Transit Network VPC  
(Cisco CSR)

AWS Implementation Guide

Steve Morad
Bryan Miller
Sidhartha Chauhan

July 2016

Last updated: December 2017 (see revisions)
# Contents

Overview ........................................................................................................................................... 3
Cost and Licenses ................................................................................................................................. 4
Architecture Overview ......................................................................................................................... 5
Solution Features ................................................................................................................................. 7
Transit VPC Components .................................................................................................................... 7
AWS CloudFormation Templates .......................................................................................................... 9
Automated Deployment ......................................................................................................................... 10
  Prerequisites ....................................................................................................................................... 10
  What We’ll Cover ................................................................................................................................. 10
Step 1. Accept the Cisco Software Terms ............................................................................................. 11
Step 2. Launch the Stack ....................................................................................................................... 11
Step 3. Tag the Spoke VPCs ................................................................................................................ 15
Step 4. Connect an Additional AWS Account (Optional) ................................................................. 17
Security .................................................................................................................................................. 19
  Security Groups ................................................................................................................................. 19
  Additional Security Settings ............................................................................................................... 19
Amazon CloudWatch ............................................................................................................................ 20
Testing ................................................................................................................................................... 21
  Transit VPC Test with Tsunami UDP ................................................................................................. 21
  Spoke VPC Templates ....................................................................................................................... 22
Additional Resources ........................................................................................................................... 22
Appendix A: Component Details .......................................................................................................... 23
  VGW Poller .......................................................................................................................................... 23
  Cisco Configurator ............................................................................................................................. 24
  BGP and Failover ............................................................................................................................... 24
Appendix B: Existing VPC Template .................................................................................................... 26
Appendix C: Configuring Permissions for Additional AWS Accounts ................................................. 28
Appendix D: Integration with AWS Direct Connect .............................................................................. 29
Appendix E: Collection of Anonymous Data ......................................................................................... 31
Send Us Feedback ................................................................................................................................ 31
Document Revisions ............................................................................................................................ 32
About This Guide

This implementation guide discusses architectural considerations and configuration steps for deploying a transit VPC on the Amazon Web Services (AWS) Cloud. It includes links to AWS CloudFormation templates that launch, configure, and run the AWS compute, network, storage, and other services required to deploy this solution on AWS, using AWS best practices for security and availability.

The guide is intended for IT infrastructure architects, administrators, and DevOps professionals who have practical experience architecting on the AWS Cloud.

Overview

Amazon Virtual Private Cloud (Amazon VPC) provides customers with the ability to create as many virtual networks as they need, as well as different options for connecting those networks to each other and to non-AWS infrastructure. One common strategy for connecting multiple, geographically disperse VPCs and remote networks is to create a transit VPC that serves as a global network transit center. A transit VPC simplifies network management and minimizes the number of connections required to connect multiple VPCs and remote networks. This design can save time and effort and also reduce costs, as it is implemented virtually without the traditional expense of establishing a physical presence in a colocation transit hub or deploying physical network gear.

This guide provides infrastructure and configuration information for planning and deploying a transit VPC that assumes a typical hub-and-spoke network topology, as depicted in the diagram to the left. In this design, remote VPCs access each other and remote networks through the central, transit VPC.

The AWS Cloud provides a suite of infrastructure services that enable you to deploy a transit VPC solution in a highly available, fault-tolerant, and affordable way. By integrating Cisco Cloud Services Router (CSR) with the AWS Cloud, you can take advantage of the functionality of enterprise-class

networking services and VPN along with the flexibility and security of AWS.

The information in this guide assumes basic knowledge of highly available remote-network connectivity, IPsec VPNs, network addressing, subnetting, and routing. The following sections do not include general installation or configuration tasks for Cisco CSR. For additional general guidance, best practices, and licensing details consult the Cisco product documentation.

**Cost and Licenses**

You are responsible for the cost of the AWS services used while running this reference deployment. You are also responsible for the Cisco CSR licenses, which you can either purchase beforehand or request from the AWS Marketplace, depending on the deployment model you choose: Bring Your Own License (BYOL) or License Included.

As of the date of publication, the cost for running a transit VPC with this solution’s default settings in US East (N. Virginia) is as shown in the table below.

<table>
<thead>
<tr>
<th>Transit VPC Deployment Size</th>
<th>BYOL Cost/Hour</th>
<th>License Included Cost/Hour</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 x 500 Mbps</td>
<td>$0.21</td>
<td>$4.21</td>
</tr>
<tr>
<td>2 x 1 Gbps</td>
<td>$0.84</td>
<td>$6.22</td>
</tr>
<tr>
<td>2 x 2 Gbps</td>
<td>$0.84</td>
<td>$7.68</td>
</tr>
<tr>
<td>2 x 4.5 Gbps</td>
<td>$1.676</td>
<td>$10.736</td>
</tr>
</tbody>
</table>

Each spoke VPC connected to the transit network costs an additional $0.10/hour, plus network transit costs. Additionally, the solution creates a unique AWS Key Management Service (KMS) customer master key (CMK) for protecting network configuration information, which costs $1/month. Prices are subject to change. For full details, see the pricing webpage for each AWS service you will be using in this solution.

---

2 See the **Prerequisites** section for detailed information.
Architecture Overview

Deploying this solution with the default parameters builds the following environment in the AWS Cloud.

This highly available design deploys two VPN appliances (Cisco CSR 1000v instances) into separate Availability Zones of a dedicated transit VPC. Customers can choose to automatically create a new VPC or to use an existing VPC as the transit hub (see AWS CloudFormation Templates for details). Spoke VPCs are connected to the transit network through dynamically routed VPN connections between their virtual private gateways (VGWs) and the CSR instances. This design uses VPN connections to enable routing between any connected network, including external networks or spoke VPCs in other AWS Regions. VPN connections also allow spoke VPC resources to leverage VGW capabilities for routing and failover in order to maintain highly available network
connections to the transit VPC instances. Remote networks also connect to the transit VPC using redundant, dynamically routed VPN connections between their customer gateways and the CSR instances. This design supports dynamic routing protocols, which customers can use to automatically route traffic around potential network failures as well as to propagate network routes to remote networks.

Note that all communication with the CSR instances, including the VPN connections between corporate data centers or other provider networks and the transit VPC, uses the transit VPC Internet gateway and the instances’ Elastic IP addresses. Each CSR instance has an associated Amazon CloudWatch alarm that enables automatic recovery of the instance if the underlying EC2 hardware fails.

Along with providing direct network routing between VPCs and on-premises networks, this design also enables the transit VPC to implement more complex routing rules, such as network address translation (NAT) between overlapping network ranges, or to add additional network-level packet filtering or inspection. Although it is possible, we recommend that you consider implementing non-overlapping network ranges for your private networks to simplify the ability to route between remote networks. Although a transit network can be an excellent place to implement NAT rules to compensate for overlapping networks, this adds additional complexity to the network design.

The AWS-to-AWS VPN connectivity in this design relies on software VPN appliances running on the Cisco CSR 1000v instances. It also relies on the capabilities of the hardware VPN device deployed on premises that connects to the software VPN appliance. While every effort has been made to come up with a solid configuration, each customer should verify the configuration and adapt it to their specific needs.

Note: We do not recommend adding other resources in the transit VPC because the primary function of this VPC is to provide a transit overlay network connecting spoke networks. If you deploy resources into the transit VPC, it will require extra steps to connect those resources to the transit network and will not leverage the design’s highly available VGW connections.

---

Solution Features
The automated transit VPC solution provides the following features:

- **AWS network connectivity**: You can connect any spoke VPCs that you wish—within the same AWS Region, across AWS Regions, and even from additional AWS accounts.

- **Remote network connectivity**: You can connect to your own data centers, other colocation providers, Managed Service Providers (MSPs), or even other cloud providers.

- **Automated configuration**: This solution leverages AWS Lambda to automatically configure VPN connections for the spoke VPCs you want to add to the transit network. Note that manual configuration of the CSR instances is required to connect remote networks directly to the transit VPC.

- **Additional functionality**: You can customize this solution to provide additional network features:
  - IDS/IPS, Next-Generation Firewall
  - NAT to bridge overlapping networks
  - NAT to access VPC-E, EFS, or internal Amazon Route 53

Transit VPC Components
As described in the Architecture Overview, at the core of the design is a VPC (the transit VPC) that acts as a central hub for traffic flowing to any other destination, whether it be another VPC or a remote network. The transit VPC hosts two CSR instances that allow for VPN termination and routing. This solution uses two AWS Lambda functions, the VGW Poller and the Cisco Configurator, to automatically configure VPN connections between these instances and spoke VPCs.

The following diagram gives an overview of the various components and steps involved in connecting spoke VPCs to the transit network. (See Appendix A for detailed information.)
Figure 2: Connecting spoke VPCs to the transit network

The process for adding a new spoke VPC is as follows:

1. Every (1) minute, an Amazon CloudWatch event invokes the VGW Poller Lambda function, which iterates through each AWS Region of a customer’s account, searching for appropriately tagged spoke VGWs (default tag key `transitvpc:spoke`, default tag value `true`) that do not have existing transit VPC VPN connections.

2. When the VGW Poller identifies an applicable spoke VGW, it creates the corresponding customer gateways (if required) and VPN connections to each CSR, and then saves this connection information to an Amazon S3 bucket using S3 SSE-KMS. All data in the S3 bucket is encrypted using a solution-specific AWS KMS managed customer master key (CMK).

3. The S3 Put event invokes the Cisco Configurator Lambda function, which parses the VPN connection information and generates the necessary config files to create new VPN connections.

4. The Cisco Configurator pushes the configuration to the CSR instances using SSH.

5. As soon as the Cisco configuration is applied onto the CSR instances, the VPN tunnels come up and Border Gateway Protocol (BGP) neighbor relationships are established to the spoke VPCs.

The process for removing an existing spoke VPC is as follows:
1. Every (1) minute, an Amazon CloudWatch event invokes the VGW Poller Lambda function, which iterates through each AWS Region of a customer’s account, searching for spoke VGWs to remove. Spoke VGWs to remove have existing transit VPC connections, but either lack the spoke VPC tag (because it has been removed) or the spoke VPC tag value has been changed from the expected value.

2. When the VGW Poller identifies an applicable spoke VGW for removal, it deletes existing VPN connections to each CSR, saves the existing connection information to an Amazon S3 bucket using S3 SSE-KMS, and deletes any corresponding customer gateways (if no longer required). All data in the S3 bucket is encrypted using a solution-specific AWS KMS managed customer master key (CMK).

3. The S3 Put event invokes the Cisco Configurator Lambda function, which parses the VPN connection information and generates the necessary config files to remove the VPN connections.

4. The Cisco Configurator pushes the connection-removal configuration to the CSR instances using SSH.

AWS CloudFormation Templates

This solution uses AWS CloudFormation to bootstrap AWS infrastructure and automate the deployment of a transit VPC on the AWS Cloud. It includes the following CloudFormation templates, which you can download before deployment:

- **transit-vpc-primary-account.template**: Use this template to launch the transit VPC and all associated components, as described in the previous section. The default configuration offers four deployment size options based on common network bandwidth requirements, but you can also customize the template based on your specific network needs.

- **transit-vpc-primary-account-existing-vpc.template**: Use this template to use an existing VPC as the transit VPC. It launches two CSR instances into your existing VPC and configures all other solution components. For detailed requirements and parameters, see Appendix B.

- **transit-vpc-second-account.template**: Use this template to connect additional AWS accounts to the transit network. It launches the VGW Poller Lambda function, which automatically searches and sets up VPN connections for appropriately tagged spoke VPCs from that account. The Automated Deployment section contains instructions for adding one other AWS account. Note that manual configuration steps are required to add further AWS accounts to the network. See Appendix C for detailed instructions.
Automated Deployment

Before you launch the automated deployment, please review the architecture, configuration, network security, and other considerations discussed in this guide. Follow the step-by-step instructions in this section to configure and deploy a transit VPC into your account.

**Time to deploy:** Approximately five (5) minutes

**Prerequisites**

You must decide on the licensing model for the Cisco Cloud Services Router (CSR) before launching the AWS CloudFormation template. If you choose to bring your own license (the **BYOL** model), you will need to sign up for the BYOL version of Cisco Cloud Services Router (CSR) 1000V - BYOL for Maximum Performance, located here: [https://aws.amazon.com/marketplace/pp/B00NF48FI2](https://aws.amazon.com/marketplace/pp/B00NF48FI2)

Otherwise, use the **License Included** model and sign up for the License Included version of Cisco Cloud Services Router (CSR) 1000V - AX Pkg. Max Performance, located here: [https://aws.amazon.com/marketplace/pp/B00OCG4OAA](https://aws.amazon.com/marketplace/pp/B00OCG4OAA)

**What We’ll Cover**

The procedure for deploying this architecture on AWS consists of the following steps. For detailed instructions, follow the links for each step.

**Step 1. Accept the Cisco Software Terms**

- Choose the proper Cisco Cloud Services Router (CSR) from the AWS Marketplace and click through the agreement.

**Step 2. Launch the Stack**

- Launch the AWS CloudFormation template into your AWS account.
- Enter values for required parameters: **Stack name, SSH Key to access CSR, License Model**
- Review the other template parameters, and adjust if necessary.

**Step 3. Tag the Spoke VPCs**

- Tag the VGW of each VPC that you want to add to the transit network.

**Step 4. Connect an Additional AWS Account (Optional)**

- Launch the `transit-vpc-second-account` AWS CloudFormation template in another AWS account that contains spoke VPCs.
Step 1. Accept the Cisco Software Terms

Before launching the Cisco CSR AMI from AWS Marketplace, you are required to accept the Cisco Software terms.

1. In AWS Marketplace, go to the page of the appropriate Cisco Software package based on your preference:
   - BYOL version of Cisco Cloud Services Router (CSR) 1000V: https://aws.amazon.com/marketplace/pp/B00NF48FI2
   - License Included version of Cisco Cloud Services Router (CSR) 1000V: https://aws.amazon.com/marketplace/pp/B00OCG4OAA

   Review the Pricing Details for the AWS Region where you place to deploy the transit VPC, and then choose Continue.

2. Click Manual Launch. Be sure to read the End User License Agreement (EULA) and AWS Customer Agreement. If you accept these terms, choose Accept Software Terms.

   You should see a subscription confirmation message. You will also receive a confirmation email.

Step 2. Launch the Stack

This automated AWS CloudFormation template deploys a transit VPC on the AWS Cloud. Please make sure that you’ve selected the proper Cisco CSR in AWS Marketplace before launching the stack and have accepted the software terms. If you plan to add VPCs from an additional AWS account to the transit network, make sure to note its account ID before you launch this stack.

Note: You are responsible for the cost of the AWS services used while running this solution. See the Cost and Licenses section for more details. For full details, see the pricing webpage for each AWS service you will be using in this solution.

1. Log in to the AWS Management Console and click the button to the right to launch the transit-vpc-primary-account AWS CloudFormation template.
You can also download the template as a starting point for your own implementation.

2. The template is launched in the US East (N. Virginia) Region by default. To launch the transit VPC in a different AWS Region, use the region selector in the console navigation bar.

   Note: This solution uses the AWS Lambda service, which is currently available in specific AWS Regions only. Therefore, you must launch this solution an AWS Region where Lambda is available.  

3. On the Select Template page, verify that you selected the correct template and choose Next.

4. On the Specify Details page, assign a name to your transit VPC in the Stack name field.

5. Under Parameters, review the parameters for the template and modify them as necessary. This solution uses the following default values.

   Note: If you use the transit-vpc-primary-account-existing-vpc AWS CloudFormation template, review the template requirements and parameters in Appendix B.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSR Throughput Requirements</td>
<td>2x500Mbps</td>
<td>A drop-down box with four options: 2x500Mbps (c4.large), 2x1Gbps (c3.2xlarge), 2x2Gbps (c4.2xlarge), and 2x4.5Gbps (c4.4xlarge)</td>
</tr>
<tr>
<td>SSH Key to access CSR</td>
<td>&lt;Requires input&gt;</td>
<td>Public/private key pair, which allows you to connect securely to your instance after it launches. When you created an AWS account, this is the key pair you created in your preferred AWS Region.</td>
</tr>
<tr>
<td>License Model</td>
<td>LicenseIncluded</td>
<td>A drop-down box with two choices: LicenseIncluded and BYOL</td>
</tr>
<tr>
<td>Enable Termination Protection</td>
<td>Yes</td>
<td>Allows termination protection to be enabled on the CSR instances to help prevent accidental CSR termination (we recommend termination protection for production deployments).</td>
</tr>
<tr>
<td>Prefix for S3 Objects</td>
<td>vpnconfigs/</td>
<td>The text string you want to use to specify the name of the prefix for the Amazon S3 objects that are created. You must end the string with a forward slash (/).</td>
</tr>
</tbody>
</table>

4 For the most current AWS Lambda availability by region, see https://aws.amazon.com/about-aws/global-infrastructure/regional-product-services/
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Additional Account ID</td>
<td>AWS</td>
<td>Enter the account ID of one additional AWS account that you want to connect to the transit network. This is necessary to grant that account access to the S3 bucket and the AWS KMS customer master key.</td>
</tr>
<tr>
<td>Note:</td>
<td></td>
<td>Enter one additional account ID in this field. If you want to connect more than one additional AWS account to the transit network, you must manually configure permissions for the remaining accounts. See Appendix C for detailed instructions.</td>
</tr>
<tr>
<td>Transit VPC CIDR Block</td>
<td>100.64.127.224/27</td>
<td>CIDR block for the transit VPC. You can modify the VPC and subnet CIDR address ranges to avoid collisions with your network.</td>
</tr>
<tr>
<td>1st Subnet Network</td>
<td>100.64.127.224/28</td>
<td>CIDR block for the transit VPC subnet created in AZ1</td>
</tr>
<tr>
<td>2nd Subnet Network</td>
<td>100.64.127.240/28</td>
<td>CIDR block for the transit VPC subnet created in AZ2</td>
</tr>
<tr>
<td>Transit VPC BGP ASN</td>
<td>64512</td>
<td>BGP ASN to use for the transit VPC</td>
</tr>
<tr>
<td>Spoke VPC Tag Name</td>
<td>transitvpc:spoke</td>
<td>Tag name (key) to identify spoke VPCs to connect to the transit VPC. You can modify the tag name to align with any existing naming conventions. Use a name that is not likely to be used on VGWs for a different purpose to ensure you do not mistakenly add a VPC to the transit network.</td>
</tr>
<tr>
<td>Spoke VPC Tag Value</td>
<td>true</td>
<td>Tag value to determine which spoke VPCs to connect to the transit VPC. You can modify the tag value to align with any existing naming conventions. Be sure to use a value that is easy to understand and implement consistently.</td>
</tr>
<tr>
<td>Preferred VPN Endpoint Tag Name</td>
<td>transitvpc:prefer red-path</td>
<td>Tag name (key) to identify a preferred CSR instance for defining active/passive paths through the transit network. You can modify the tag name to align with any existing naming conventions. For more information about preferred path values and implementation details, see Step 3.</td>
</tr>
<tr>
<td>1st Subnet AZ #</td>
<td>0</td>
<td>The Availability Zone number for the first public subnet</td>
</tr>
<tr>
<td>Note:</td>
<td></td>
<td>We recommend you keep the default setting for this parameter. However, in some cases, you might have to select a different AZ. For example, change this parameter if the default AZ does not support a service or the instance type required for the solution.</td>
</tr>
<tr>
<td>2nd Subnet AZ #</td>
<td>1</td>
<td>The Availability Zone number for the second public subnet</td>
</tr>
<tr>
<td>Parameter</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CSR1</td>
<td></td>
<td>Public IP address for CSR 1, which is necessary for manually configuring CSR instances or connecting the transit VPC to remote networks</td>
</tr>
<tr>
<td>CSR2</td>
<td></td>
<td>Public IP address for CSR 2, which is necessary for manually configuring CSR instances or connecting the transit VPC to remote networks</td>
</tr>
<tr>
<td>ConfigS3Bucket</td>
<td></td>
<td>S3 bucket created by this template, and used to store VPN connection information</td>
</tr>
<tr>
<td>BucketPrefix</td>
<td></td>
<td>Should match the string you entered in the template’s Prefix for S3 Objects value</td>
</tr>
<tr>
<td>SpokeVPCTag</td>
<td></td>
<td>Should match the spoke VPC tag name you entered in the template</td>
</tr>
</tbody>
</table>

**Note:** We recommend you keep the default setting for this parameter. However, in some cases, you might have to select a different AZ. For example, change this parameter if the default AZ does not support a service or the instance type required for the solution.

6. Choose **Next**.

7. On the **Options** page, you can specify tags (key-value pairs) for resources in your stack and set additional options, and then choose **Next**.

8. On the **Review** page, review and confirm the settings. Be sure to check the box acknowledging that the template will create AWS Identity and Access Management (IAM) resources.

9. Choose **Create** to deploy the stack.

   You can view the status of the stack in the AWS CloudFormation console in the **Status** column. You should see a status of **CREATE_COMPLETE** in roughly five (5) minutes.

**Note:** This solution relies on stateful Cisco CSR instances to maintain long-running VPN connections to support the transit network. The solution does not support updates to the CloudFormation stack. If you need to modify a stack resource at a later time, we recommend that you relaunch the solution stack with a new tag value and migrate your VGW connections to the new CSRs.

10. To see details for the stack resources, choose the **Outputs** tab. The following tables describes each of these outputs in more detail.

<table>
<thead>
<tr>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSR1</td>
<td>Public IP address for CSR 1, which is necessary for manually configuring CSR instances or connecting the transit VPC to remote networks</td>
</tr>
<tr>
<td>CSR2</td>
<td>Public IP address for CSR 2, which is necessary for manually configuring CSR instances or connecting the transit VPC to remote networks</td>
</tr>
<tr>
<td>ConfigS3Bucket</td>
<td>S3 bucket created by this template, and used to store VPN connection information</td>
</tr>
<tr>
<td><strong>Important:</strong></td>
<td>If you plan on connecting an additional AWS account to the transit VPC, you must note this value. You will enter the name of this S3 bucket as a parameter in the transit-vpc-second-account template that you launch in <strong>Step 4</strong>.</td>
</tr>
<tr>
<td>BucketPrefix</td>
<td>Should match the string you entered in the template’s Prefix for S3 Objects value</td>
</tr>
<tr>
<td>SpokeVPCTag</td>
<td>Should match the spoke VPC tag name you entered in the template</td>
</tr>
</tbody>
</table>
### Key

<table>
<thead>
<tr>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SpokeVPCTagValue</td>
<td>Should match the spoke VPC tag value you entered in the template</td>
</tr>
<tr>
<td>PreferredPathTagName</td>
<td>Should match the preferred VPN endpoint tag name you entered in the template</td>
</tr>
</tbody>
</table>

**Note:** In addition to the VGW Poller (poller) and Cisco Configurator (configurator) Lambda functions, this solution includes a CloudFormation custom resource Lambda function (solution-helper), which runs only when the CloudFormation stack is launched, updated or deleted. When running this solution, you will see all Lambda functions in the AWS Lambda console, but only the two primary solution functions are regularly active. However, do not delete the solution-helper function as it is necessary to manage associated resources.

### Step 3. Tag the Spoke VPCs

After the transit VPC stack launch completes, you can apply tags to existing VGWs that you wish to add to the transit network. Make sure to use the tag name and tag value that you specified during the stack creation. These VGWs will be automatically connected to the transit VPC within a few minutes.

**Note:** For each subnet you intend to connect to the transit VPC network, make sure that you either enable route propagation or create a static routing entry in its route table. For more information, see [Route Tables for a Virtual Private Gateway](#) in the Amazon VPC User Guide.

For illustration purposes, assume you want to add a developer-environment VPC to your transit network, which has a VGW named `devVPC-VGW`.

1. In the left navigation pane of the Amazon VPC console, choose **Virtual Private Gateways**.
2. Select the VGW you want to modify, choose the **Tags** tab, and choose **Edit**.
3. Add the spoke VPC tag key and value that you defined in the AWS CloudFormation template. For our example, we didn’t change the default values for these parameters, so we will enter the **Key** `transitvpc:spoke` and the **Value** `true`.
Within a minute or less, the VGW Poller Lambda function will find the tag and create a VPN connection from the spoke VGW to the CSR instances located in the transit VPC.

4. Optional: Apply the preferred VPN endpoint tag that you defined in the AWS CloudFormation template. This setting is especially useful when integrating a transit VPC with stateful firewalls that require symmetric routing and need active/passive network paths.

The default value for this Key is `transitvpc:preferred-path` and its Value can be `none`, `CSR1`, or `CSR2`. The following table describes the behavior of each value.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>If a VGW does not have a preferred VPN endpoint tag, or has one with a value that is set to 'none', then both CSR instances are viewed as equal and each spoke VPC will independently choose to send traffic through either CSR instance.</td>
</tr>
<tr>
<td>CSR1</td>
<td>The transit VPC will configure CSR1 as the preferred route by this spoke VGW. AS path prepend will be used to make CSR1 more attractive route than CSR2.</td>
</tr>
<tr>
<td>CSR2</td>
<td>The transit VPC will configure CSR2 as the preferred route by this spoke VGW. AS path prepend will be used to make CSR2 a more attractive route than CSR1.</td>
</tr>
</tbody>
</table>

5. In the left navigation pane, choose **VPN Connections** to confirm the spoke VPC was successfully added to the transit network.
You should see two new VPN connections in pending state. This should soon change to available.

6. To see the tunnel status and BGP routes received, choose the **Tunnel Details** tab.

7. To test connectivity in your new transit network, we recommend pinging between EC2 instances (with appropriate security group configurations) in two spoke VPCs.

**Step 4. Connect an Additional AWS Account (Optional)**

As explained in the [AWS CloudFormation Templates](#) section, the `transit-vpc-second-account` template is used to launch the VGW Poller Lambda function in an additional AWS account, so that it can search for and add VPCs from that account to the transit network.

During the initial launch of the transit VPC in the primary account (Step 2), if you specified an additional AWS account to participate in the transit network, the solution will automatically configure permissions to the Amazon S3 bucket and AWS KMS customer master key. You then launch the template in that account and tag its spoke VPCs accordingly.
1. Log in to the AWS Management Console of the applicable account, and click the button to the right to launch the `transit-vpc-second-account` AWS CloudFormation template.

You can also download the template to use it as a starting point for your own implementation.

2. The template is launched in the US East (N. Virginia) Region by default. To launch the VGW Poller in a different AWS Region, use the region selector in the console navigation bar.

3. On the Select Template page, keep the default settings for Stack and Template Source.

4. On the Specify Details page, name the stack.

5. Under Parameters, review the parameters for the template and modify them as necessary. They include the following default values.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BucketName</td>
<td><code>transit-vpc&lt;xyz&gt;</code></td>
<td>Use the exact bucket name that was an output from the Transit VPC template you deployed in Step 2. All transit VPC configuration files are stored in the same S3 bucket.</td>
</tr>
<tr>
<td>BucketPrefix</td>
<td><code>vpnconfigs/</code></td>
<td>Use the exact string you entered for the Prefix for S3 Objects value when you launched the <code>transit-vpc-primary-account</code> template (Step 2).</td>
</tr>
</tbody>
</table>

6. Choose Next.

7. On the Options page, you can specify tags (key-value pairs) for resources in your stack and set additional options, and then choose Next.

8. On the Review page, review and confirm the settings. Be sure to check the box acknowledging that the template will create IAM resources.

9. Choose Create to deploy the stack.

You can view the status of the stack in the AWS CloudFormation console in the Status column. You should see a status of CREATE_COMPLETE in roughly five (5) minutes.

10. To see details for the stack resources, choose the Outputs tab. The following tables describes each of these outputs in more detail.
<table>
<thead>
<tr>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PollerFunction</td>
<td>The name of the Lambda poller function</td>
</tr>
<tr>
<td>PollerFunctionARN</td>
<td>The ARN for the new Lambda poller function</td>
</tr>
</tbody>
</table>

11. The VGW Poller Lambda function is now running in this additional AWS account, and you can apply tags to the VGWs that you wish to add to the transit network (see Step 3, Tag the Spoke VPCs).

Security

When you build systems on AWS infrastructure, security responsibilities are shared between you and AWS. This shared model can reduce your operational burden as AWS operates, manages, and controls the components from the host operating system and virtualization layer down to the physical security of the facilities in which the services operate. For more information about security on AWS, visit the AWS Security Center.

Security Groups

The security groups created in this solution are designed to control and isolate network traffic between the AWS Lambda functions, CSR instances, and remote VPN endpoints. To perform testing, troubleshooting, or remote configuration, you will need to update the CSR security group to allow inbound SSH traffic. We recommend that you review the security groups and further restrict access as needed once the deployment is up and running.

In the transit VPC network, all VPN connections originate from the CSR instances. Therefore, no inbound traffic is necessary other than for access to the CSR instances. This solution includes a security group rule that grants access to inbound SSH traffic from the Cisco Configurator Lambda function only.

Additional Security Settings

Password authorization is explicitly disabled. The Cisco Configurator Lambda function generates an SSH key pair, stores it securely in the Amazon S3 bucket, and uses that key pair for authentication to access the CSR instances. The Cisco Configurator Lambda function is configured to run inside the transit VPC only.

All files in the S3 bucket are encrypted using server-side encryption with AWS KMS (S3 SSE-KMS). An Amazon S3 bucket policy controls which additional accounts can have access to the bucket, and an AWS KMS key policy controls which accounts are authorized to use the solution-specific customer master key for decryption, therefore enabling those accounts to connect their VGWs to the transit VPC network. These policies may be modified manually to add additional accounts to the transit VPC network (see Appendix C for details).
Amazon CloudWatch

This solution creates the following CloudWatch rules and logs to invoke and log associated AWS Lambda functions.

CloudWatch Logs:

- `/aws/lambda/transit-vpc-poller`: Logged actions of the VGW Poller Lambda function
- `/aws/lambda/transit-vpc-cisco-configuration`: Logged actions of the Cisco Configurator Lambda function
- `/aws/lambda/transit-vpc-solution-helper-XXXXXXXXXXXXXXX` or `/aws/lambda/transit-vpc-poller-solution-helper-XXXXXXXXXXXXXXX`: Created by the solution-helper customer resource when launching the `transit-vpc-primary-account` AWS CloudFormation template or the `transit-vpc-poller` AWS CloudFormation template.

CloudWatch Rule:

- `vgw-poller-1min`: This rule invokes the VGW Poller Lambda function every (1) minute.

This solution also logs debugging, informational, warning, error, and critical messages for the VGW Poller and Cisco Configurator Lambda functions. To choose the type of messages to log, find the applicable function in the Lambda console and change the `LOG_LEVEL` environment variable to the applicable type of message.

<table>
<thead>
<tr>
<th>Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRITICAL</td>
<td>Logs will include information on anything that causes the function to fail.</td>
</tr>
<tr>
<td>ERROR</td>
<td>Logs will include information on anything that causes an operation to fail.</td>
</tr>
<tr>
<td></td>
<td>Logs will also include CRITICAL messages.</td>
</tr>
<tr>
<td>WARNING</td>
<td>Logs will include information on anything that can potentially cause inconsistencies in the function but might not necessarily cause the operation to fail. Logs will also include ERROR and CRITICAL messages.</td>
</tr>
<tr>
<td>INFO</td>
<td>Logs will include high-level information about how the function is operating.</td>
</tr>
<tr>
<td></td>
<td>Logs will also include WARNING, ERROR, and CRITICAL messages.</td>
</tr>
<tr>
<td>DEBUG</td>
<td>Logs will include information that might be helpful when debugging a problem with the function. Logs will also include INFO, WARNING, ERROR, and CRITICAL messages.</td>
</tr>
</tbody>
</table>
Testing

This solution includes additional AWS CloudFormation templates that you can use to test the transit network in your AWS account.

Transit VPC Test with Tsunami UDP

The transit-vpc-spoke-vpc-withec2-tsunamiudp AWS CloudFormation template creates the transit VPC, automatically installs the server-side tsunami_UDP protocol, and downloads two test files from an Amazon S3 bucket. Use the following steps to test your transit network with this template.

1. Identify the spoke VPCs you want to use for the test and tag them as follows:
   - Tag name: transitvpc:spoke
   - Tag value: true

   **Note:** To quickly create a VPC with an Amazon EC2 instance, use the transit-vpc-spoke-vpc-withec2 AWS CloudFormation template (see the next section).

2. Click the button to the right to launch the transit-vpc-spoke-vpc-withec2-tsunamiudp AWS CloudFormation template.
   You can also download the template.

3. Use the default settings. When the stack build completes, you will have a new transit VPC with a preinstalled Tsunami server and two test files.

4. To test connectivity in your new transit network, we recommend pinging (or using Tsunami UDP to transfer data) between EC2 instances (with appropriate security group configurations) in two spoke VPCs.

5. Go to the Tsunami_UDP_Usage_file and follow the instructions in 1. Very Quick Guide. This walks you through a simple test using the GET command from client to server.
Spoke VPC Templates

Use the following templates to quickly launch VPCs in your AWS account for your own testing protocol.

**transit-vpc-spoke-vpc.template:** This template launches a basic VPC with a VGW and route table. Use this to create new VPCs in your accounts or as a starting point to retrofit your existing VPC templates with examples for creating transit VPC tagged spoke VGWs and related resources.

**transit-vpc-spoke-vpc-withec2.template:** This template launches a more complete VPC along with a small EC2 instance. Use this template as a starting point for new VPCs or for testing connectivity between spoke VPCs.

Additional Resources

AWS services documentation
- AWS CloudFormation
- Amazon VPC
- AWS Lambda
- Amazon EC2 user guide for Linux instances
- Amazon S3
- Amazon CloudWatch
- AWS KMS
- AWS IAM

Cisco products and documentation
- AWS Marketplace offerings:
  - Cisco Cloud Services Router (CSR) 1000V - Bring Your Own License (BYOL) for Maximum Performance
  - Cisco Cloud Services Router (CSR) 1000V - AX Pkg. Max Performance
Appendix A: Component Details

VGW Poller

An Amazon CloudWatch rule invokes the VGW Poller Lambda function every minute. The VGW Poller is configured to iterate through each AWS Region of a customer’s account, searching for appropriately tagged spoke VGWs (transitvpc:spoke = true by default) that do not have existing VPN connections to the transit VPC. Once a spoke VGW is identified, the function creates corresponding customer gateways (if required) and VPN connections to each CSR. After creating VPN connections, the function retrieves the VPN configuration and saves it to an Amazon S3 bucket. The following diagram describes the VGW Poller logic in more detail.

Figure 3: VGW Poller logic

Similar logic is used to remove spoke VPCs from the transit VPC, however the VGW Poller function looks for connected spoke VGWs that either do not have the spoke VPC tag (it was removed) or that have a spoke VPC tag without the expected tag value (true by default). For example, by default, deleting the transitvpc:spoke tag or changing its value to <blank>, false, or any other value (besides true) will result in the VGW Poller removing the spoke VPC from the transit network.
Cisco Configurator
The Cisco Configurator is an AWS Lambda function that is invoked by Amazon S3 Put events to the solution’s S3 bucket. After the VGW Poller function writes new VPN connection details to the S3 bucket, the Cisco Configurator function parses the VPN connection information and generates the necessary config files to establish or remove VPN connections to spoke VPCs. It then pushes the configuration to the CSR instances using SSH. The following diagram describes the Cisco Configurator logic in more detail.

BGP and Failover
As soon as the Cisco configuration is applied onto the CSR instances, the VPN tunnels come up and Border Gateway Protocol (BGP) neighbor relationships are established.
By default, the transit VPC CSR instances will be deployed in an active/active configuration with each spoke VPC receiving equal cost routes. Customers can apply the preferred VPN endpoint tag on each spoke’s VGW, on a spoke-by-spoke basis, to configure the CSR instances in an active/standby configuration. By default, the preferred VPN endpoint tag name (key) is `transitvpc:preferred-path`, which accepts the following values.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;blank&gt; or 'none'</td>
<td>If a VGW does not have the preferred VPN endpoint tag, or has one with a tag value that is blank or set to 'none', then both CSR instances are viewed as equal and each spoke VPC will independently choose to send traffic through either CSR instance.</td>
</tr>
<tr>
<td>CSR1</td>
<td>The transit VPC will configure CSR1 as the preferred route by this spoke VGW. AS path prepending will be used to make CSR1 a more attractive route than CSR2.</td>
</tr>
<tr>
<td>CSR2</td>
<td>The transit VPC will configure CSR2 as the preferred route by this spoke VGW. AS path prepending will be used to make CSR2 a more attractive route than CSR1.</td>
</tr>
</tbody>
</table>
Appendix B: Existing VPC Template

This solution includes an AWS CloudFormation template that allows you to use an existing VPC as the transit VPC.

Review the following prerequisites before deploying the solution for an existing VPC:

- This solution’s AWS Lambda function requires access to an Amazon Simple Storage Service (Amazon S3) bucket that will store VPN connection information. Therefore, you must configure the Amazon VPC-provided Domain Name System (DNS) feature or a customer-provided DNS forwarder to allow the solution’s Lambda functions to resolve Amazon S3 domain names.

- By default, this template creates a new Amazon S3 endpoint and updates an existing VPC route table to provide Amazon S3 access to the solution’s Lambda functions. You can opt to use an existing Amazon S3 endpoint, a NAT gateway, a proxy server, or a corporate network instead, which you must configure in advance.

Click the button to the right to launch the transit-vpc-primary-account-existing-vpc AWS CloudFormation template. You can also download the template. Review the template parameters and modify them as necessary for your implementation.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSR Throughput Requirements</td>
<td>2x500Mbps</td>
<td>A drop-down box with four options: 2x500Mbps (c4.large), 2x1Gbps (c3.2xlarge), 2x2Gbps (c4.2xlarge), and 2x4.5Gbps (c4.4xlarge)</td>
</tr>
<tr>
<td>SSH Key to access CSR</td>
<td>&lt;Requires input&gt;</td>
<td>Public/private key pair, which allows you to connect securely to your instance after it launches. When you created an AWS account, this is the key pair you created in your preferred AWS Region.</td>
</tr>
<tr>
<td>License Model</td>
<td>LicenseIncluded</td>
<td>A drop-down box with two choices: LicenseIncluded and BYOL</td>
</tr>
<tr>
<td>Enable Termination Protection</td>
<td>Yes</td>
<td>Allows termination protection to be enabled on the CSR instances to help prevent accidental CSR termination (we recommend termination protection for production deployments).</td>
</tr>
<tr>
<td>Prefix for S3 Objects</td>
<td>vpnconfigs/</td>
<td>The text string you want to use to specify the name of the prefix for the Amazon S3 objects that are created. You must end the string with a forward slash (/).</td>
</tr>
<tr>
<td>Additional Account ID</td>
<td>&lt;Optional input&gt;</td>
<td>Enter the account ID of one additional AWS account that you want to connect to the transit network. This is necessary to grant that account access to the S3 bucket and the AWS KMS customer master key.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Note:</strong> Enter one additional account ID in this field. If you want to connect more than one additional AWS account to the transit network, you must manually configure permissions for the remaining accounts. See Appendix C for detailed instructions.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Existing VPC ID</td>
<td>&lt;Requires input&gt;</td>
<td>The ID of the existing VPC where you want to deploy the CSR instances.</td>
</tr>
<tr>
<td>1st Subnet Network</td>
<td>&lt;Requires input&gt;</td>
<td>CIDR block for the transit VPC subnet for the first CSR instance.</td>
</tr>
<tr>
<td>2nd Subnet Network</td>
<td>&lt;Requires input&gt;</td>
<td>CIDR block for the transit VPC subnet for the second CSR instance.</td>
</tr>
<tr>
<td>Create S3 Endpoint</td>
<td>Yes</td>
<td>Specify whether you want the solution to create an S3 endpoint in your VPC. If you provide alternative network connectivity to Amazon S3 from your VPC, choose No.</td>
</tr>
<tr>
<td>Route Table ID</td>
<td>rtb-12345678</td>
<td>The route table ID for the S3 endpoint to update with S3 routes.</td>
</tr>
<tr>
<td><strong>Note:</strong> Use this parameter only if you choose Yes to Create S3 Endpoint. If you choose No, ignore this parameter.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transit VPC BGP ASN</td>
<td>64512</td>
<td>BGP ASN to use for the transit VPC</td>
</tr>
<tr>
<td>Spoke VPC Tag Name</td>
<td>transitvpc:spoke</td>
<td>Tag name (key) to identify spoke VPCs to connect to the transit VPC. You can modify the tag name to align with any existing naming conventions. Use a name that is not likely to be used on VGWs for a different purpose to ensure you do not mistakenly add a VPC to the transit network.</td>
</tr>
<tr>
<td>Spoke VPC Tag Value</td>
<td>True</td>
<td>Tag value to determine which spoke VPCs to connect to the transit VPC. You can modify the tag value to align with any existing naming conventions. Be sure to use a value that is easy to understand and implement consistently.</td>
</tr>
<tr>
<td>Preferred VPN Endpoint Tag Name</td>
<td>transitvpc:prefer red-path</td>
<td>Tag name (key) to identify a preferred CSR instance for defining active/passive paths through the transit network. You can modify the tag name to align with any existing naming conventions. For more information about preferred path values and implementation details, see Step 3.</td>
</tr>
</tbody>
</table>
Appendix C: Configuring Permissions for Additional AWS Accounts

Each AWS account that you connect to the transit network requires access to the Amazon S3 bucket where configuration data is stored, as well as authorization to use the AWS Key Management Service (AWS KMS) customer master key for encrypting and decrypting transit VPC configuration files.

This solution automatically configures the necessary permissions for one additional account, using the account ID that you specify in the Additional AWS Account ID parameter in Step 2. If you want to add other AWS accounts (or you did not specify an additional AWS account ID when initially launching the AWS CloudFormation template in the primary account), you must complete the following steps to manually update the Amazon S3 bucket and AWS KMS policies to grant access to those accounts.

1. Note the account IDs for all AWS accounts you want to add to the transit VPC network.
2. Log in to the AWS Management Console of the primary account (where you launched the transit VPC).
3. Go to the Amazon S3 console and choose the bucket that was created when you launched the solution. (The bucket name shows in the Output tab of the CloudFormation stack.)
4. Choose Properties, and under Permissions choose Edit bucket policy.
5. Add a new line (as shown in red in the following code block) for each additional AWS account. If you connected one additional account, it should already have its own line.

```json
{
  "Version": "2008-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::<account-1-ID>:root",
          "arn:aws:iam::<account-2-ID>:root"
        ],
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:PutObjectAcl"
      ]
    }
  ]
}
```
6. Go to the AWS Key Management Service (AWS KMS) console, and in the left navigation pane, choose **Customer managed keys**.

7. Choose the encryption key for this solution (You will see “Transit VPC” in the key description), and in the **Key Policy** section, choose **Edit**.

8. In the list of roles allowed to use the master key, add a new line (as shown in red below) for each additional account ID.

```json
{
  "Sid": "Allow use of the key",
  "Effect": "Allow",
  "Principal": {
    "AWS": [
      "arn:aws:iam::<transit-vpc-primary-account-id>:role/TransitVPC-TransitVpcPollerRole-[cloudformation-id]",
      "arn:aws:iam::<transit-vpc-primary-account-id>:role/TransitVPC-CiscoConfigFunctionRole-[cloudformation-id]",
      "arn:aws:iam::<transit-vpc-primary-account-id>:role/TransitVPC-LambdaLoaderRole-[cloudformation-id]",
      "arn:aws:iam::<account-1-id>:root",
      "arn:aws:iam::<account-2-id>:root"
    ]
  } }
```

9. Select **Save changes**.

After you configure permissions for each AWS account that you want to add to the network, complete Step 3 to tag applicable spoke VPCs and Step 4 to launch the VGW Poller in those accounts.

**Appendix D: Integration with AWS Direct Connect**

Customers have different options for connecting on-premises networks to the transit VPC. A common approach is to manually configure the CSR instances with point-to-point VPN or DMVPN connections over the Internet.

Another approach is to leverage a detached virtual private gateway (VGW) to conceptually attach a VGW to a data center. In this approach, a customer creates a VGW, then adds a spoke VPC tag (default tag key `transitvpc:spoke`, default tag value `true`) without attaching the VGW to a specific VPC. This will cause the VGW to be automatically
connected to the transit VPC CSR instances, which will start broadcasting any routes they have learned to the new VGW. Then, to connect the VGW to remote networks, associate it with an AWS Direct Connect virtual interface or create a standard VGW VPN connection. Once a virtual interface or VPN connection is connected to the VGW, the VGW will start broadcasting any routes that it learned from the CSR instances over the remote connection, as depicted in the following diagram.

**Note:** We recommend that you use a different Border Gateway Protocol (BGP) Autonomous System Number (ASN) between your corporate data center and the customer gateway than you use for the transit VPC network. This will allow routes to be more easily propagated between your data center and your spoke VPCs.

![Diagram](image)

**Figure 6: Connecting to remote networks**

This is the recommended approach for customers who have up to 1 Gbps AWS Direct Connect connections. For larger AWS Direct Connect connections, we recommend establishing tunnels directly to the transit VPC CSR instances over either a public or private VIF.
Appendix E: Collection of Anonymous Data

This solution includes an option to send anonymous usage data to AWS. We use this data to better understand how customers use this solution to improve the services and products that we offer. When enabled, two event types trigger the collection and sending of anonymous data to AWS.

During initial deployment of the transit-vpc-primary-account AWS CloudFormation template, the following information is collected:

- **Solution ID**: The AWS solution identifier
- **Unique ID (UUID)**: Randomly generated, unique identifier for each transit VPC deployment
- **Timestamp**: Data-collection timestamp
- **CSR Type**: The CSR instance throughput requirements selected (e.g., 2x500Mbps)
- **CSR License Model**: BYOL or license included

Each time the VGW Poller AWS Lambda function adds a new spoke VPC, the following information is collected:

- **UUID**: Randomly generated, unique identifier for each VPN connection
- **Timestamp**: Data-collection timestamp
- **AWS Region**: VPN connections created (e.g., "us-east-1" : 2)

Note that AWS will own the data gathered via this survey. Data collection will be subject to the AWS Privacy Policy. To opt out of this feature, modify the AWS CloudFormation template mapping section as follows:

```
"Send" : {
  "AnonymousUsage" : { "Data" : "Yes" }
},
```

to

```
"Send" : {
  "AnonymousUsage" : { "Data" : "No" }
},
```

Send Us Feedback

We welcome your questions and comments. Please post your feedback on the AWS Solutions Discussion Forum.

You can visit the Transit VPC GitHub repository to download the templates and scripts for this solution, and to share your customizations with others.
Document Revisions

<table>
<thead>
<tr>
<th>Date</th>
<th>Change</th>
<th>In sections</th>
</tr>
</thead>
<tbody>
<tr>
<td>July 2016</td>
<td>Initial publication</td>
<td>-</td>
</tr>
<tr>
<td>August 2016</td>
<td>Addition of AWS KMS service and updated diagram, information about the collection of anonymous data, and new procedure to manually configure permissions for multiple accounts; Corrected links to Cisco CSR products</td>
<td>Changes throughout guide; addition of Appendix B and Appendix C</td>
</tr>
<tr>
<td>September 2016</td>
<td>Correction to IAM policy</td>
<td>Appendix C</td>
</tr>
<tr>
<td>October 2016</td>
<td>Addition of information about spoke VPC VPN connection removal, BGP preferred path route manipulation, and guidance for integrating the transit VPC with AWS Direct Connect</td>
<td>Changes throughout guide; addition of Appendix D</td>
</tr>
<tr>
<td>October 2016</td>
<td>Updated to support CSR version 16.3.1a, Asia Pacific (Seoul) and US East (Ohio) Regions, and c4 instance types (which provide a 2x4.5Gbps option)</td>
<td>Cost and Licenses; CloudFormation template parameters in Step 2</td>
</tr>
<tr>
<td>January 2017</td>
<td>Additional guidance about limiting resources in the transit VPC, and for testing spoke VPCs</td>
<td>Architectural Overview, Step 3, Testing</td>
</tr>
<tr>
<td>February 2017</td>
<td>Added note about customer updates to CloudFormation stack</td>
<td>AWS CloudFormation Templates</td>
</tr>
<tr>
<td>April 2017</td>
<td>Added information on new template that deploys CSRs in an existing VPC; clarification on software VPN</td>
<td>Architecture Overview; AWS CloudFormation Templates; Appendix B</td>
</tr>
<tr>
<td>June 2017</td>
<td>Clarification on S3 prefix string; added recommendation to use unique BGP ASNs when integrating with AWS Direct Connect</td>
<td>Step 2, Appendix B, Appendix D</td>
</tr>
<tr>
<td>September 2017</td>
<td>Added information on new subnet AZ parameters, log level environment variables, and anonymous data</td>
<td>Step 2, Amazon CloudWatch, Appendix E</td>
</tr>
<tr>
<td>December 2017</td>
<td>Removed reference to VPC peering; added note for spoke VPC route table configuration</td>
<td>Architectural Overview, Step 3</td>
</tr>
</tbody>
</table>

**Notices**

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents AWS’s current product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS’s products or services are provided “as is” without warranties, representations, or conditions of any kind, whether express or implied. AWS’s responsibilities and liabilities to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

The Transit Network VPC (Cisco CSR) solution is licensed under the terms of the Amazon Software License available at [https://aws.amazon.com/asl/](https://aws.amazon.com/asl/).

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.