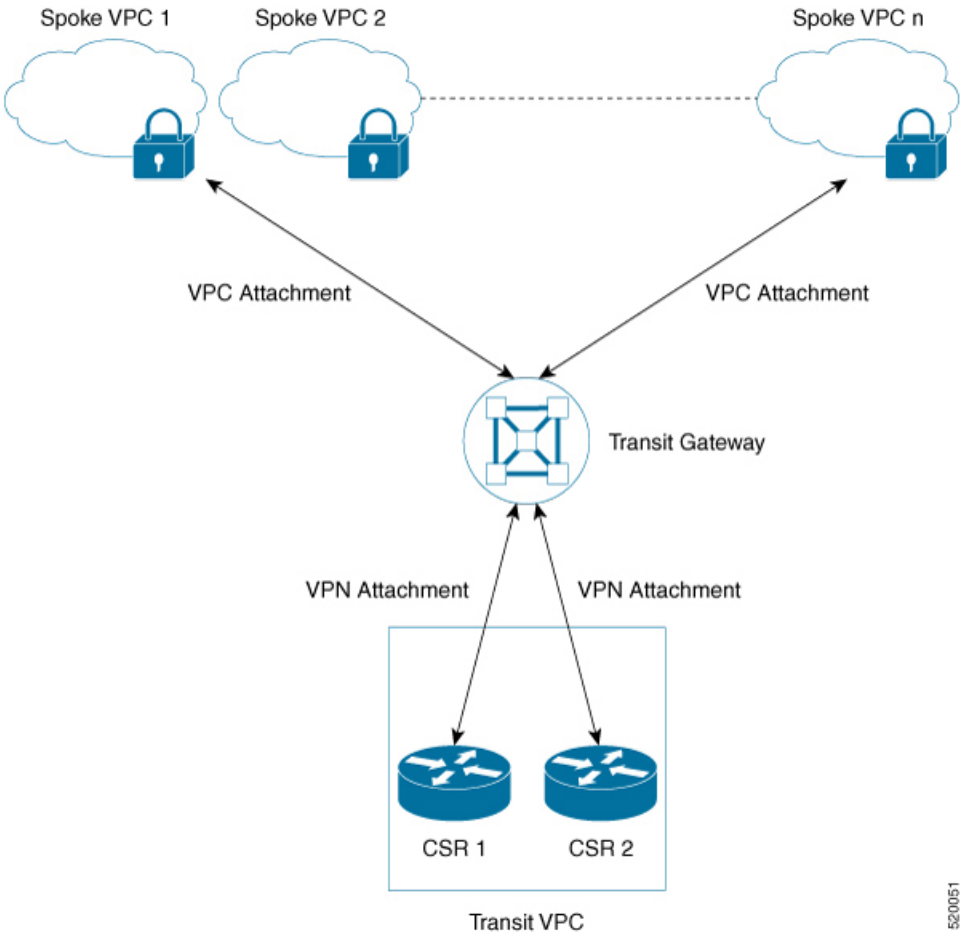
トランジット ゲートウェイ ソリューションを使用して AWS トランジット VPC を展開する方法を確認するには、この章で説明する設定手順を実行します。

トランジット VPC - トランジット ゲートウェイ コンポーネント

トランジット ゲートウェイ ソリューションには、スポーク間 VPC 接続を提供するためのハブとして機能するトランジットゲートウェイがあります。トランジット VPC は、スポーク VPC からリモートネットワークに流れるトラフィックの中央ハブとして機能するもう 1 つのコアコ

ンポーネントです。トランジット VPC は、VPN の終端とルーティングを可能にする 2 つの Cisco Catalyst 8000V インスタンスをホストします。

図 *1:* トランジット ゲートウェイ ソリューションのサンプルトポロジ



このソリューションでは、Solution Helper と Cisco Configurator という 2 つの AWS Lambda 関数を使用して、インスタンスとスポーク VPC 間の VPN 接続を自動的に設定します。

- **Solution Helper Lambda**：このコンポーネントは、cloudformation テンプレートを展開するとトリガーされます。このコンポーネントでは、トランジットゲートウェイ、Cisco Catalyst 8000V インスタンスとの VPN 接続、およびインスタンスとトランジットゲートウェイ間の VPN 接続が作成されます。その後、Lambda 関数は S3 SSE-KMS を使用して VPN 接続情報を Amazon S3 バケットに保存します。

- **Cisco Configurator Lambda**：S3 Put イベントによって Cisco Configurator Lambda 関数が呼び出されます。この関数により、VPN 接続情報が解析され、新しい VPN 接続を作成するために必要な設定ファイルが生成されます。Cisco Configurator Lambda は、SSH を使用して IOS 設定を Cisco Catalyst 8000V インスタンスにプッシュします。シスコの設定が Cisco Catalyst 8000V インスタンスに適用されると、即座に VPN トンネルが起動し、トランジットゲートウェイとの間にボーダー ゲートウェイ プロトコル（BGP）ネイバー関係が確立されます。

AWS トランジット ゲートウェイ ソリューションの利点

- トランジット ゲートウェイ ソリューションには、拡張性と復元力があります。
- トランジット ゲートウェイ ソリューションはマネージドサービスです。つまり、高可用性およびモニタリング機能が組み込まれており、CloudWatch などのメトリックを使用してソリューションを追跡できます。
- トランジット ゲートウェイ ソリューションを使用すると、ネットワークアーキテクチャの簡素化が可能になるため、運用コストの削減を実現できます。
- セキュリティを含めて、ソリューションを一元管理できます。

AWS トランジット ゲートウェイ ソリューションの前提条件

- Elastic IP、VPC、TGW、および VPN 接続に十分な制限が課されている必要があります。
- *cloudformation* サービスを管理する IAM 権限があることを確認します。

AWS トランジット ゲートウェイ ソリューションの制限事項

- 自動スケーリングは、このバージョンのソリューションではサポートされていません。
- このソリューションを展開した後、VPC 接続を使用して、スポーク VPC をトランジットゲートウェイに手動で追加する必要があります。

AWS トランジット ゲートウェイ ソリューションの設定

ステップ 1 Amazon Web Services Marketplace にログインします。

ステップ 2 **Cisco Catalyst 8000V – Transit Network VPC** テンプレートを検索して、このテンプレートを選択します。

ステップ 3 自分の所在地に該当する地域でテンプレートを起動します。[AWS Cloudformation Service] ページが表示されます。[Next] をクリックします。

ステップ 4 次の [Stack Details] を指定します。

パラメータ	説明
C8000V Throughput Requirements	Cisco Catalyst 8000V インスタンスに必要なスループット。この値により、起動するインスタンスタイプが決まります。デフォルト値は 2 x 500 Mbps です。
SSH Key to access C8000V	Cisco Catalyst 8000V インスタンスの起動後に、インスタンスへのセキュアな接続を可能にする公開/秘密キーペア。 公開/秘密キーペアを入力する必要があります。このキーペアは、AWS アカウントの作成時に、設定した地域で作成されます。
License Model	BYOL は、現在サポートされている唯一のライセンスモデルです。
Enable Termination Protection	Cisco Catalyst 8000V インスタンスの終了保護を有効にするには、このフィールドを有効にします。この機能により、偶発的な Cisco Catalyst 8000V の終了が防止されます。実稼働環境でこのフィールドを有効にすることを推奨します。デフォルトでは、このフィールドの値は [Yes] に設定されます。
Prefix for S3 Objects	Amazon S3 オブジェクトの作成時にプレフィックスとして使用する必要があるテキスト文字列。デフォルトの値は **vpnconfigs/** です。
Additional AWS Account ID	S3 バケットと AWS KMS カスタマーマスターキーへのアクセスを許可するトランジットネットワークに関連付けられた AWS アカウントのアカウント ID。 （注） このフィールドには、追加の AWS アカウント ID を 1 つだけ入力できます。複数の追加の AWS アカウントをトランジットネットワークに接続する場合は、追加のアカウントのアクセス許可を手動で設定する必要があります。
Transit VPC CIDR Block	トランジット VPC の CIDR ブロック。VPC とサブネット CIDR のアドレス範囲を変更して、ネットワー

パラメータ	説明
	クとのコリジョンを回避します。デフォルトの値は **100.64.127.224/27** です。
1st Subnet Network	AZ1 で作成されたトランジット VPC サブネットの CIDR ブロック。デフォルトの値は **100.64.127.224/28** です。
2nd Subnet Network	AZ2 で作成されたトランジット VPC サブネットの CIDR ブロック。デフォルトの値は **100.64.127.240/28** です。
Transit VPC BGP ASN	トランジット VPC の BGP 自律システム番号（ASN）。デフォルトの値は **64512** です。
Spoke VPC Tag Name	トランジット VPC に接続するスポーク VPC の識別に使用するタグ。
Preferred VPN Endpoint Tag Name	トランジット VPC Cisco Catalyst 8000V インスタンスを通過するトラフィックフローを制御する優先 Cisco Catalyst 8000V VPN エンドポイントを設定するために使用するタグ。たとえば、ステートフルオンプレミス ファイアウォールと統合する場合に使用します。
Optional AZ configuration 1st Subnet	Public Subnet1 の可用性ゾーン番号。
Optional AZ configuration 2nd Subnet	Public Subnet2 の可用性ゾーン番号。

ステップ 5 設定を確認して確定します。AWS Identity and Access Management（IAM）によってリソースが作成され、CAPABILITY_AUTO_EXPAND 機能が必要になる可能性があることを承認するには、このチェックボックスをオンにします。

ステップ 6 [Create] をクリックして、スタックを展開します。
展開が成功すると、AWS Cloud Formation コンソールの [Status] 列に [CREATE_COMPLETE] と表示されます。

設定例

次に、トランジット ゲートウェイ ソリューションを使用して AWS トランジット VPC を展開する設定例を示します。

```
ip-100-64-127-234#sh run
Building configuration...

Current configuration : 7284 bytes
```

```
!
! Last configuration change at 14:10:57 UTC Thu Oct 10 2020
!
version 17.4
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
platform qfp utilization monitor load 80
no platform punt-keepalive disable-kernel-core
platform console virtual
!
hostname ip-100-64-127-234
!
boot-start-marker
boot-end-marker
!
!
vrf definition GS
 rd 100:100
 !
 address-family ipv4
 exit-address-family
!
logging persistent size 1000000 filesize 8192 immediate
!
no aaa new-model
!
ip vrf vpn-0f56b2afc60b1d492
 rd 64525:1
 route-target export 64525:0
 route-target import 64525:0
!
ip vrf vpn0
 rd 64525:0
!
ip admission watch-list expiry-time 0
!
subscriber templating
!
multilink bundle-name authenticated
!
crypto pki trustpoint TP-self-signed-572041569
 enrollment selfsigned
subject-name cn=IOS-Self-Signed-Certificate-572041569
 revocation-check none
 rsakeypair TP-self-signed-572041569
!
!
crypto pki certificate chain TP-self-signed-572041569
 certificate self-signed 01
  3082032E 30820216 A0030201 02020101 300D0609 2A864886 F70D0101 05050030
  30312E30 2C060355 04031325 494F532D 53656C66 2D536967 6E65642D 43657274
  69666963 6174652D 35373230 34313536 39301E17 0D313931 30313031 34303631
  355A170D 33303031 30313030 30303030 5A303031 2E302C06 03550403 1325494F
  532D5365 6C662D53 69676E65 642D4365 72746966 69636174 652D3537 32303431
  35363930 82012230 0D06092A 864886F7 0D010101 05000382 010F0030 82010A02
  82010100 A974EDB7 292BBB6A 09026F6A 381F7852 714775E3 E25F1F89 CED40FCB
  F45204F9 2F2F5FEE C46A9D16 A8D7307A C5433234 10D3F709 B4B18B3D 009B4A7A
  85980EEB 1282D1F7 C3CD4429 16042D4D 544315F4 E3ABA673 21E66C52 187AD1E6
  6B21F98A F0537D0A 8171618E 6CDF3B70 E2C8B553 8096C2D6 B4CD1AE4 B6DFD615
  844924B8 83DBE166 3CBC90F1 889CB00F 1644ECCE F2E70D81 CA35B555 D9757BE4
  34440FD9 D15580FA C50181CD D646AB6C 22F707A7 1D9F98CA 19897AF4 7488762B
  35ECA78F D2B249C7 8079255F 72BE5CF8 214B5135 E97B1104 A9CB449E A4A1D996
  9B99EC0E 18EF94FE FE73706A BF417262 12771D33 FF61A325 4479CAFB 10D0EEAA
```

```
  810E3437 02030100 01A35330 51300F06 03551D13 0101FF04 05300301 01FF301F
  0603551D 23041830 16801476 E85FEE9B EAE114A4 74C542FD E923856D 6F17F830
  1D060355 1D0E0416 041476E8 5FEE9BEA E114A474 C542FDE9 23856D6F 17F8300D
  06092A86 4886F70D 01010505 00038201 010043A6 03287F7E 1F13A7D4 26D661FE
  D11FED41 FE195D3E 6ADEA111 267C534B 266F587A 6A2F395D C50F5894 4C01F62B
  A179B852 F5F8ED62 DFF35587 3CFF352C 523F8D3D 8A786E61 A73EA8BB C8FC0A8D
  C2F0C260 0BB25D28 01B26B2B 27D71A31 2CE81DA5 6296D4AA 756A6658 0ADB89FB
  52BE1E9F A8BF17AA B2A0379A 1921AF64 834455CF B6307205 CE12C83A 2D29AEF2
  D79B79F7 9701F86E EB51B8E2 95BA7D5A C67A05F8 2AA7A8E0 3626D155 FC2D79EC
  9506D897 D79B8E65 A1D89F8A 6EC21FD1 15BFBD79 8A6FEB77 15C10DEE 0A50A7A5
  C8109573 9C58A869 D2740BC4 61D953F2 7AA92870 69BF035C 08DA0EFB B4AB9AC1
  BD4DB053 66ADD9E3 B5957D2B 8E467A91 258A
   quit
!
license udi pid CSR1000V sn 9YGGWBVUY3N
no license smart enable
diagnostic bootup level minimal
!
spanning-tree extend system-id
!
username ec2-user privilege 15 secret 5 $1$Gf9p$OfANl/ujuCIvpunuRDwKi1
username automate privilege 15 secret 8
$8$g62y2e1pz0O4/n$M8DmVAM/G9yySvjbB1I2tBJAW4IWZRIc44Icent4bps
!
redundancy
!
crypto keyring keyring-vpn-0f56b2afc60b1d492-2
  local-address GigabitEthernet1
  pre-shared-key address 52.54.79.47 key lhvPlpTYxUTno.lNTbR25F9743HEguaH
crypto keyring keyring-vpn-0f56b2afc60b1d492-1
  local-address GigabitEthernet1
  pre-shared-key address 52.44.80.94 key Qq4fLolOMf1iW3d7gJhtzF8h8Tu3IlNT
!
crypto isakmp policy 200
 encr aes
 authentication pre-share
 group 2
 lifetime 28800
crypto isakmp keepalive 10 10 periodic
crypto isakmp profile isakmp-vpn-0f56b2afc60b1d492-1
   keyring keyring-vpn-0f56b2afc60b1d492-1
   match identity address 52.44.80.94 255.255.255.255
   local-address GigabitEthernet1
   rekey
crypto isakmp profile isakmp-vpn-0f56b2afc60b1d492-2
   keyring keyring-vpn-0f56b2afc60b1d492-2
   match identity address 52.54.79.47 255.255.255.255
   local-address GigabitEthernet1
   rekey
!
crypto ipsec security-association replay window-size 1024
!
crypto ipsec transform-set ipsec-prop-vpn-aws esp-aes esp-sha-hmac
 mode tunnel
crypto ipsec df-bit clear
no crypto ipsec nat-transparency udp-encapsulation
!
crypto ipsec profile ipsec-vpn-aws
 set transform-set ipsec-prop-vpn-aws
 set pfs group2
!
interface Tunnel1
 description vpn-0f56b2afc60b1d492 from TGW to cgw-00d8fbb76cc59295e for account
902347396780
```

```
 ip vrf forwarding vpn-0f56b2afc60b1d492
 ip address 169.254.185.70 255.255.255.252
 ip tcp adjust-mss 1387
 tunnel source GigabitEthernet1
 tunnel mode ipsec ipv4
 tunnel destination 52.44.80.94
 tunnel protection ipsec profile ipsec-vpn-aws
 ip virtual-reassembly
!
interface Tunnel2
 description vpn-0f56b2afc60b1d492 from TGW to cgw-00d8fbb76cc59295e for account
902347396780
 ip vrf forwarding vpn-0f56b2afc60b1d492
 ip address 169.254.232.90 255.255.255.252
 ip tcp adjust-mss 1387
 tunnel source GigabitEthernet1
 tunnel mode ipsec ipv4
 tunnel destination 52.54.79.47
 tunnel protection ipsec profile ipsec-vpn-aws
 ip virtual-reassembly
!
interface VirtualPortGroup0
 vrf forwarding GS
 ip address 192.168.35.101 255.255.255.0
 ip nat inside
 no mop enabled
 no mop sysid
!
interface GigabitEthernet1
 ip address 100.64.127.234 255.255.255.240
 ip nat outside
 negotiation auto
 no mop enabled
 no mop sysid
!
router bgp 64525
 bgp log-neighbor-changes
 !
 address-family ipv4 vrf vpn-0f56b2afc60b1d492
  neighbor 169.254.185.69 remote-as 64526
  neighbor 169.254.185.69 timers 10 30 30
  neighbor 169.254.185.69 activate
  neighbor 169.254.185.69 next-hop-self
  neighbor 169.254.185.69 default-originate
  neighbor 169.254.185.69 as-override
  neighbor 169.254.185.69 soft-reconfiguration inbound
  neighbor 169.254.232.89 remote-as 64526
  neighbor 169.254.232.89 timers 10 30 30
  neighbor 169.254.232.89 activate
  neighbor 169.254.232.89 next-hop-self
  neighbor 169.254.232.89 default-originate
  neighbor 169.254.232.89 as-override
  neighbor 169.254.232.89 soft-reconfiguration inbound
 exit-address-family
!
iox
ip nat inside source list GS_NAT_ACL interface GigabitEthernet1 vrf GS overload
ip forward-protocol nd
ip tcp window-size 8192
ip http server
ip http authentication local
ip http secure-server
ip route 0.0.0.0 0.0.0.0 GigabitEthernet1 100.64.127.225
ip route vrf GS 0.0.0.0 0.0.0.0 GigabitEthernet1 100.64.127.225 global
```

```
!
ip ssh rsa keypair-name ssh-key
ip ssh version 2
ip ssh pubkey-chain
  username ec2-user
   key-hash ssh-rsa F1B0DF92FC2E25F7D98A01B99FCE5F13 ec2-user
  username automate
   key-hash ssh-rsa ED4B0757CE2AC22C89B28BE55EDE7691
ip ssh server algorithm authentication publickey
ip scp server enable
!
ip access-list standard GS_NAT_ACL
 permit 192.168.35.0 0.0.0.255
!
control-plane
!
line con 0
 stopbits 1
line vty 0 4
 login local
 transport input ssh
!
app-hosting appid guestshell
 app-vnic gateway1 virtualportgroup 0 guest-interface 0
  guest-ipaddress 192.168.35.102 netmask 255.255.255.0
 app-default-gateway 192.168.35.101 guest-interface 0
 name-server0 8.8.8.8
end
```

AWS Transit Gateway ソリューションの削除

AWS Transit Gateway ソリューションを削除するには、次の手順を実行します。

ステップ 1 [CloudFormation] ページに移動します。

ステップ 2 削除するスタックをクリックし、[Delete] をクリックします。

削除操作が開始されます。ただし、一部のリソースは手動で削除する必要があるため、[Deletion Failed] ステータスが表示される場合があります。CloudFormation の外部で作成されたリソースは、[Delete] をクリックしても削除されません。このシナリオでは、次のようなリソースを手動で削除する必要があります。

- サイト間 VPN 接続の削除（c8000v-tgw から tgw-xxxxxx を削除）
- カスタマーゲートウェイの削除（Tranist VPC エンドポイント 1 および 2）
- トランジットゲートウェイ接続の削除
- トランジットゲートウェイの削除（c8000v-tgw）

（注） それでもソリューションが正常に削除されない場合は、Cisco Catalyst 8000V インスタンスで終了保護が有効になっているかどうかを確認してください。終了保護が有効になっている場合は、無効にしてから、Transit Gateway ソリューションを削除してみてください。

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。