Serverless Transit
Network Orchestrator
Implementation Guide
Serverless Transit Network Orchestrator: Implementation Guide
Copyright © 2022 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.
Table of Contents

Welcome ................................................................................................................................. 1
Cost .......................................................................................................................................... 2
Architecture overview ........................................................................................................... 4
Components ........................................................................................................................... 7
   AWS Lambda ....................................................................................................................... 7
   AWS Step Functions ........................................................................................................... 7
   Amazon DynamoDB ........................................................................................................... 7
   AWS Resource Access Manager ......................................................................................... 7
   Transit Network Management web interface ...................................................................... 8
   AWS Transit Gateway Network Manager .......................................................................... 8
Considerations ....................................................................................................................... 9
   AWS accounts .................................................................................................................... 9
   Transit Gateway Routing Tables ....................................................................................... 9
Tagging ................................................................................................................................. 9
   Manual Approval Tagging ................................................................................................. 11
   Copying tags to attachment ............................................................................................. 11
Choosing a Hub Account ....................................................................................................... 11
Spoke Template .................................................................................................................. 12
Regional deployment ........................................................................................................... 12
Serverless Transit Network Orchestrator update ................................................................ 12
Templates ............................................................................................................................ 13
Deployment .......................................................................................................................... 14
   Prerequisites .................................................................................................................... 14
      Activate AWS RAM for AWS Organizations accounts ................................................... 14
   Deployment overview ...................................................................................................... 14
   Step 1. (Optional) Launch the role stack .......................................................................... 15
   Step 2. Launch the hub stack ............................................................................................. 16
   Step 3. Launch the spoke stack .......................................................................................... 22
   Step 4. Add tags ................................................................................................................ 23
      Add tags to VPC ............................................................................................................. 23
      Add tags to Subnet ......................................................................................................... 23
Manage network activities ................................................................................................... 24
   Sign in to the Transit Network Management web interface ........................................... 24
   Manage network activities ............................................................................................... 24
   Access the dashboard ........................................................................................................ 24
   Access the Action Items .................................................................................................... 25
   Approve or Reject Requests ............................................................................................. 25
   View history of a request ................................................................................................... 25
   Item expiration ................................................................................................................... 25
Security .................................................................................................................................. 26
   IAM roles ........................................................................................................................... 26
   Amazon CloudFront ......................................................................................................... 26
   Amazon Cognito ............................................................................................................... 27
   Amazon EventBridge ........................................................................................................ 27
Resources ............................................................................................................................ 28
Update the stack ................................................................................................................... 29
   Hub template ................................................................................................................... 29
      Create DynamoDB backup for old table ...................................................................... 29
      Restore DynamoDB backup to new table .................................................................. 29
   Spoke template ............................................................................................................... 12
Solution workflows .............................................................................................................. 31
   Automated Approval Process .......................................................................................... 31
   Manual approval process ................................................................................................. 32
   Transit gateway peering attachment workflow ........................................................... 33
Automate setting up and managing your transit networks with Serverless Transit Network Orchestrator

Publication date: November 2019 (last update (p. 45): July 2022)

The Serverless Transit Network Orchestrator solution automates the process of setting up and managing transit networks in distributed AWS environments. A web interface is created to help you control, audit, and approve (transit) network changes. Serverless Transit Network Orchestrator supports AWS Organizations and standalone AWS accounts. You can use this solution with the default deployment template, or customize it to meet your specific use case.

AWS Transit Gateway connects Amazon Virtual Private Cloud (Amazon VPC) and on-premises networks through a central hub that controls how traffic is routed among all the connected networks, which act like spokes. This hub and spoke model simplify network management and reduces operational costs because each network only connects to AWS Transit Gateway. Any new VPC or subnet that is connected to AWS Transit Gateway is automatically available to every other network connected through AWS Transit Gateway.

This implementation guide discusses architectural considerations and configuration steps for deploying the Serverless Transit Network Orchestrator solution on Amazon Web Services (AWS). It includes links to AWS CloudFormation templates that launch and configure the AWS services required to deploy this solution using AWS best practices for security and availability.

The guide is intended for IT infrastructure architects, administrators, and networking professionals who have practical experience with networking and architecting in the AWS Cloud.
Cost

You are responsible for the cost of the AWS services used while running this solution. As of July 2022, the cost for running this solution with the default settings in the US East (N. Virginia) Region with two VPCs attached to AWS Transit Gateway, with each VPC containing two subnets and using Amazon DynamoDB to store network data is approximately **$80.60 per month**. Note that AWS Transit Gateway charges for the amount of data transferred through it.

This cost estimate assumes the following:

- The solution manages two VPCs attached to AWS Transit Gateway, each VPC containing two subnets in different Availability Zones.
- Automated queries are made to Amazon DynamoDB from an actively running Transit Network Management web interface every five minutes. This estimate does not include manual queries.
- A total of 1 GB of data is sent between all VPCs via the transit gateway.
- The number of requests to the GraphQL API is 10,000 a month.

<table>
<thead>
<tr>
<th>AWS Service</th>
<th>Dimensions</th>
<th>Total Cost/Month</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Variable Costs</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AWS Transit Gateway</td>
<td>Hourly charge (containing two VPC attachments)</td>
<td>$72.00</td>
</tr>
<tr>
<td>AWS Transit Gateway</td>
<td>Data-processing charge (data transfer of 1 GB from two attached Amazon VPCs)</td>
<td>$0.60</td>
</tr>
<tr>
<td>AWS Transit Gateway</td>
<td>Data processing and outbound inter-Region data transfer charge (data transfer of 1 GB between two inter-Region peer transit gateways)</td>
<td>$0.40</td>
</tr>
<tr>
<td>Amazon DynamoDB</td>
<td>Includes automated queries only</td>
<td>$0.00009</td>
</tr>
<tr>
<td>AWS AppSync</td>
<td>Includes auto approval workflow only</td>
<td>$0.035</td>
</tr>
<tr>
<td><strong>Fixed Costs</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Amazon EventBridge</td>
<td></td>
<td>$0.00003</td>
</tr>
<tr>
<td>AWS Web Application Firewall</td>
<td></td>
<td>$7.606</td>
</tr>
<tr>
<td><strong>Total:</strong></td>
<td></td>
<td><strong>80.60</strong></td>
</tr>
</tbody>
</table>
Note
When the solution is deployed with zero Amazon VPC attachments, the cost for running the solution is less than $8.10 per month.

Prices are subject to change. For full details, refer to the pricing webpage for each AWS service used in this solution.
Architecture overview

This solution includes an AWS CloudFormation template (aws-transit-network-orchestrator-hub) that you deploy in the primary account you want to act as the hub in the solution's hub-and-spoke model. For guidance on choosing a hub account, refer to Choosing a Hub Account (p. 11). This template launches all the components necessary to automatically connect your VPCs to AWS Transit Gateway.

Note
Before you launch the hub template, note the spoke account IDs or the AWS Organizations ARNs. You will enter them into the applicable template parameters during deployment.

The hub template launches resources for the following services:

- AWS Lambda
- AWS Step Functions
- Amazon DynamoDB
- Amazon EventBridge
- Amazon Simple Notification Service (Amazon SNS)
- AWS Resource Access Manager (AWS RAM)
- AWS Transit Gateway
- AWS Network Manager

The template also deploys a Transit Network Management web interface that includes AWS Network Manager, Amazon Simple Storage Service (Amazon S3), Amazon CloudFront, AWS AppSync, and Amazon Cognito.

The solution also includes a template (aws-transit-network-orchestrator-spoke) to deploy in spoke accounts.

Deploying this solution with the default parameters builds the following environment in the AWS Cloud.
1. This template deploys a CloudWatch Events rule that monitors VPC and subnet tags. To identify the VPCs (spoke accounts) for the solution to manage, tag the VPCs and the selected subnets within those VPCs.

2. This tag change is sent to the hub account through an EventBridge bus.

3. When the event is received in the hub account, an AWS Lambda function is initiated to start the Serverless Transit Network Orchestrator workflow. For more information, refer to Solution workflows (p. 31).

Note
You must wait for the hub stack launch to complete before you launch spoke templates. The spoke accounts depend on the Amazon CloudWatch Events bus policy that is created during the hub stack launch.

4. AWS Step Functions (Serverless Transit Network Orchestrator state machine) and Lambda process network requests from the spoke accounts and event details are stored in DynamoDB. You can approve requests automatically or manually.
5. If you choose to approve requests automatically, the VPC attaches to AWS Transit Gateway. If you choose to approve request manually, Amazon SNS sends an email to request approval. After the request is approved, the Serverless Transit Network Orchestrator state machine applies the network change.

6. If the request is rejected, DynamoDB and the spoke resources tag are updated with the rejected status. When a request is approved, the solution updates the route table associated with the subnet in the spoke account with a default route with AWS Transit Gateway as the target, which provides bi-directional connectivity. The solution workflow updates the subnet’s route table with the default route as defined in the hub template.

7. The transit gateway provisioned by the solution also gets registered with global network manager. Refer to AWS Transit Gateway Network Manager for more information.

The solution also adds (or updates) a Serverless Transit Network Orchestrator Status tag with the request status as a mechanism to update the spoke account user. The spoke account user checks on the status using either the Transit Network Management web interface (if they have permission) or by viewing the tag in their spoke account.

**Note**
Serverless Transit Network Orchestrator will not overwrite existing default routes with different targets.

The Transit Network Management web interface includes a ReactJS web application which is deployed into an Amazon S3 bucket configured for static web hosting. Amazon CloudFront is used to provide public access to the solution’s bucket contents. Amazon Cognito is used to manage user access to the user interface. If required, you can integrate the Cognito user pool with your existing SAML-based identity provider (for example, Microsoft Active Directory) to allow your users to log in with their existing credentials.

Users can view tagging event details and the history of network requests from different accounts, and monitor their status in the web interface. Administrators can accept or reject requests when manual approval is required.
Solution components

AWS Lambda

This solution deploys three AWS Lambda functions.

**custom-resource**: This function is responsible for solution helper tasks like generating unique ID for the deployment, sending metrics to aws-solutions and triggering the solution core state machine.

**state-machine**: This function performs all the transit gateway related tasks, including transit network changes, Amazon DynamoDB updates, sending Amazon Simple Notification Service (Amazon SNS) notifications, tagging spoke resources, and accepting AWS Resource Access Manager (AWS RAM) resource share invitations from the spoke account.

**tgw-peering**: This function handles creating, updating, and deleting transit gateway peering attachments, thereby establishing intra/inter-Region peering connections between transit gateways.

AWS Step Functions

The Serverless Transit Network Orchestrator state machine contains AWS Step Functions that orchestrate the changes required to tether the network components. The state machine activates network administrators to analyze each event and troubleshoot any unexpected errors.

The Peering Attachment state machine contains AWS Step Functions that coordinate the changes required to peer inter-Region transit gateway connections.

Amazon DynamoDB

Amazon DynamoDB stores all tagging events made by users in the spoke accounts. It activates the administrator to retain and audit network changes made based on the tag changes.

By default, items expire after 90 days, but you can change the value by changing the Audit Trail Retention Period parameter in the hub template.

AWS Resource Access Manager

This solution uses AWS Resource Access Manager (AWS RAM) to create a resource share for transit gateway and managed prefix lists (if provided). Accounts that were identified during the hub template deployment, or within AWS Organizations, depending on your network environment, are shared through the transit gateway.

For accounts that use AWS Organizations, you must manually activate AWS RAM in the Organizations console and obtain the AWS Organizations management account ID and organization ID. AWS RAM allows you to share your resources through AWS Organizations. For steps to activate AWS RAM with AWS Organizations, refer to Activate AWS RAM for AWS Organizations Accounts (p. 14).
Transit Network Management web interface

The Transit Network Management web interface includes a ReactJS web application and is hosted in Amazon S3, delivered by CloudFront. The CloudFront distribution is utilizing a CloudFront Function to add security-related HTTP headers. Users authenticate for the web application through Amazon Cognito. The web interface leverages AWS AppSync to interact with DynamoDB and calls Lambda functions to initiate the Serverless Transit Network Orchestrator state machine in the manual approval workflow (p. 32). The web interface provides a dashboard for administrators to resolve manual approval requests for network changes and allows other users to view network changes.

AWS Transit Gateway Network Manager

This solution deploys or uses an existing Global Network in the hub account. AWS Transit Gateway Network Manager provides a single global view of your private network. This solution also automatically registers the AWS Transit Gateway managed by the solution. This feature allows you to centrally monitor your network from the dashboard of the AWS Transit Gateway Network Manager.
Deployment considerations

AWS accounts

Serverless Transit Network Orchestrator supports and can automatically activate authentication of AWS accounts that use AWS Organizations.

The solution also supports accounts that do not use AWS Organizations. For those accounts, each individual account (that deploys the spoke template) in your network must be authenticated individually. Enter each spoke account ID in the **Principals** parameter in the hub template. Authentication allows a standalone AWS account to initiate the Serverless Transit Network Orchestrator workflow to create transit gateway attachments to the VPCs.

Transit gateway routing tables

Transit gateway route tables configure routing for your transit gateway attachments. Serverless Transit Network Orchestrator uses routing tables in the tag value to associate a transit gateway route table with a transit gateway attachment, and to add a route from a route table to the attachment. To establish an association and propagation, the VPC administrator tags the network with the appropriate key-value pair, which generates a tag request that is sent to the hub account. For more information about tagging, refer to Tagging (p. 9).

This solution includes the following default routing tables: Flat, Isolated, Infrastructure, and On-premises. However, you can create your own custom routing tables to work with this solution. For a sample table of policy types and guidance for creating your own routing tables, refer to Create custom route tables (p. 36).

Tagging

Tags identify applicable resources (VPCs and subnets) in your spoke accounts. Tags allow CRUD operations to run on the transit gateway route table associations and propagation.

**Note**
Verify that you have the appropriate access privileges to tag VPCs in spoke accounts. Or, identify the appropriate administrator in your organization.

The solution uses the following tags for VPCs.

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Associate-with</td>
<td><em>&lt;Requires input&gt;</em></td>
<td>The default key is Associate-with. The value can be one of the default route table names (Flat, Isolated, Infrastructure, or On-premises) or a custom key. You can change the name of the key in the template during initial configuration, but you must use the same key name when you tag the VPC.</td>
</tr>
</tbody>
</table>
Key | Value | Description
--- | --- | ---
| Propagate-to | <Requires input> | The default key is Propagate-to. The value can be one or more default route table names (Flat, Isolated, Infrastructure, or On-premises) a key or keys that you created. You can change the name of the key in the template during initial configuration, but you must use the same key name when you tag the VPC. For sample route table options, refer to Create custom route tables (p. 36). |

**Important**
For this solution to manage the VPC, the VPC in the spoke account must be tagged with both the **Associate-with** and **Propagate-to** keys. You must also add or remove both keys at the same time. By default, the tags are configured for automatic approval.

The solution uses the following tag for subnets.

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attach-to-tgw</td>
<td>&lt;Leave blank&gt;</td>
<td>The default key is Attach-to-tgw. Do NOT enter a value. You can change the name of the key in the template during initial configuration, but you must use the same key name when you tag the subnet.</td>
</tr>
</tbody>
</table>

For an AWS Transit Gateway attachment to a VPC, you can add only one subnet per Availability Zone. If you tag a second subnet in the same Availability Zone, the subnet will not attach to the AWS Transit Gateway. However, AWS Transit Gateway is added as a destination to the default route defined in the hub template in the route table associated with the second subnet. When this happens, the solution generates a **DuplicateSubnetsInSameZoneError** error.

The solution uses the following tag for AWS Transit Gateway.

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TgwPeer</td>
<td>&lt;Requires input&gt;</td>
<td>The default tag key is TgwPeer. The value must follow the format <code>&lt;transit-gateway-</code></td>
</tr>
</tbody>
</table>
Manual Approval Tagging

### Key | Value | Description
--- | --- | ---
| **id** | _<aws_region>_ | For example, if you want to create a peering attachment with a transit gateway (tgw-2222) in the Ohio Region (us-east-2), the tag value would be tgw-222_us-east-2.

You can add multiple peering transit gateways by using the delimiter _/>. If you want to peer with tgw-3333 in us-west-1, the tag value would be tgw-2222_us-east-2/tgw-3333_us-west-1.

The tag value can be updated at any time to create new or delete old peering attachments. To delete all peering attachments managed by the solution, the tag value can be set to **delete**.

To route traffic between peered transit gateways, you must add a static route to the selected transit gateway route table that points to the transit gateway peering attachment. After you create the route, associate the same transit gateway route table with the transit gateway peering attachment. Refer to Add a route to the transit gateway route table for more information.

### Manual approval tagging

Administrators can change from the default automatic approval setup to manual approval by adding the **ApprovalRequired** tag to the AWS Transit Gateway route table. For information about enabling manual approval tagging, refer to Set up for manual approvals (p. 39).

### Copying tags to attachment

Customers can copy the tags they need from the VPC over to the transit gateway attachments. To copy the tags, specify the keys of the tags to be copied in the **ListOfVpcTagsForAttachment** input parameter of the hub template. The specified tags will then be copied over from the VPC to the transit gateway attachment. The tags will be copied to the attachments in both Hub and Spoke accounts.

### Choosing a hub account

We recommend deploying the hub template in an account that uses AWS Direct Connect or has existing VPN connections because it enables you to create transit gateway attachments to a Direct Connect Gateway(s) or VPN connection(s). For information about extending your network, creating VPN attachments, and attaching a transit gateway to a Direct Connect gateway, refer to On-premises connectivity (p. 40).
Spoke template

Deploy the spoke template in account(s) that contain your VPC(s). If you want to attach a VPC in your hub account with AWS Transit Gateway, you must deploy the spoke template in your hub account.

Regional deployment

This solution uses AWS AppSync, Amazon Cognito, AWS Transit Gateway, and Amazon EventBridge which are available in specific AWS Regions only. Therefore, you must launch this solution in a Region where these services are available. For the most current service availability by Region, refer to the AWS Regional Service List.

Serverless Transit Network Orchestrator update

If you have previously deployed the solution, you must update the solution’s CloudFormation hub stack and spoke stack to get the latest version of the solution’s framework. For details, refer to Update the Stack (p. 29).
AWS CloudFormation templates

This solution uses AWS CloudFormation to automate the deployment of Serverless Transit Network Orchestrator in the AWS Cloud. It includes the following AWS CloudFormation templates, which you can download before deployment.

aws-transit-network-orchestrator-hub.template: Use this template to launch the solution and all associated components in your AWS network hub account. The default configuration deploys AWS Transit Gateway, four AWS Transit Gateway route tables, AWS Step Functions (to orchestrate VPC and Transit Gateway attachments), an AWS Resource Access Manager resource share, an Amazon Simple Notification Service topic, an AWS AppSync API with WAF, an Amazon DynamoDB table, a global network in AWS Transit Gateway network manager, an Amazon Cognito user pool, one Amazon CloudFront distribution with CloudFront function, Amazon Simple Storage Service buckets, Amazon EventBridge event bus and rules, AWS Identity and Access Management (IAM) roles, and the Transit Network Management web interface for network management. You can also customize the template based on your specific needs.

aws-transit-network-orchestrator-spoke.template: Use this template to launch the solution and all associated components in your spoke account. The default configuration deploys EventBridge and IAM roles. You can also customize the template based on your specific needs.

aws-transit-network-orchestrator-organization-role.template: Use this template to create an IAM role in the Organizations Management account. The hub account requires the role to create names for the transit gateway attachments that are easily identifiable, using a combination of OU path and VPC name.
Automated deployment

Before you launch the automated deployment, review the architecture and other considerations discussed in this guide. Follow the step-by-step instructions in this section to configure and deploy the solution into your account.

**Time to deploy:** Approximately 25 minutes

**Prerequisites**

If your accounts are part of AWS Organizations, you must manually activate AWS Resource Access Manager (AWS RAM) in the Organizations console and obtain the AWS Organizations management account ID and organization ID before deploying the Serverless Transit Network Orchestrator templates.

**Activate AWS RAM for AWS Organizations accounts**

Use the following procedure to activate AWS RAM using the AWS Organizations console.

1. Sign in to the AWS Organizations console.
2. In the navigation pane, select **Settings**, scroll down to **AWS RAM**, and select **Enable access**.

Activate the sharing option in the AWS RAM console.

1. Sign in to the AWS RAM console.
2. In the navigation pane, select **Settings**.
3. Select **Enable sharing within your AWS Organization**.
4. Choose **Save settings**.

**Deployment overview**

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

**Step 1. (Optional) Launch the role stack (p. 15)**

- Launch the AWS CloudFormation template in your Organizations Management account.
- Enter values for the required parameter: Network (Hub) Account.

**Step 2. Launch the hub stack (p. 16)**

- Launch the AWS CloudFormation template in your hub account.
- Enter values for the required parameters under the following sections: **Stack Name**, **Account List or AWS Organizations ARN**, and **Console login information email**.
- Review the other template parameters and adjust, if necessary.
Step 1. (Optional) Launch the role stack

1. Sign in to your AWS spoke account using the AWS Management Console and select the button to launch the `aws-transit-network-orchestrator-organization-role` AWS CloudFormation template.

   ![Launch Template]

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

2. Launch this template in the same Region as the hub and spoke template. The template launches in the US East (N. Virginia) Region by default.

3. On the **Create stack** page, verify that the correct template URL shows in the Amazon S3 URL text box and choose **Next**.

4. On the **Specify stack details** page, assign a name to your solution stack.

5. Under **Parameters**, review the parameters for the template and modify them as necessary.

   This stack uses the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Account ID of the network account where AWS Transit Gateway resides</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Network (Hub) Account</td>
<td><code>&lt;Requires input&gt;</code></td>
<td>The account ID for the hub account.</td>
</tr>
</tbody>
</table>

Record the ARN for the role from the **Outputs** section of the stack. The ARN will be needed as input for `OrganizationManagementAccountRoleArn` parameter in the Hub template.
Step 2. Launch the hub stack

**Important**
This solution includes an option to send anonymous operational metrics to AWS. We use this data to better understand how customers use this solution and related services and products. AWS owns the data gathered though this survey. Data collection is subject to the [AWS Privacy Policy](https://aws.amazon.com/privacy/).

To opt out of this feature, download the template, modify the AWS CloudFormation mapping section, and then use the AWS CloudFormation console to upload your template and deploy the solution. For more information, refer to the [Collection of operational metrics (p. 43)](https://aws.amazon.com/privacy/) section of this guide.

This automated AWS CloudFormation template deploys Serverless Transit Network Orchestrator.

**Note**
You are responsible for the cost of the AWS services used while running this solution. Refer to the [Cost (p. 2)](https://aws.amazon.com/pricing/) section for more details. For full details, refer to the pricing webpage for each AWS service used in this solution.

1. Sign in to the AWS Management Console with your AWS network hub account and select the button to launch the `aws-transit-network-orchestrator-hub` AWS CloudFormation template.

You can also [download the template](https://aws.amazon.com/privacy/) as a starting point for your own implementation.

2. The template launches in the US East (N. Virginia) Region by default. To launch this solution in a different AWS Region, use the Region selector in the console navigation bar.

**Note**
This solution uses AWS AppSync, Amazon Cognito, AWS Transit Gateway, and Amazon EventBridge which are available in specific AWS Regions only. Therefore, you must launch this solution in a Region where these services are available. For the most current service availability by Region, refer to [AWS Regional Services List](https://aws.amazon.com/about-aws/global-infrastructure/region-info/).

3. On the **Create stack** page, verify that the correct template URL shows in the **Amazon S3 URL** text box and choose **Next**.

4. On the **Specify stack details** page, assign a name to your solution stack.

5. Under **Parameters**, review the parameters for the template and modify them as necessary.

This stack uses the following parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Account Structure</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Principal Type</td>
<td><strong>List of Accounts</strong></td>
<td>Choose whether to use the default <strong>List of Accounts</strong> or select <strong>AWS Organizations ARN</strong>. For guidance, refer to <a href="https://aws.amazon.com/about-aws/global-infrastructure/region-info/">AWS Accounts (p. 9)</a>.</td>
</tr>
<tr>
<td>Account List or AWS Organizations ARN</td>
<td><strong>&lt;Requires input&gt;</strong></td>
<td>To use an account list, enter a comma-separated list of</td>
</tr>
</tbody>
</table>
### Parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS account numbers</td>
<td></td>
<td>AWS account numbers. For example, 123456789012. To use AWS Organizations, enter the AWS Organizations ARN to share transit gateway with the principals. For example, arn:aws:organizations::&lt;org_master_account_ID&gt;:organization/o-&lt;example-org-ID&gt;. For additional guidance to identify the ARN value, refer to Identify the AWS Organizations ARN (p. 35).</td>
</tr>
<tr>
<td>IAM Role ARN of Management Account</td>
<td>&lt;Optional input&gt;</td>
<td>To tag attachments with the account name and OU path, provide ARN for the role in the management account which can be assumed by the hub account. Leave blank if deploying this solution in the management account.</td>
</tr>
<tr>
<td>Web UI Security Settings</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AllowListedRanges</td>
<td>&lt;Requires input&gt;</td>
<td>Comma separated list of CIDR ranges allowed to access Graphql API. To allow the entire internet, use 0.0.0.0/1,128.0.0.0/1.</td>
</tr>
<tr>
<td>Notification Settings</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receive approval notifications</td>
<td>No</td>
<td>Choose whether to receive approval notifications.</td>
</tr>
<tr>
<td>Notification Email for the network admins</td>
<td>&lt;Optional input&gt;</td>
<td>The email address for approval notifications. To use this parameter, you must set the Receive approval notifications parameter to Yes.</td>
</tr>
<tr>
<td>User Settings</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Console login information email</td>
<td>&lt;Requires input&gt;</td>
<td>The email address of the administrator user for the web interface. After launch, an email will be sent to this address with a temporary password for the web interface.</td>
</tr>
<tr>
<td>Admin Username</td>
<td>adminuser</td>
<td>The username for network admins with full read and write permissions to the Transit Network Management web interface.</td>
</tr>
</tbody>
</table>
### Step 2. Launch the hub stack

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Read-Only Username</strong></td>
<td>readonlyuser</td>
<td>The username for users with read-only permission to the Transit Network Management web interface.</td>
</tr>
<tr>
<td><strong>AWS Transit Gateway Settings</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| **Provide the existing transit gateway ID** | *Optional input* | The existing transit gateway ID in the current Region. For example, tgw-a1b2c3d4e5.  

**Note**  
If you don't provide a value, the solution creates a new transit gateway. You must ensure that the existing transit gateway has activated auto accept shared attachments flag.
### Step 2. Launch the hub stack

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provide the existing global network</td>
<td>&lt;Optional input&gt;</td>
<td>The existing global network id. For example, <code>global-network-01231231231231231</code>. Note: If you don’t provide a value, the solution creates a new global network. If you use this solution in more than one Region it is strongly recommended to use the global network created by the first Serverless Transit Network Orchestrator solution deployment to register all the transit gateways deployed by the solution in all the Regions with the same global network. If the provided transit gateway is already registered with the provided global network ID, unregister the provided transit gateway from the provided global network ID before updating the CloudFormation template. The CloudFormation template will rollback if the existing global network is not unregistered.</td>
</tr>
</tbody>
</table>

**SAML Identity Provider Configuration (Optional)**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cognito Domain Prefix</td>
<td>&lt;Optional input&gt;</td>
<td>This prefix will become part of the Cognito Hosted UI to sign in users to the web UI.</td>
</tr>
<tr>
<td>SAML Provider Name</td>
<td>&lt;Optional input&gt;</td>
<td>If you want to connect an external Identity Provider, specify a name that will show up on the UI.</td>
</tr>
</tbody>
</table>
## Step 2. Launch the hub stack

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAML Provider Metadata URL</td>
<td>&lt;Optional input&gt;</td>
<td>If you want to connect an external Identity Provider, enter the URL to the metadata file of your SAML-based Identity Provider. The URL must begin with https://.</td>
</tr>
</tbody>
</table>

### VPC Route Table Settings

**Choose the type of destination for target AWS Transit Gateway**

<table>
<thead>
<tr>
<th>Destination Type</th>
<th>Default</th>
<th>Description</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>All-traffic (0/0)</td>
<td></td>
<td>Specify the default route setting for the route table associated with the tagged subnets. Choose from All-traffic (0/0), RFC-1918 (10/8, 172.16/12, 192.168/16), Custom-Destinations or, Configure-Manually.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: If the route already exists, the solution will not overwrite it.</td>
<td></td>
</tr>
</tbody>
</table>

**Provide a comma separated list of CIDR Blocks**

<table>
<thead>
<tr>
<th>Default</th>
<th>Option to provide CIDR block(s). For example, 192.168.1.0/24, 192.168.2.0/24</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;Optional input&gt;</td>
<td>Note: OPTIONAL if providing prefix list id(s).</td>
</tr>
</tbody>
</table>

**Provide the list of Customer-managed Prefix List IDs**

<table>
<thead>
<tr>
<th>Default</th>
<th>Option to provide customer-managed Prefix List ID(s). For example, pl-abcd1234, pl-efgh5678</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;Optional input&gt;</td>
<td>Note: OPTIONAL if providing CIDR block(s).</td>
</tr>
</tbody>
</table>

### Tag Settings
## Parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
</table>
| Tag key for TGW Attachment                                               | Attach-to-tgw | Specify a custom tag key name without value to initiate the transit gateway attachment workflow.  
**Note**  
After initial deployment, do not change this solution’s default parameter. If you change this parameter after deployment, you will have to manually update the tags on your VPCs. |
| Tag key for TGW Route Table Association with TGW Attachment              | Associate-with | Specify a custom tag key name to initiate the transit gateway route table association with transit gateway attachment workflow.  
**Note**  
After initial deployment, do not change this solution’s default parameter. If you change this parameter after deployment, you will have to manually update the tags on your VPCs. |
| Tag key for Route Propagation to TGW Route Table(s)                      | Propagate-to | Specify a custom tag key name to initiate the route propagation to transit gateway route table(s) workflow.  
**Note**  
After initial deployment, do not change this solution’s default parameter. If you change this parameter after deployment, you will have to manually update the tags on your VPCs. |
Step 3. Launch the spoke stack

Use this procedure to configure spoke accounts.

**Note**
You must wait for the hub stack deployment to complete before you launch spoke templates. The spoke templates depend on the Amazon CloudWatch Events bus policy that is created during the hub stack launch. Additionally, all templates should be deployed in the same preferred region.

1. Sign in to your AWS spoke account using the AWS Management Console and select the button below to launch the `aws-transit-network-orchestrator-spoke` AWS CloudFormation template.

### Table: Parameter Settings

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Transit Gateway Peering Tag</strong></td>
<td>TgwPeer</td>
<td>Transit Gateway tag to monitor for peering connections. The tag value must follow the format <code>tgw-id_aws-region/tgw-id_aws-region</code>. For example, use <code>tgw-12345678_us-east-1/tgw-567890123_us-east-2</code> to create peering attachments with the two peers. The value can be updated at any time per requirement.</td>
</tr>
<tr>
<td><em>(Optional) Comma separated list of VPC tag keys to copy from VPC to TGW Attachments</em></td>
<td>Associate-with, Propagate-to</td>
<td>Comma separated list of tag keys (do not include Name). If the VPC has these tag keys, the tag key/value is copied to any created TGW attachment.</td>
</tr>
</tbody>
</table>

6. Choose **Next**.
7. On the **Configure stack options** page, choose **Next**.
8. On the **Review** page, review and confirm the settings. Check the box acknowledging that the template will create IAM resources.
9. Choose **Create stack** 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 approximately 25 minutes.

After the stack is created, you will receive two emails that contain temporary passwords for the read-only user and the admin user. If you activated approval notification, Amazon SNS sends a subscription confirmation email with a link to the Transit Network Management web interface. You can also find the link to the web interface in the AWS CloudFormation stack **Outputs** tab. The link is the **Value** of the **Console URL**. The system-generated password must be changed the first time that you sign in.

**Note**
The temporary account expires if you do not log in within seven days. Your new password must be at least 10 characters long.
You can also download the template as a starting point for your own implementation.

2. Launch this template in the same Region as the hub template. The template launches in the US East (N. Virginia) Region by default.

3. On the Create stack page, verify that the correct template URL shows in the Amazon S3 URL text box and choose Next.

4. On the Specify stack details page, assign a name to your solution stack.

5. Under Parameters, review the parameter for the template and modify it as necessary.

   This stack uses the following parameter.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Account ID of the network account where AWS Transit Gateway resides</td>
<td><strong>&lt;Requires input&gt;</strong></td>
<td>The account ID for the hub account.</td>
</tr>
</tbody>
</table>

6. Choose Next.

7. On the Configure stack options page, 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 stack to deploy the stack.

You can view the status of the stack in the AWS CloudFormation console in the Status column.

**Step 4. Add tags**

Review the tags used by the solution as highlighted in the Tagging (p. 9) section. Go to the spoke account and follow steps to add tags.

**Add tags to VPC**

1. Navigate to the VPC console. Select Tags and Manage tags, then choose Add new tag.

2. Add given key-pairs in a single action. Set Tag Value as needed, for example:
   - **Associate-with:** Flat
   - **Propagate-to:** Flat/Infrastructure

**Add tags to Subnet**

1. Navigate to the Subnet console.

2. Select the subnet that you want to attach to Transit Gateway.
### Step 5. Manage network activities

You can use the Transit Network Management web interface to access the dashboard to view network changes; access action items to view; approve or reject network requests when manual approval is required; and view the history of all changes made within Serverless Transit Network Orchestrator.

**Sign in to the Transit Network Management web interface**

After the hub stack is successfully deployed, you will receive two emails containing a link to the Transit Network Management web interface and login credentials. By default, one Amazon Cognito **admin**user (in the admin group) and one Amazon Cognito **readonly**user (in the read-only group) are created. For more information, refer to Managing and Searching for User Accounts in the Amazon Cognito Developer Guide.

If you have configured an external SAML-based Identity Provider in step 2, instead of signing in with username and password, you can click the button that redirects to your identity provider’s login page. On the first login, every user will be automatically added to the **ReadOnlyUserGroup** and thereby granted read access to the web interface. After a user has logged in with read access, you can assign them to the **AdminGroup** via Amazon Cognito if needed.

Select the link to open the web interface and enter the provided user credentials to sign in. The system-generated password must be changed the first time that you sign in.

**Note**

The temporary account expires if you do not log in within seven days. Your new password must be at least 10 characters long.

**Manage network activities**

You can use the Transit Network Management web interface to access the dashboard to view network changes; access action items to view; approve or reject network requests when manual approval is required; and view the history of all changes made within Serverless Transit Network Orchestrator.

**Access the dashboard**

The Dashboard tab displays the fields containing information about network changes stored in Amazon DynamoDB such as **VPC ID**, **VPC CIDR**, **Status**, **Association Route Table**, **Propagation Route Tables**, **Spoke Account**, **Subnet ID**, **Availability Zone**, and other relevant information. You have the flexibility to sort by any of these fields. You can view the **Status** of each network change including whether it was approved, rejected, auto-approved or auto-rejected.
Access the action items

The Action Items tab displays the requests that must be manually approved. If you chose to automatically approve requests, this tab will be empty. For manual approvals, each request contains the same fields as those in the Dashboard tab. Requests can have the following status: requested, processing, or failed. Failed requests are highlighted in red. The reason for the failure is noted in the comment column.

Approve or reject requests

When manual approval is activated for requests, the administrator approves or rejects the request using the Transit Network Management web interface. Only users in the admin group can approve or reject requests. Users from the read only group can only view requests. When an administrator approves or rejects the request, the status is set to processing.

When a request is processing, no further action can be taken from the web interface. The web interface calls an AWS Lambda function which initiates the Serverless Transit Network Orchestrator state machine to process the request. After the process completes, the status of the request is updated by the Serverless Transit Network Orchestrator state machine accordingly and reflected in the web interface.

View history of a request

View History displays the complete history of a selected network request. To view the history of a request, select the request from either the Dashboard or Action Items tab and then choose View History.

Item expiration

Information and history for a VPC are set to expire based on the time you specify in the hub template at stack launch. The default time is 90 days. Expired requests are automatically deleted from DynamoDB within approximately 48 hours and are not shown in the web interface after deletion. You can specify a different value by changing the Audit Trail Retention Period parameter in the AWS CloudFormation hub stack at stack launch.
Security

When you build systems on AWS infrastructure, security responsibilities are shared between you and AWS. This shared model can reduce your operational burden when 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 AWS Cloud Security.

IAM roles

AWS Identity and Access Management (IAM) roles allow customers to assign granular access policies and permissions to services and users on AWS. This solution creates IAM roles and sets permissions in the respective accounts to allow the solution to assume a defined role in the spoke and management account to make changes when necessary. As can be seen in below workflow, hub account assumes role in the Management account.

Amazon CloudFront

This solution deploys a static website hosted in an Amazon S3 bucket. To help reduce latency and improve security, this solution includes an Amazon CloudFront distribution with an origin access identity, which is a special CloudFront user that helps provide public access to the solution’s website bucket contents. For more information, refer to Restricting Access to Amazon S3 Content by Using an Origin Access Identity.
Amazon Cognito

This solution creates Amazon Cognito user accounts for signing in to the Transit Network Management web interface, and grants the administrator and the read-only users with the appropriate permissions to control user access to data.

If you connect an external Identity Provider via SAML, be aware that every user from your Identity Provider will have read access to the web UI. To prevent giving read access to all users by default, you must modify the cognito-trigger lambda function deployed by this solution.

Amazon EventBridge

The Amazon EventBridge event rule contains a filtering mechanism that identifies events that are tagged with the tags defined in the hub template. If the tag keys match the Associate-with, Propagate-to, and Attach-to-tgw values, the Amazon CloudWatch Events rule invokes the Serverless Transit Network Orchestrator workflow. For more information, refer to Creating an Event Bus in the Amazon EventBridge User Guide.
Additional resources

- AWS Transit Gateway
- AWS Step Functions
- AWS Resource Access Manager
- AWS AppSync
- Amazon DynamoDB
- Amazon Cognito
- Amazon CloudFront
- AWS CloudFormation
- AWS Lambda
- Amazon CloudWatch Events
- Amazon EventBridge
- AWS Organizations
- Amazon Virtual Private Cloud
- Amazon Simple Storage Service
Update the stack

If you have previously deployed the solution, follow this procedure to update the Serverless Transit Network Orchestrator CloudFormation hub stack and spoke stack to get the latest version of the solution’s framework.

Hub template

Deploy the v3 template along side v2 template and delete v2 template later.

1. Download the v3 hub template. Make the following modification under the mappings section and save the file.

   ```
   NotificationConfiguration:
     SNS:
       DisplayName: "AWS Transit Network Change Approval Notification for v3"
       TopicName: AWS-Transit-Network-Approval-Notifications-v3
     EventBridge:
       Bus:
         Name: "STNO-EventBridge-v3"
   ```

2. Deploy the saved template from Step 1. in the same Region as older deployment. For parameters, review Transit Gateway settings:

   - Provide the existing transit gateway id: copy the transit gateway id created by v2 deployment.
   - Provide the existing global network id: copy the global network created by v2 deployment (leave blank if v2 did not create global network).

   **Note**

   Ignore the parameters that were removed from the v3 hub template. Record the rest of the values you entered for v2 deployment.

3. Migrate DynamoDB table data to retain historical network-change events.

   For the purpose of these instructions, consider old_table as the table created by older STNO deployment and new_table as the table created by the newer (v3.0.0) STNO deployment.

Create DynamoDB backup for old table

1. Navigate to the DynamoDB console. For the old_table deployed by the old deployment, choose `<StackName>-DynamoDBTable-<XXXX>>.

2. Select **Backups** from the navigation menu. Choose **Create backup**, then **Create on-demand backup**.

3. Under **Backup settings**, select **Customize settings** then **Backup with DynamoDB**.

4. Enter **Backup name** and choose **Create backup**.

Restore DynamoDB backup to new table

Delete the DynamoDB table created by the new deployment as we will be restoring from old_table.
1. Navigate to the DynamoDB console. Select the backup created earlier under table **Backups**, and choose **Restore**.
2. For **Name of restored table**, enter the same name to the new_table as deployed by the template. You can find the table name from template **Outputs** section.
3. Select **Restore the entire table**, then **Same Region**.
4. Under encryption settings, select **Owned by Amazon DynamoDB** and choose **Restore**.

new_table will be restored with the old_table backup in approximately 10 minutes. You can now delete the older serverless-transit-network-orchestrator stack deployment.

**Spoke template**

For spoke template, update the CloudFormation stack by following these steps.

1. Download the v3 spoke template. Make the following modification under the mappings section and save the file.

   ```
   EventBridge:
     Bus:
       Name: "STNO-EventBridge-v3" # should be same as hub template
   ```

2. Navigate to the CloudFormation console, and select the older deployment.
3. Choose **Update**, then choose **Replace current template**. Choose **Upload a template file** and select the template saved in Step 1.
4. Choose **Next**. Select I acknowledge that AWS CloudFormation might create IAM resources with custom names then choose **Update Stack**.
Solution workflows

The following workflows detail the automated and the manual approval processes of the Serverless Transit Network Orchestrator solution.

Automated Approval Process

Figure 2: Automated approval process for Serverless Transit Network Orchestrator

1. When the VPC and/or subnet is tagged, Amazon CloudWatch Events is initiated. The target for this CloudWatch Events rule is Amazon EventBridge in the hub account.

2. EventBridge receives CloudWatch Events in the hub account. The EventBridge policy is configured to accept events from the trusted accounts provided by the customer in the hub template.

   Note
   If you use AWS Organizations, the EventBridge policy trusts all accounts in the organization.
3. EventBridge invokes another CloudWatch Events rule in the hub account that has the Serverless Transit Network Orchestrator Lambda function as the target.

4. The Lambda function analyzes the event and starts a state machine using the CloudWatch Events details as input to the state machine.

5. Depending on the event, the state machine can create, update, and delete transit gateway attachments to the VPC. It can also create or update transit gateway route table associations, and turn on or off transit gateway route table propagations. The state machine adds a new status tag to the VPC or the subnet with the status of the request.

6. After the state machine finishes the tasks described in Step 5, it updates the Amazon DynamoDB table to activate the network administrator to audit the network change history. The changes in DynamoDB are automatically reflected in the web interface dashboard. Administrators and users can log into the Transit Network Management web interface to review the history of all changes that occurred in the network.

Manual approval process

Figure 3: Manual approval process for Serverless Transit Network Orchestrator
1. When the VPC and/or subnet is tagged, Amazon CloudWatch Events are initiated. The target for this CloudWatch Events rule is Amazon EventBridge in the hub account.

2. EventBridge receives CloudWatch Events in the hub account. The EventBridge policy is configured to accept events from the trusted accounts provided by the customer in the hub template parameter.

   **Note**
   If you use AWS Organizations, the built-in Amazon EventBridge policy will trust all accounts in the organization.

3. EventBridge invokes another CloudWatch Events rule in the hub account that has the Serverless Transit Network Orchestrator Lambda function as the target.

4. If the ApprovalRequired flag is set to Yes in the transit gateway route table, the state machine skips any changes. To set up this flag, refer to On-premises connectivity (p. 40).

5. The state machine notifies the network administrator by email using the email you specified during hub stack launch. The state machine also adds a new status tag to the VPC or subnet with the status of the request.

6. Amazon Simple Notification Service (Amazon SNS) generates a notification that is sent to the administrator.

7. In DynamoDB, the status is set to requested. The status change is shown in the web interface in the **Action Items** tab.

8. The administrator signs in to the web interface, navigates to the **Action Items** tab, and chooses to either approve or reject each request.

9. After the admin resolves the action items, AWS AppSync GraphQL is initiated and changes the DynamoDB request status to processing.

10. AWS AppSync GraphQL also invokes the Serverless Transit Network Orchestrator Lambda function to initiate the downstream approve or reject process.

11. The Lambda function validates the event and starts the state machine to complete the request.

12. If the administrator approved the request, the state machine makes changes based on the details in the event. If the request is rejected, the state machine skips all the network related changes. In both cases, the status tag is updated with the status of the request.

13. The DynamoDB table is updated with either the approved or rejected status and other event details.

---

**Transit gateway peering attachment workflow**

1. **Transit Gateway**: Tag-change event.
2. **CloudWatch event (event-based)**.
3. **Lambda function**.
4. **AWS Transit Gateway**.
5. **Transit gateway peering attachment**.
6. **AWS Transit Gateway**.
Figure 4: Transit gateway peering attachment workflow

1. When the transit gateway is tagged, CloudWatch Events is initiated. The target for this CloudWatch Events rule is the transit gateway peering attachment Lambda function in the hub account.

2. The transit gateway peering Lambda function creates the peering attachment between the transit gateways based on the tag key and value. The peering attachment state transitions from InitializingRequest to PendingAcceptance.

3. The Lambda function accepts the peering attachment request in the remote Region.
Identify the AWS Organizations ARN

To use the Serverless Transit Network Orchestrator with accounts connected to AWS Organizations, you must specify the AWS Organizations Amazon Resource Name (ARN) when you launch the hub template. The ARN value consists of the AWS Organizations management account ID and the organization ID. You can build the ARN string manually if you have access to the AWS Organizations management account ID and the organization ID, or you can use the AWS Command Line Interface (AWS CLI) to query the Organization ARN.

**Note**
If you do not have access to the management account ID and the organizations ID, contact your organization's management account administrator.

Use the following steps to build the AWS Organizations ARN manually after you have the AWS Organizations management account ID and the organization ID. Use the following sample ARN and replace the placeholders with your account information.

1. Sign in to [AWS Organizations console](https://console.aws.amazon.com/organizations) from your organization’s management account.
2. In the **Accounts** tab, identify the management account and record the **Account ID**.
3. Select the **Settings** tab in the upper-right corner of the page.
4. From the **Organization details** page, record the **Organization ID**.
5. Use the following sample to manually build the Organization ARN. Replace the placeholders with your management account and organization IDs.

```
arn:aws:organizations::<12-digit-org_master_account_ID>::organization/o-<org-ID>
```

To use the AWS CLI to query the ARN, use the `describe-organization` API call. To set up AWS CLI, refer to [Configuring the AWS CLI](https://docs.aws.amazon.com/cli/latest/userguide/install-cli.html) in the [AWS Command Line Interface User Guide](https://docs.aws.amazon.com/cli/latest/userguide/index.html).
Create custom route tables

The Serverless Transit Network Orchestrator (STNO) creates the following default transit gateway route tables: Flat, Isolated, Infrastructure, and On-premises. Each route table and suggested propagations include a policy for common use cases.

- **Flat**: VPCs associated with the Flat policy can reach other VPCs associated with the Flat, Shared Services, or Hybrid policies. The Flat policy enables a VPC to have connectivity to many other VPCs.

- **Isolated**: VPCs associated with the Isolated policy can only reach VPCs with the Shared Services and Hybrid policies. VPCs in the Isolated policy cannot use AWS Transit Gateway to connect to other VPCs in the Isolated policy. This policy is for VPCs that do not communicate with each other.

- **Infrastructure**: VPCs associated with Shared Services can reach other VPCs associated with the Isolated, Flat, or Hybrid policies. The Infrastructure policy is used for VPCs that many other VPCs may rely on, such as shared authentication, shared tooling, or orchestration tools.

- **On-premises**: This route table is used for connecting to on-premises through either VPN or AWS Direct Connect. Associate your On-premises connections to the On-premises route table.

<table>
<thead>
<tr>
<th>Policy Types</th>
<th>Associate with (route table name)</th>
<th>Propagate to (list of route table names)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flat (East-West)</td>
<td>Flat</td>
<td>Flat, On-premises, Infrastructure</td>
</tr>
<tr>
<td>Isolated (North-South)</td>
<td>Isolated</td>
<td>On-premises, Infrastructure</td>
</tr>
<tr>
<td>SharedServices</td>
<td>Infrastructure</td>
<td>Flat, On-premises, Isolated</td>
</tr>
<tr>
<td>Hybrid</td>
<td>On-premises</td>
<td>Flat, Infrastructure, Isolated</td>
</tr>
</tbody>
</table>

**Note**

In this implementation guide, a policy is defined by both an association to a single transit gateway route table, and the transit gateway route table propagation. To implement these concepts, both the association and propagation must be tagged on each spoke VPC according to the intended design because the policies below are not centrally managed. Inconsistent tagging can create drift between the desired policy and what is configured. You can use the ApprovalRequired tag on route tables that need manual control. By default, Serverless Transit Network Orchestrator is set up for automatic approval, but you can change this tag to set up manual approval. For more information, refer to On-premises connectivity (p. 40).

Custom route tables

If the default policies do not meet your requirements, you can create your own transit gateway route table configurations. For example, if you need a policy that allows developers to create VPCs that do not have access to sensitive resources in the Isolated or Infrastructure route tables, you can create a new Development policy by creating a new transit gateway route table that propagates to the Flat, On-premises, and unique route tables. Those route tables would also propagate to the new Development route table. For instructions on how to set up a custom route table, refer to Create a Custom Route Table and Attachment (p. 37).
Alternatively, if you do not need the provided Flat policy, you can modify the existing Flat policy to meet your custom requirements. Turn off propagation to the Infrastructure route table and remove the Flat propagation from the Infrastructure route table.

**Note**
You do not need a separate route table for each VPC to achieve segmentation. Segmentation is accomplished by controlling the propagation. For example, an Isolated route table does not propagate to itself and, as a result, nothing associated with an Isolated route table is able to reach other Isolated resources through the transit gateway.

The administrator can create new transit gateway route tables in the Amazon Virtual Private Cloud console in the hub account. The combination of route tables and propagation provided with the transit gateway allows for a wide variety of connection policies.

## Create a custom route table and attachment

Use the following sample table and steps as a guide for setting your routing policies. For this example, you implement a Development policy and route table.

<table>
<thead>
<tr>
<th>Policy Types</th>
<th>Associate with (route table name)</th>
<th>Propagate to (list of route table names)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flat (East-West)</td>
<td>Flat</td>
<td>Flat, On-premises, Infrastructure, Development</td>
</tr>
<tr>
<td>Isolated (North-South)</td>
<td>Isolated</td>
<td>On-premises, Infrastructure</td>
</tr>
<tr>
<td>SharedServices</td>
<td>Infrastructure</td>
<td>Flat, On-premises, Isolated</td>
</tr>
<tr>
<td>Hybrid</td>
<td>On-premises</td>
<td>Flat, Infrastructure, Isolated, Development</td>
</tr>
<tr>
<td>Development</td>
<td>Development</td>
<td>Development, Flat, Hybrid</td>
</tr>
</tbody>
</table>

1. In the hub account, sign in to the Amazon Virtual Private Cloud console.
2. In the navigation pane, choose Transit Gateway Route Tables.
3. Choose Create Transit Gateway Route Table.
4. For Name tag, enter a name for the route table. For this example, enter Development.
5. For Transit Gateway ID, select the appropriate transit gateway.
6. Choose Create Transit Gateway Route Table.
   You will receive a confirmation message.
7. Select Close.
8. Optional: If you want changes to this route table to be manually approved:
   a. Select the newly created route table from the list.
   b. Choose the Tags tab.
   c. Choose Add/Edit tags and then choose Create Tag.
   d. In the Key field, enter ApprovalRequired and in the Value field, enter Yes.
   e. Choose Save.

After the new route table is created, determine the access model. For two transit gateway attachments to communicate, each of their associated route tables must have each other’s routes.
First, determine where your new route table should propagate routes, defined by the other transit gateway route tables. For example, should the Infrastructure route table be propagated from your new route table?

Next, check that the other route tables are reciprocating the propagation for two-way communication. For example, you may want to propagate Infrastructure, Flat, and Hybrid route table to your new route table.

**Note**
If your custom route table requires access to VPCs that have already been attached, you must change the `Propagate-to` tag for each spoke VPC to include your new route table.

To associate a new VPC to this route table:

1. Tag the new VPC with the `Associate-with` key and reference the new route table name in the value.
   
   For example, `Associate-with: <ExampleRouteTable>`

2. Tag the new VPC with the `Propagate-to` key and reference the route tables you want to propagate to from the previous step.
   
   For example, `Propagate-to: Infrastructure, Flat, Hybrid`

3. Tag one subnet in each Availability Zone.
   
   For example, `Attach-to-tgw: <leave blank>`

   **Note**
   If you configured manual approval, you may need to sign in to the Transit Network Management web interface to approve the change. If that is the case, you will receive an email with the request that will contain the link to the Transit Network Management web interface.

To confirm the attachment, you can look for attachments on the transit gateway in the VPC management console, in the Transit Network Management interface, or in the state machine history.

For more information about transit gateway route tables, refer to [Transit gateway route tables](#).
Set up for manual approvals

The Serverless Transit Network Orchestrator can invoke a manual approval workflow for route table associations. Use the following procedure to activate this feature.

1. Navigate to the Amazon Virtual Private Cloud console in the hub account.
2. In the navigation pane, choose Transit Gateway Route Tables.
3. Choose the Tags tab and select Add/Edit Tags.
4. In the Key column, locate the ApprovalRequired key and update the Value to Yes.

After the network administrator updates the tag value, future requests or changes related to the transit gateway route table association will require approval and, if activated, will initiate a notification being sent to the administrator. The administrator must approve or reject the request. For more information, refer to Manual approval process (p. 32).
On-premises connectivity

Serverless Transit Network Orchestrator builds the base network, giving you the automation to attach VPCs to AWS Transit Gateway. You can extend your network by creating transit gateway route tables using the web interface, creating VPN attachments, or attaching a transit gateway to an AWS Direct Connect gateway.

For instructions on how to manually attach a VPN to the transit gateway for on-premises connectivity, refer to Transit gateway VPN attachments in the Amazon VPC Transit Gateways guide.

For instructions on how to manually attach Direct Connect to the transit gateway for on-premises connectivity, refer to Transit gateway attachments to a Direct Connect gateway in the Amazon VPC Transit Gateways guide.
Troubleshooting

The solution logs error, warning, informational, and debugging messages for the solution's AWS 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>ERROR</td>
<td>Logs will include information on anything that causes an operation to fail.</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 messages.</td>
</tr>
<tr>
<td>INFO</td>
<td>Logs will include high-level information about how the function is operating. Logs will also include ERROR and WARN 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 ERROR, WARNING, and INFO messages.</td>
</tr>
</tbody>
</table>
Uninstall the solution

You can uninstall the Serverless Transit Network Orchestrator solution from the AWS Management Console or by using the AWS Command Line Interface. You must manually delete the AWS Transit Gateway, route tables, AWS Global Transit Network, and Amazon Simple Storage Service (Amazon S3) buckets created by this solution. AWS Solutions Implementations do not automatically delete these resources in case you have stored data to retain.

Using the AWS Management Console

1. Sign in to the AWS CloudFormation console.
2. On the Stacks page, select this solution’s installation stack.
3. Choose Delete.

Using AWS Command Line Interface

Determine whether the AWS Command Line Interface (AWS CLI) is available in your environment. For installation instructions, refer to What Is the AWS Command Line Interface in the AWS CLI User Guide. After confirming that the AWS CLI is available, run the following command.

```
$ aws cloudformation delete-stack --stack-name <installation-stack-name>
```

Manually delete resources

This solution is configured to retain specific solution-created resources to prevent accidental data loss if you decide to delete the AWS CloudFormation stack. After uninstalling the solution, you can manually delete these resources if you do not need to retain the data. Refer to these guides to delete the following resources.

- Delete a transit gateway
- Delete an Amazon S3 bucket
- Delete a route table
- Delete an AWS Network Manager global network
Collection of operational metrics

This solution includes an option to send anonymous usage data to AWS. We use this data to better understand how customers use this solution and related products and services. When enabled, the following information is collected and sent to AWS every time a solution template is launched:

- **Solution ID**: The AWS solution identifier
- **Unique ID (UUID)**: Randomly generated, unique identifier for each solution deployment
- **Timestamp**: Data-collection timestamp
- **Launch Region**: The AWS Region where the stack was launched

AWS owns the data gathered via this survey. Data collection is subject to the AWS Privacy Policy. To opt out of this feature, complete the following steps before launching the AWS CloudFormation template.

1. Download the AWS CloudFormation template to your local hard drive.
2. Open the AWS CloudFormation template with a text editor.
3. Modify the AWS CloudFormation template mapping section from:

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

   to:

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

4. Sign in to the AWS CloudFormation console.
5. Select **Create stack**.
6. On the **Create stack** page, **Specify template** section, select **Upload a template file**.
7. Under **Upload a template file**, choose **Choose file** and select the edited template from your local drive.
8. Choose **Next** and follow the steps in **Launch the role stack (p. 43)** in the Automated deployment section of this guide.
Source code

This solution is coded in Python and the web interface is coded in ReactJS.

Visit our GitHub repository to download the source files for this solution and to share your customizations with others.
## Document revisions

<table>
<thead>
<tr>
<th>Date</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>November 2019</td>
<td>Initial release.</td>
</tr>
<tr>
<td>August 2020</td>
<td>Release version 2.0.0: Added support for AWS Transit Gateway inter-Region peering and Amazon VPC prefix lists; updated cost estimate; for additional information about changes to v2.0, refer to the CHANGELOG.md file in the GitHub repository.</td>
</tr>
<tr>
<td>December 2020</td>
<td>Added instructions for updating the solution's CloudFormation stacks.</td>
</tr>
<tr>
<td>July 2021</td>
<td>Document enhancements, including updates to diagrams and architecture overview section.</td>
</tr>
<tr>
<td>March 2022</td>
<td>Release version 3.0.0: Added support for parallel tagging events of multiple Subnets/VPCs, Organizations tag policies, and descriptive transit gateway attachment name. Also, moved transit gateway peering attachment capability to AWS Lambda. For additional information about changes in v3.0.0, refer to the CHANGELOG.md file in the GitHub repository.</td>
</tr>
<tr>
<td>April 2022</td>
<td>Release version 3.0.1: Minor updates and bug fixes. For information about changes in v3.0.1, refer to the CHANGELOG.md file in the GitHub repository.</td>
</tr>
<tr>
<td>July 2022</td>
<td>Release version 3.1.0: Minor updates and bug fixes. For more information, refer to the CHANGELOG.md file in the GitHub repository.</td>
</tr>
</tbody>
</table>
Contributors

- Chaitanya Deolankar
- Garvit Singh
- Thiemo Belmega
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 current AWS 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 products or services are provided “as is” without warranties, representations, or conditions of any kind, whether express or implied. The responsibilities and liabilities of AWS 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 Serverless Transit Network Orchestrator solution is licensed under the terms of the Apache License Version 2.0 available at The Apache Software Foundation
AWS glossary

For the latest AWS terminology, see the AWS glossary in the AWS General Reference.