aws

User Guide

AWS RTB Fabric

AWS RTB Fabric: User Guide

Table of Contents

What is AWS RTB Fabric?

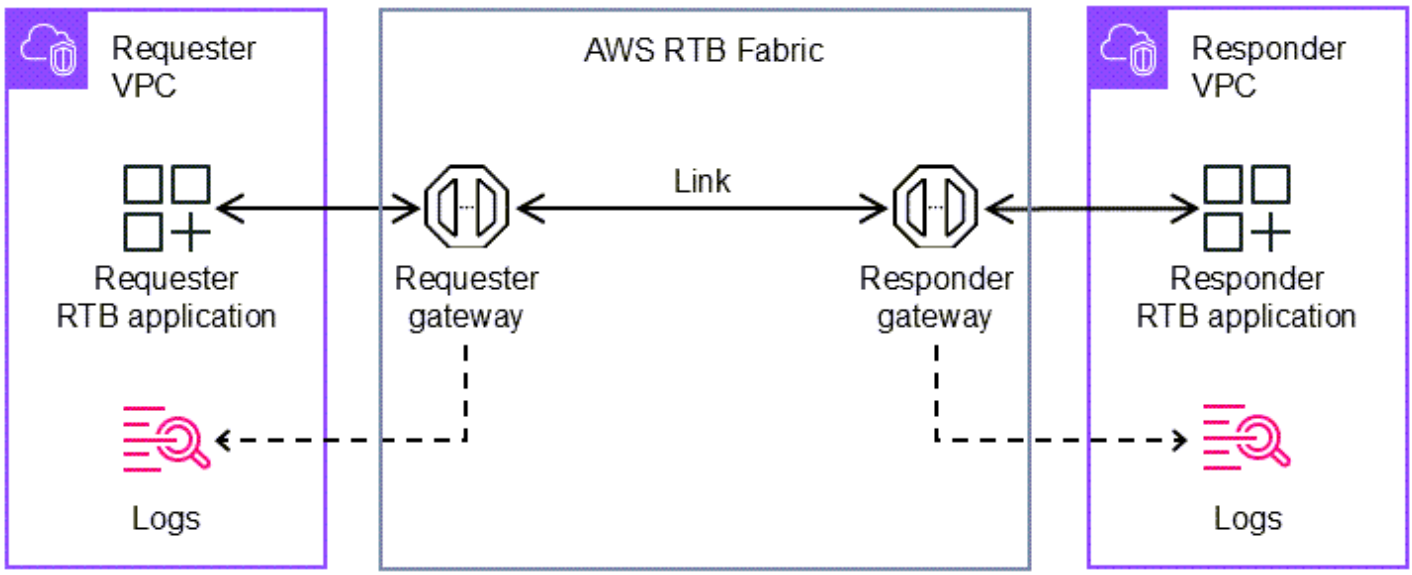
RTB Fabric is an AWS service that provides secure, low-latency infrastructure for connecting real-time bidding (RTB) applications. Rather than hosting applications directly, RTB Fabric acts as the connecting fabric that enables your applications to communicate efficiently over private networks instead of the public internet. You maintain complete control over your applications, data, and bidding decisions, while RTB Fabric provides the underlying infrastructure for secure, reliable connectivity. RTB Fabric is available in multiple AWS Regions. For a complete list of supported Regions, see RTB Fabric endpoints and quotas in the *AWS General Reference*.

> **Note**
>
> RTB Fabric is exclusively designed and optimized for real-time bidding traffic patterns and protocols.

How RTB Fabric works

RTB Fabric enables bidirectional communication between RTB applications through a request-response model. The service creates secure pathways that allow one application to send OpenRTB requests to another application and receive responses, all while maintaining low latency and high throughput.

The following diagram shows the high-level architecture of RTB Fabric:



This architecture demonstrates how RTB Fabric acts as connecting infrastructure between your applications. Requester applications connect to RTB Fabric requester gateways, which forward requests through links to RTB Fabric responder gateways. The responder gateways then forward requests to responder applications, which process them and return responses through the same pathway.

RTB Fabric components

RTB Fabric provides the following infrastructure components to connect your applications:

- **Requester RTB application** – A customer-owned and customer-operated RTB application that initiates bid requests to other RTB applications. Examples include supply-side platforms (SSPs), ad servers, and publishers. Requester applications run in customer-controlled environments and connect to RTB Fabric through AWS managed requester gateways colocated with the customer's VPC.
- **External requester RTB application** – An RTB application that runs outside AWS and does not use VPCs, but sends bid requests to RTB Fabric.
- **Requester gateway** – An AWS managed network endpoint that operates colocated with the customer's VPC to facilitate secure communication from requester applications to RTB Fabric infrastructure. Requester gateways serve as connection points that route outbound bid requests and receive responses but do not host, store, or process the customer's application logic or business data.
- **Link** – An AWS managed connection component within RTB Fabric infrastructure that enables secure, bidirectional communication between requester and responder gateways. Links are created by requester gateways and must be accepted by the AWS account that owns the responder gateway before becoming active.

  > **Note**
  >
  > To create a link, requester gateway owners need the target responder gateway ID. Contact your RTB Fabric partner directly to obtain their gateway ID. AWS does not provide gateway IDs.

- **Responder gateway** – An AWS managed network endpoint that operates colocated with the customer's VPC to facilitate secure communication from RTB Fabric infrastructure to responder applications. Responder gateways serve as connection points that route inbound bid requests

and return responses but do not host, store, or process the customer's application logic or business data.

- **Responder RTB application** – A customer-owned and customer-operated RTB application that receives and processes bid requests from requester applications. Examples include demand-side platforms (DSPs). Responder applications run in customer-controlled environments and connect to RTB Fabric through AWS managed responder gateways colocated with the customer's VPC.
- **External responder RTB application** – An RTB application that runs outside AWS and does not use VPCs, but receives bid requests from RTB Fabric.
- **Logs** – Optional requester and responder logs generated by links that can be delivered to external AWS services such as Amazon CloudWatch Logs or Amazon Simple Storage Service (Amazon S3). Both requester and responder sides can independently configure log delivery through sampling rates for error logs and filter logs. Log delivery requires setup of log delivery sources, destinations, and appropriate AWS Identity and Access Management (IAM) permissions. Logs are not stored within the RTB Fabric infrastructure.

> **Note**
>
> For important service limits including HTTP request timeouts and performance considerations, see Quotas for AWS RTB Fabric.

Modules

Modules are configurable components that process RTB traffic flowing through your links to implement rate limiting, filtering, error handling, and other traffic management capabilities. RTB Fabric provides built-in modules for rate limiting, OpenRTB filtering, and error masking that are available at no additional charge. You can configure modules using the RTB Fabric API. For more information, see Modules.

Common use cases

RTB Fabric supports three connectivity patterns. The following are three common example use cases:

- **Internal to internal** – Both requester and responder applications are RTB Fabric customers with single applications on each end of the link. Requester applications send requests to RTB

Fabric requester gateways, which forward them through links to RTB Fabric responder gateways. Responder gateways forward requests to responder applications, which process them and return responses through the same pathway. A typical example is an SSP connecting to a DSP that both use RTB Fabric.

- **External to internal** – Requester applications are external to RTB Fabric (not RTB Fabric customers) and may represent multiple applications that can be located on or outside AWS. External requester applications send requests to RTB Fabric requester gateways, which forward them through links to RTB Fabric responder gateways. Responder gateways forward requests to responder applications, which process them and send responses back through the same pathway. A typical example is an SSP that is not on AWS connecting to a DSP that uses RTB Fabric.
- **Internal to external** – Requester applications send requests to RTB Fabric requester gateways, which forward them through links to external RTB Fabric responder gateways. External responder gateways forward requests to a single responder application that is external to RTB Fabric (not an RTB Fabric customer) and can be located on or outside AWS. The external application processes requests and returns responses through the same pathway. A typical example is an SSP that uses RTB Fabric to connect to a DSP that is not on AWS.

Accessing RTB Fabric

You can work with RTB Fabric in the following ways:

- **AWS Management Console** – The console is a browser-based interface that you can use to create and manage RTB Fabric gateways and links. Sign in to the AWS Management Console and open the RTB Fabric console at https://console.aws.amazon.com/rtbfabric.
- **AWS CloudFormation** – Provides templates to create and manage RTB Fabric resources as code. You can use AWS CloudFormation to automate the deployment and configuration of gateways and links. For more information, see the .
- **AWS Command Line Interface** – Provides commands for a broad set of AWS services, including RTB Fabric. It's supported on Windows, macOS, and Linux. For more information about getting started with the AWS CLI, see the AWS Command Line Interface User Guide.
- **AWS SDKs** – Provide language-specific APIs and take care of many of the connection details, such as calculating signatures, handling request retries, and error handling. For more information, see AWS SDKs and Tools Reference Guide.
- **HTTPS API** – Provides programmatic access to RTB Fabric and AWS. The HTTPS API lets you issue HTTPS requests directly to the service. When you use the HTTPS API, you must include code to

digitally sign requests using your credentials. For more information, see the AWS RTB Fabric API Reference.

RTB Fabric pricing

For information about RTB Fabric pricing, see RTB Fabric Pricing.

Setting up AWS RTB Fabric

Before you can use RTB Fabric, you must complete several setup tasks including account configuration, IAM permissions, and VPC networking. This chapter walks you through the prerequisites and configuration steps needed to get started with RTB Fabric.

Prerequisites

Before you begin using RTB Fabric, ensure you have completed the following prerequisites:

- **AWS account** – You need an AWS account with appropriate permissions to create and manage RTB Fabric resources. If you don't have an AWS account, you can sign up at https://portal.aws.amazon.com/billing/signup.
- **IAM permissions** – You must configure appropriate AWS Identity and Access Management (IAM) permissions to create and manage RTB Fabric resources. RTB Fabric requires permissions for core operations such as creating gateways and links, as well as optional permissions for features like log delivery.

  For details about required permissions and example policies, see the section called "Identity and access management". RTB Fabric also uses service-linked roles that are automatically created when you first use the service.

VPC requirements

RTB Fabric gateways connect to your existing virtual private clouds (VPCs). Most customers already have an existing VPC where their RTB application (SSP or DSP) is running. RTB Fabric gateways connect to this existing VPC to facilitate real-time bidding traffic. If you don't have an existing VPC, see What is Amazon VPC in the *Amazon VPC User Guide* to create one.

Your VPC must meet the following requirements for RTB Fabric:

- **IP address availability** – RTB Fabric supports IPv6 only. Ensure you have sufficient free IPv6 addresses in each VPC subnet where you plan to connect your gateway. The number of required addresses depends on your expected traffic scale.
- **Security group configuration** – Configure security groups with appropriate inbound rules based on your role:

  - **For requesters**: HTTPS (TCP port 443) inbound from your VPC Classless Inter-Domain Routing (CIDR) range or compute instance IP.
  - **For responders**: HTTPS (TCP port 443) inbound to your VPC CIDR range or fleet endpoint IP.
- **Network access controls** – Configure network ACLs, security groups, and routes to prevent unauthorized access within your AWS account.
- **DNS configuration** – For requesters, set your DNS TTL (time to live) value to 30 seconds for clients sending requests to the service.

When selecting or configuring your VPC for RTB Fabric, ensure you have the following information ready:

- **VPC ID** – The VPC where your RTB application runs.
- **Subnet IDs** – Subnets with sufficient IPv6 addresses for gateway connections.
- **Availability Zone configuration** – RTB Fabric supports single Availability Zone deployment by default. Multi-AZ deployment is optional and may require a service quota increase.
- **Security group ID** – Configured with the appropriate inbound rules for your role (requester or responder).

You will provide this VPC information when creating RTB Fabric gateways to connect your RTB applications.

RTB Fabric gateways

RTB Fabric gateways are AWS managed network endpoints that operate colocated with your VPC to facilitate secure communication between RTB applications and RTB Fabric infrastructure. Gateways serve as connection points that route RTB traffic. There are two types of gateways: requester gateways that route outbound bid requests and receive responses, and responder gateways that route inbound bid requests and return responses.

Requester gateways

Requester gateways are RTB Fabric infrastructure components that serve as connection points for customer applications. Requester gateways receive requests from requester applications and forward them through links to responder gateways. Gateways operate colocated with your VPC and provide routing, load balancing, and processing capabilities. You maintain full control over your bidding logic, data, and auction decisions, while RTB Fabric provides the secure infrastructure for connectivity. Requester gateways are typically used by supply-side platforms (SSPs).

Topics

Creating a requester gateway

Create a new requester gateway that can forward bid requests for ad impressions and receive responses.

> **Note**
>
> You are responsible for the data you send through RTB Fabric, including ensuring that personally identifiable information (PII) is handled according to your privacy requirements and applicable regulations.

To create a requester gateway

1. Sign in to the AWS Management Console and open the RTB Fabric console at https://console.aws.amazon.com/rtbfabric.
2. In the navigation pane, choose **Requester gateway**.
3. Choose **Create requester gateway**.
4. In the **Requester gateway information** section, for **Requester gateway description**, enter a description of the gateway's purpose. The description can have up to 255 characters.
5. In the **VPC configuration** section, configure the following settings:
   a. For **VPC ID**, enter the ID of the virtual private cloud (VPC) where you want to connect the requester gateway. The VPC ID must start with "vpc-" followed by either 8 or 17 hexadecimal characters in lowercase. For example: vpc-0123abc4567def890.
   b. For **Subnet ID**, enter the subnet IDs where you want to connect your gateway. Enter up to 5 subnet IDs (format: subnet-0123abc4567def89a), separated by commas. Must be from the specified VPC. These subnets should match your core workload deployment subnets or be secondary CIDR subnets within the same Availability Zones.
   c. For **Security group ID**, enter the IDs of 1-5 security groups, separated by commas. We recommend you create new security groups for your gateway for security.
6. Choose **Create gateway**.
7. Your new requester gateway appears in the gateways list with an **Activating** status. The gateway status will remain **Activating** for 2-5 minutes until creation is complete.

AWS CLI

Use the following command to create a requester gateway using the AWS Command Line Interface (AWS CLI).

```
# Create a requester gateway with required parameters
aws rtbfabric create-requester-gateway \
    --description "My RTB requester gateway" \
    --vpc-id vpc-12345678 \
    --subnet-ids subnet-abc12345 subnet-def67890 \
    --security-group-ids sg-12345678 \
    --client-token "unique-client-token-123"

# Create with optional tags
```

```
aws rtbfabric create-requester-gateway \
    --description "My RTB requester gateway" \
    --vpc-id vpc-12345678 \
    --subnet-ids subnet-abc12345 subnet-def67890 \
    --security-group-ids sg-12345678 \
    --client-token "unique-client-token-123" \
    --tags Environment=Production Team=RTB
```

Updating gateway description

You can update the gateway description using the RTB Fabric API. For more information, see the AWS RTB Fabric API Reference.

Searching for requester gateways

Use the search functionality in the console to locate specific gateways associated with your account. The applications table displays key information including application ID, status, name, creation date, and associated resources.

To search for requester gateways

1. In the **Find requester gateways** search box, enter your search criteria.
2. You can search across requester gateway ID, status, name, or creation date.
3. The table automatically filters to show matching applications as you type.
4. If no gateways exist, the console displays **No requester gateways** with an option to create a gateway.

AWS CLI

Use the following command to get details for a specific requester gateway using the AWS Command Line Interface (AWS CLI).

```
# Get details for a specific requester gateway
aws rtbfabric get-requester-gateway --gateway-id "rtb-gw-req-12345"
```

Viewing associated links

Each requester gateway can have associated links that connect it to responder applications. You can view these links directly from the applications table and see detailed connection information.

To view associated links for an application

1. In the **Requester gateways** table, locate the application whose links you want to view.
2. Select the radio button for the application row.
3. The application details expand below the table, showing the application ID with a collapsible section.
4. In the expanded section, view the **Links associated with this requester gateway** section, which displays the total number of links in parentheses.
5. Review the links table, which shows detailed information for each associated link including link ID, status, creation date, requester application name, and responder application ID.

The links table includes the following columns:

- **Link ID** – Unique identifier for the link.
- **Link status** – Current operational status of the link.
- **Link creation date (UTC)** – When the link was created.
- **Requester gateway name** – Name of the requesting application.
- **Responder Gateway ID** – ID of the responding application.

AWS CLI

Use the following command to list all links associated with a specific requester gateway using the AWS Command Line Interface (AWS CLI).

```
# List all links associated with a gateway
aws rtbfabric list-links --gateway-id "rtb-gw-dsj34i23nsllka"

# List links with pagination
aws rtbfabric list-links --gateway-id "rtb-gw-dsj34i23nsllka" --max-results 10 --next-
token "token"
```

Deleting requester gateways

When you no longer need a requester gateway, you can delete it from your environment. This action is irreversible and will terminate all bidding activities associated with the application.

We recommend deleting unused requester gateways to optimize resource usage and costs. AWS may delete unused gateways after 30 days of inactivity to manage infrastructure resources.

> **Warning**
>
> Deleting a requester gateway is permanent and cannot be undone. Check your gateway metrics to verify there is no active traffic before proceeding with deletion.

To delete a requester gateway

1. On the **Requester gateways** page, select the radio button next to the application you want to delete.
2. Choose **Delete** from the action buttons at the top of the page.
3. If the gateway has associated links, a dialog appears with the message "To delete this application, you must first delete all of its associated links. You can delete links on the Links table." Follow the provided instructions to delete associated links first, then return to delete the application. For more information, see Deleting gateway links.
4. If the application has no associated links, a confirmation dialog appears. Verify that you want to delete the selected application.
5. Choose **Delete** to confirm the deletion.

AWS CLI

Use the following command to delete a requester gateway using the AWS Command Line Interface (AWS CLI).

```
# Delete a requester gateway
aws rtbfabric delete-requester-gateway --gateway-id "rtb-gw-dsj34i23nsllka"
```

Responder gateways

Responder gateways are RTB Fabric infrastructure components that serve as connection points for customer applications. Responder gateways receive requests from requester gateways and forward them to responder applications, then return responses through the same pathway. Gateways operate colocated with your VPC and provide routing, load balancing, and processing capabilities.

You maintain complete control over your bidding algorithms, response logic, and data processing, while RTB Fabric provides the secure infrastructure for connectivity.

Topics

Creating a responder gateway

Create a new responder gateway that can respond to bid opportunities.

> **Note**
>
> You are responsible for the data you process through RTB Fabric, including ensuring that personally identifiable information (PII) is handled according to your privacy requirements and applicable regulations.

To create a responder gateway

1. Sign in to the AWS Management Console and open the RTB Fabric console at https://console.aws.amazon.com/rtbfabric.
2. In the navigation pane, choose **Responder gateway**.
3. Choose **Create responder gateway**.
4. In the **Responder gateway information** section, for **Gateway description**, enter a description of the gateway's purpose. The description can have up to 255 characters.
5. In the **VPC configuration** section, configure the network settings:
    a. For **VPC ID**, enter a valid VPC ID. For example: vpc-01f345ad6524a6d7.
    b. For **Subnet ID**, enter the IDs of 1-5 subnets, separated by commas. Subnets must have at least 200 free IP addresses. These subnets should match your core workload deployment subnets or be secondary CIDR subnets within the same Availability Zones.
    c. For **Security group ID**, enter the IDs of 1-5 security groups, separated by commas. We recommend you create new security groups for your gateways for security.

6. In the **Responder endpoint configuration** section, configure the endpoint where your gateway receives network traffic:

   a. For **DNS name**, enter a fully qualified domain name (FQDN) where you want your gateway to be accessed. Valid characters are a-z, A-Z, 0-9, periods (.), and hyphens (-). Maximum length is 253 characters.

   b. (Optional) For **CA certificate chain**, enter the CA certificate chain for your domain. Include the intermediate and root certificates in PEM format. Maximum size: 2048 characters.

   c. For **Network port number**, enter the network port number where your gateway will listen for incoming traffic. Enter a number from 1 to 65535. Common ports are 80 and 443.

   d. For **Protocol**, choose the web protocol that your gateway will use for communication. Select either **https://** or **http://**.

7. Choose **Create Gateway**.

8. Your new responder gateway appears in the gateways list with an **Activating** status. The gateway status will remain **Activating** for 2-5 minutes until creation is complete.

After creating your application, you can view its details, monitor performance metrics, and make configuration changes as needed.

Updating gateway description

You can update the gateway description using the RTB Fabric API. For more information, see the AWS RTB Fabric API Reference.

AWS CLI

Use the following command to create a responder gateway using the AWS Command Line Interface (AWS CLI).

```
# Create a responder gateway with required parameters
aws rtbfabric create-responder-gateway \
    --description "My RTB responder gateway" \
    --vpc-id vpc-01f345ad6524a6d7 \
    --subnet-ids subnet-abc12345 subnet-def67890 \
    --security-group-ids sg-12345678 \
    --port 443 \
    --protocol HTTPS \
    --client-token "unique-client-token-123"
```

```
# Create with optional domain name and trust store configuration
aws rtbfabric create-responder-gateway \
    --description "My RTB responder gateway" \
    --vpc-id vpc-01f345ad6524a6d7 \
    --subnet-ids subnet-abc12345 subnet-def67890 \
    --security-group-ids sg-12345678 \
    --domain-name responder.example.com \
    --port 443 \
    --protocol HTTPS \
    --client-token "unique-client-token-123" \
    --trust-store-configuration certificateAuthorityCertificates="-----BEGIN
 CERTIFICATE-----..." \
    --tags Environment=Production Team=RTB
```

Logging

When logging is configured, default sampling behavior applies. Service logs capture all error logs (`error_log` sampling rate of 1) and no filter logs (`filter_log` sampling rate of 0). To modify sampling rates after creation, see UpdateLink in the *AWS RTB Fabric API Reference*.

Searching for responder gateways

Use the search functionality in the console to locate specific gateways in your environment. The gateways table displays key information including gateway ID, status, name, associated links, and creation date.

To search for responder gateways

1. Sign in to the AWS Management Console and open the RTB Fabric console at https://console.aws.amazon.com/rtbfabric.
2. In the navigation pane, choose **Responder gateways**.
3. In the **Find responder gateways** search box, enter your search criteria to locate specific applications.
4. The table automatically filters to show matching applications as you type.
5. If no applications exist, the console displays **No responder gateways** with an option to create your first application.

AWS CLI

Use the following command to get details for a specific responder gateway using the AWS Command Line Interface (AWS CLI).

```
# Get details for a specific responder gateway
aws rtbfabric get-responder-gateway --gateway-id "rtb-gw-kasoi29asfdhn"
```

Viewing associated links

Each responder gateway can have associated links that connect it to requester gateways. You can view these links and their details through the console.

To view associated links for a responder application

1. On the **Responder gateways** page, select the radio button next to the responder gateway you want to view.
2. Choose **View details** to see comprehensive information about the application, including its configuration, status, and associated resources.
3. Choose the **Associated links** tab to view existing links and their details.

AWS CLI

Use the following command to list all links associated with a specific responder gateway using the AWS Command Line Interface (AWS CLI).

```
# List all links associated with a gateway
aws rtbfabric list-links --gateway-id "rtb-gw-dsj34i23nsllka"

# List links with pagination
aws rtbfabric list-links --gateway-id "rtb-gw-dsj34i23nsllka" --max-results 10 --next-
token "token"
```

Deleting responder gateways

When you no longer need a responder gateway, you can delete it from your environment. This action is irreversible and will terminate all bidding activities associated with the application.

We recommend deleting unused responder gateways to optimize resource usage and costs. AWS may delete unused gateways after 30 days of inactivity to manage infrastructure resources.

> **Warning**
>
> Deleting a responder gateway is permanent and cannot be undone. Check your gateway metrics to verify there is no active traffic before proceeding with deletion.

To delete a responder gateway

1. On the **Responder gateways** page, select the radio button next to the responder gateway you want to delete.
2. Choose **Delete** from the action buttons at the top of the page.
3. If the application has associated links, a dialog appears with the message "To delete this application, you must first delete all of its associated links. You can delete links on the Links table." Follow the provided instructions to delete associated links first, then return to delete the application. For more information, see Deleting gateway links.
4. If the application has no associated links, confirm the deletion when prompted.

AWS CLI

Use the following command to delete a responder gateway using the AWS Command Line Interface (AWS CLI).

```
# Delete a responder gateway
aws rtbfabric delete-responder-gateway --gateway-id "rtb-gw-kasoi29asfdhn"
```

Managed endpoints

Managed endpoints is an optional feature for responder gateways that allows RTB Fabric to distribute load directly across bidder hosts in your fleet. This feature bypasses the need for a separate load balancer by using RTB Fabric to send traffic directly to your responder application hosts.

Managed endpoints supports two infrastructure types for hosting your responder applications:

- **Amazon Elastic Kubernetes Service (Amazon EKS) clusters** – RTB Fabric integrates with your EKS cluster to send traffic to your bidder application pods

- **Amazon Elastic Compute Cloud (Amazon EC2) Auto Scaling groups** – RTB Fabric uses information from your Auto Scaling groups to determine the set of IP addresses to send traffic to

If you are interested in using managed endpoints for your responder gateway, please contact your AWS solution architect (SA).

IAM role requirements

To use managed endpoints, you must provide an IAM role that RTB Fabric can assume to interact with your infrastructure. RTB Fabric uses a service-linked role for most operations, but requires this additional role specifically for managed endpoint functionality.

> **Important**
>
> All managed endpoint IAM roles must include the tag `RTBFabricManagedEndpoint` with any value. RTB Fabric requires this tag to assume the role for security purposes. Roles without this tag cannot be used for managed endpoints.

For Auto Scaling group managed endpoints, the IAM role must include the following permissions:

- `autoscaling:DescribeAutoScalingGroups`
- `ec2:DescribeInstanceStatus`
- `ec2:DescribeInstances`
- `ec2:DescribeAvailabilityZones`

The role must also include a trust relationship that allows RTB Fabric to assume it:

```
{
 "Version": "2012-10-17",
 "Statement": [
   {
     "Effect": "Allow",
     "Principal": {
       "Service": "rtbfabric.amazonaws.com"
     },
     "Action": "sts:AssumeRole"
   }
```

```
 ]
}
```

When creating the IAM role, ensure that you add the required tag:

- **Tag key**: `RTBFabricManagedEndpoint`
- **Tag value**: Any value (for example, `true` or `enabled`)

Configuration requirements

The configuration requirements vary depending on your infrastructure type.

Auto Scaling groups configuration

For Auto Scaling group managed endpoints, you must provide the following configuration:

- **autoScalingGroupNames** – The names of the Auto Scaling groups where the instances responding to RTB bid requests belong to.
- **roleArn** – The ARN of an IAM role allowing RTB Fabric to query the Auto Scaling groups in `autoScalingGroupNames` for the instances to send traffic to.

The IAM role must allow the service `rtbfabric.amazonaws.com` in its trust policy:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "rtbfabric.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

The role must also allow the following permissions in its permissions policies:

```
{
   "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "EksEndpointsIpDiscovery",
            "Effect": "Allow",
            "Action": [
                "autoscaling:DescribeAutoScalingGroups",
                "ec2:DescribeInstanceStatus",
                "ec2:DescribeInstances",
                "ec2:DescribeAvailabilityZones"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "ec2:Region": "us-east-1"
                }
            }
        }
    ]
}
```

EKS endpoints configuration

For EKS managed endpoints, you must provide the following configuration:

- **roleArn** – The ARN of an IAM role allowing RTB Fabric to query the target IPs of EKS cluster to send traffic to.

The IAM role must allow the service `rtbfabric.amazonaws.com` in its trust policy:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "rtbfabric.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
```

```
        }
    ]
}
```

The role does not need to have any IAM policies attached to it, but must be associated with EKS cluster's RBAC to authorize RTB Fabric to discover IP targets in the cluster:

```
---
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: heimdall-endpoints-role
  namespace: default
rules:
  - apiGroups: [""]
    resources: ["endpoints"]
    resourceNames: ["nginx-deployment"]
    verbs: ["get"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: heimdall-endpoints-rolebinding
  namespace: default
subjects:
  - kind: User
    name: heimdall-integration
    apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: Role
  name: heimdall-endpoints-role
  apiGroup: rbac.authorization.k8s.io
---
apiVersion: v1
kind: ConfigMap
metadata:
  name: aws-auth
  namespace: kube-system
data:
  mapRoles: |
    - rolearn: arn:aws:iam::242201309515:role/
RtbFabricRoleForEksEndpointsManagedEndpoint
      username: heimdall-integration
```

HTTPS considerations

RTB Fabric can terminate the TLS connection from the requester on your behalf and send your hosts HTTP traffic. However, if you require HTTPS from RTB Fabric managed endpoints to your bidder hosts, additional configuration is required:

- **TLS certificates** – Each host must serve up a TLS certificate.
- **Certificate Authority chain** – You must provide RTB Fabric with the Certificate Authority (CA) certificate chain so that RTB Fabric hosts can trust the TLS certificate from each bidder host.
- **DNS name allowlisting** – You must provide RTB Fabric with a DNS name that matches the SAN of the TLS certificate from each bidder host. The DNS name must be allowlisted by the RTB Fabric team for your account before you can create your RTB application.

Links

RTB Fabric links establish secure connections between your requester gateways and responder gateways in your real-time bidding infrastructure. Links can be between RTB Fabric users, or between an RTB Fabric user and an external entity (external links). These links enable seamless data flow and communication between gateways, allowing your applications to build comprehensive bidding workflows through RTB Fabric infrastructure.

> **Note**
>
> To create links with other RTB Fabric users, you need their responder gateway ID. Contact your RTB Fabric partners directly to exchange gateway IDs. AWS does not provide gateway IDs for external partners, so coordinate directly with your business partners to obtain this information.

Topics

Creating links between gateways

You can create links between RTB Fabric gateways through the AWS Management Console. Only requester gateways can initiate link creation to connect with responder gateways in the system.

To create a link between gateways

1. Sign in to the AWS Management Console and open the RTB Fabric console at https://console.aws.amazon.com/rtbfabric/.
2. In the navigation pane, choose **Requester gateways**.

3. Select an RTB application from the list.
4. On the application details page, choose the **Associated links** tab.
5. Choose **Create link**.
6. On the **Create link** screen, review the **Gateway details** section, which displays information about the source gateway for this link:

   - **Gateway ID** – The unique identifier of the source gateway.
   - **Gateway name** – The name of the source gateway.
   - **Gateway created on** – The date and time when the gateway was created.
7. (Optional) In the **Link information** section, enter a **Correlation ID**. This is a unique identifier you can assign to your link for your own tracking purposes and is not visible to other RTB Fabric users. The correlation ID can have up to 64 characters.
8. In the **Application logs configuration** section, configure the sampling rates to capture exceptions, failures, and unexpected system behaviors:

   a. For **Error logs sampling rate**, enter the percentage (0.0-100.0) of error logs to deliver to your destination. These logs capture exceptions, failures, and unexpected system behaviors. Higher percentages incur additional storage costs.

   b. For **Filter logs sampling rate**, enter the percentage (0.0-100.0) of filter logs to deliver to your destination. These logs are generated from your other RTB Fabric filter modules. Higher percentages incur additional storage costs.

   Note the following:

   - AWS does not access or read your log data.
   - Range must be from 0.0-100.0.
   - To configure log delivery destinations, you must use the RTB Fabric API. For more information, see the AWS RTB Fabric API Reference.
9. In the **Target details** section, enter the **Target gateway ID** of the target gateway you want to link with. Enter a valid gateway ID (for example: **`rtb-gw-q6jlbximmmnaz1q5676wots4xy`**).

   > **Note**
   >
   > Contact your RTB Fabric partner to obtain their gateway ID. AWS does not provide gateway IDs.

10. Choose **Create link** to send the link request.
11. The link is created with a **Requested** status and sent to the target gateway for approval. The target gateway owner must accept the link request before it becomes active.

Once the target application accepts the link request, the link status changes to **Active** and begins facilitating communication between your RTB applications. You can monitor link performance and make configuration changes as needed.

AWS CLI

Use the following command to create a link between gateways using the AWS Command Line Interface (AWS CLI).

```
# Create a basic link between gateways
aws rtbfabric create-link \
    --gateway-id rtb-gw-source123 \
    --peer-gateway-id rtb-gw-target456 \
    --log-settings '{
        "applicationLogs": {
            "sampling": {
                "errorLog": 100.0,
                "filterLog": 0.0
            }
        }
    }'

# Create a link with customer-provided ID and tags
aws rtbfabric create-link \
    --gateway-id rtb-gw-source123 \
    --peer-gateway-id rtb-gw-target456 \
    --attributes customerProvidedId=my-link-correlation-123 \
    --log-settings '{
        "applicationLogs": {
            "sampling": {
                "errorLog": 100.0,
                "filterLog": 0.0
            }
        }
    }' \
    --tags Environment=Production Team=RTB
```

Creating external links

External links enable connectivity between your RTB Fabric gateways and external partners over the public internet, extending your RTB infrastructure beyond private VPC connections. This feature supports integration with external supply-side platforms (SSPs), demand-side platforms (DSPs), and other RTB partners who are not using RTB Fabric infrastructure.

External links differ from standard RTB Fabric links in several key ways:

- **Public internet connectivity** – Traffic flows over the public internet rather than private AWS network infrastructure.
- **Client IP preservation** – In inbound external links, the original client IP addresses are preserved, enabling DSPs to implement IP-based filtering and geographic targeting.
- **Opt-in feature** – External link capability must be explicitly enabled for your account. Contact AWS support to request access to external link functionality.
- **API-only creation** – External links can only be created and managed through the RTB Fabric API, not through the console.

Use inbound external links to receive traffic from external partners, and outbound external links to send traffic to external endpoints. You can only create external links using the RTB Fabric API. For more information, see CreateInboundExternalLink and CreateOutboundExternalLink in the *RTB Fabric API Reference*.

Creating inbound external links

Inbound external links allow external partners to send traffic to your responder gateway over the public internet. When you create an inbound external link, RTB Fabric provides a public domain name that external partners can use to reach your gateway. The following examples show how to create an inbound external link using the RTB Fabric API. For complete specifications, see CreateInboundExternalLink in the *RTB Fabric API Reference*.

API request

The following example shows the API request to create an inbound external link:

```
POST /responder-gateway/{gatewayId}/inbound-external-link
```

```
{
  "attributes": {
    "customerProvidedId": "external-partner-link-001"
  }
}
```

Example using curl

The following example shows how to create an inbound external link using curl:

```
curl -X POST \
  "https://rtbfabric.us-east-1.amazonaws.com/responder-gateway/rtb-gw-abc123/inbound-
external-link" \
  -H "Authorization: AWS4-HMAC-SHA256 ..." \
  -H "Content-Type: application/json" \
  -d '{
    "attributes": {
      "customerProvidedId": "external-partner-link-001"
    }
  }'
```

Response

The following example shows the response from creating an inbound external link:

```
{
  "gatewayId": "rtb-gw-abc123",
  "linkId": "link-xyz789",
  "status": "Active",
  "domainName": "abc123.rtbfabric.amazonaws.com"
}
```

The `domainName` in the response is the public endpoint that external partners should use to send traffic to your responder gateway.

RTB Fabric sets a DNS TTL (time to live) of 60 seconds for provided domain names. External partners should configure their DNS clients to respect this TTL value to ensure proper failover and load balancing behavior.

Creating outbound external links

Outbound external links allow your requester gateway to send traffic to external partner endpoints over the public internet. You must provide the public HTTPS endpoint URL of the external responder.

API request

The following example shows the API request to create an outbound external link:

```
POST /requester-gateway/{gatewayId}/outbound-external-link

{
  "publicEndpoint": "https://external-partner.com/bid-endpoint"
}
```

Example using curl

The following example shows how to create an outbound external link using curl:

```
curl -X POST \
  "https://rtbfabric.us-east-1.amazonaws.com/requester-gateway/rtb-gw-def456/outbound-
external-link" \
  -H "Authorization: AWS4-HMAC-SHA256 ..." \
  -H "Content-Type: application/json" \
  -d '{
    "publicEndpoint": "https://external-partner.com/bid-endpoint"
  }'
```

Response

The following example shows the response from creating an outbound external link:

```
{
  "gatewayId": "rtb-gw-def456",
  "linkId": "link-uvw123",
  "status": "Active"
}
```

Once created, your requester gateway can send traffic to the specified `publicEndpoint` through the RTB Fabric infrastructure.

Important

External links operate over the public internet and may have different security, performance, and cost characteristics compared to internal links. Ensure that your external endpoints support HTTPS and follow security best practices for public internet communication.

Editing links

You can only edit existing links using the API. To modify a link created in the console, you must delete the existing link and create a new one with the updated configuration. For more information, see UpdateLink in the *RTB Fabric API Reference*.

Accepting or declining a link request

When another RTB application sends you a link request, you can accept or reject it from your application's **Associated links** tab. Link requests allow other applications to connect and send bid requests to your application.

To accept or reject a link request

1. Sign in to the AWS Management Console and open the RTB Fabric console at https://console.aws.amazon.com/rtbfabric.
2. Choose either **Requester Gateways** or **Responder Gateways** depending on which application received the link request.
3. Choose the RTB application that received the link request.
4. Choose the **Associated links** tab to view pending link requests.
5. In the **Links** section, locate the link request you want to respond to in the links table. The table displays the following information:

   - **Link ID** – Unique identifier for the link.
   - **Link status** – Current status of the link request.
   - **Link creation date (UTC)** – When the request was created.
   - **Responder Gateway ID** – Target application ID.
   - **Actions** – Available actions for the link.

6. In the **Actions** column for the link request, choose one of the following:
   - **To accept the link request:**
     a. Choose **Accept**.
     b. In the confirmation dialog, choose **Accept** to confirm.
     c. The link status changes to **Active** and bid requests can now flow between the applications.
   - **To reject the link request:**
     a. Choose **Reject**.
     b. In the confirmation dialog, choose **Reject** to confirm.
     c. The link status changes to **Rejected** and the connection request is rejected.

Once you accept a link request, the requesting application can send bid requests to your application. You can view link metrics and manage the connection from the **Associated links** tab.

> **Note**
>
> You can also choose **View details** to see more information about the link request before making your decision. The links table also includes action buttons for **Accept**, **Reject**, and **Delete** operations.

AWS CLI

Use the following commands to accept or reject link requests using the AWS Command Line Interface (AWS CLI).

```
# Accept a link request
aws rtbfabric accept-link \
    --gateway-id "rtb-gw-kasoi29asfdhn" \
    --link-id "link-sedf903ujiose"

# Reject a link request
aws rtbfabric reject-link \
    --gateway-id "rtb-gw-kasoi29asfdhn" \
    --link-id "link-sedf903ujiose"
```

Deleting gateway links

When you no longer need a link between RTB applications, you can delete it from either the requester or responder application. This action permanently removes the connection and stops all communication between the linked applications.

> **Warning**
>
> Deleting a link is irreversible and will immediately stop all bid request processing between the connected RTB applications. Ensure that you no longer need the link before proceeding.

To delete an RTB application link

1. Navigate to the RTB applications service and choose either **Requester Gateways** or **Responder Gateways** depending on which application owns the link you want to delete.
2. Access the link details using one of these methods:
   - On the applications page, select the radio button next to the desired application. In the **Links associated with this RTB application** section at the bottom of the page, choose the **Link ID** you want to delete.
   - Choose the **RTB application ID** to open the application details page. Choose the **Associated links** tab and choose the **Link ID** you want to delete.
3. On the link details page, review the link information including:
   - **Link ID** – Unique identifier for the link.
   - **Link status** – Current status (Requested, Active, etc.).
   - **RTB application ID** – Source application.
   - **Peer RTB application ID** – Target application.
   - **Link created on** and **Link updated on** – Timestamps.
4. Choose **Delete link** from the action buttons at the top of the page.
5. In the confirmation dialog, verify that you want to delete the selected link.
6. Choose **Delete** to confirm the deletion.

The link is immediately removed from both RTB applications and will no longer facilitate communication between them. Any ongoing bid requests using this link will be terminated.

AWS CLI

Use the following command to delete a link between RTB applications using the AWS Command Line Interface (AWS CLI).

```
# Delete a link from a gateway
aws rtbfabric delete-link \
    --gateway-id "rtb-gw-kasoi29asfdhn" \
    --link-id "link-sedf903ujiose"
```

Adding modules to links

You can only add modules to existing links using the RTB Fabric API. Modules enable you to implement rate limiting, filtering, error handling, and other traffic management capabilities on your links. For more information, see Modules.

Configuring link logging

You can configure individual links to deliver logs to CloudWatch Logs using the RTB Fabric API. Before configuring link logging, you must first set up log delivery infrastructure. For more information about setting up log delivery destinations, see UpdateLink in the *RTB Fabric API Reference* and Configuring RTB Fabric logs with Amazon CloudWatch Logs.

Modules

Modules are configurable components that process RTB traffic flowing through your links. You can use modules to implement rate limiting, filtering, error handling, and other traffic management capabilities. RTB Fabric provides built-in modules that are available at no additional charge, and you can configure them using the RTB Fabric API.

On a link details page, choose the **Modules** tab to view information about modules that have been configured for the current link. RTB Fabric only supports viewing modules in the console. You can configure modules with the RTB Fabric API using the `UpdateLinkModuleFlow` operation.

Topics

Built-in modules

RTB Fabric provides the following built-in modules that you can configure for your links:

- **QPS rate limiter** – Controls the rate of requests flowing through the link by limiting queries per second (QPS). This module helps protect downstream systems from traffic spikes and ensures consistent performance under varying load conditions.
- **OpenRTB filter** – Filters RTB requests and responses based on OpenRTB protocol specifications. This module validates message formats, removes invalid fields, and ensures compliance with OpenRTB standards.
- **Error masker** – Masks sensitive information in error responses to prevent data leakage while maintaining debugging capabilities. This module helps protect confidential data when errors occur during RTB processing.

All built-in modules are available at no additional charge. You can configure multiple modules on a single link, and they will be applied in the order you specify during configuration.

Configuring modules

Module configuration is only available through the RTB Fabric API. You cannot configure modules using the RTB Fabric console. Use the UpdateLinkModuleFlow API operation to add, modify, or

remove modules from your links. This operation allows you to define the processing flow for RTB traffic by specifying which modules to apply, their configuration parameters, and the order in which they execute.

Example: Adding a QPS rate limiter module

The following example shows how to add a QPS rate limiter module to a link using the AWS Command Line Interface (AWS CLI).

```
# Add a QPS rate limiter module to a link
aws rtbfabric update-link-module-flow \
    --gateway-id rtb-gw-source123 \
    --link-id link-abc456def \
    --client-token "unique-update-token-789" \
    --modules '[
        {
            "name": "rate-limiter-module",
            "version": "1.0.0",
            "dependsOn": [],
            "moduleParameters": {
                "rateLimiter": {
                    "tps": 1000.0
                }
            }
        }
    ]'
```

For detailed information about module configuration parameters and API usage, see the AWS RTB Fabric API Reference.

Security in AWS RTB Fabric

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from data centers and network architectures that are built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The shared responsibility model describes this as security *of* the cloud and security *in* the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the AWS Compliance Programs. To learn about the compliance programs that apply to RTB Fabric, see AWS Services in Scope by Compliance Program.
- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your company's requirements, and applicable laws and regulations.

> **Important**
>
> **Security Risk:** RTB Fabric supports creating links without TLS encryption or authentication. Disabling TLS encryption allows any actor with access to the data in transit to view, tamper with, or spoof the data in transit. Disabling authentication allows anyone with network access to the endpoint to submit RTB requests. Always enable TLS and authentication for production environments.

This documentation helps you understand how to apply the shared responsibility model when using RTB Fabric. The following topics show you how to configure RTB Fabric to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your RTB Fabric resources.

Topics

- Compliance validation for AWS RTB Fabric

Identity and access management for AWS RTB Fabric

AWS Identity and Access Management (IAM) is an AWS service that helps an administrator securely control access to AWS resources. IAM administrators control who can be *authenticated* (signed in) and *authorized* (have permissions) to use RTB Fabric resources. IAM is an AWS service that you can use with no additional charge.

Topics

- Audience
- Authenticating with identities
- Managing access using policies
- How RTB Fabric works with IAM
- Identity-based policy examples for RTB Fabric
- AWS managed policies for AWS RTB Fabric
- Troubleshooting RTB Fabric identity and access
- Using service-linked roles for RTB Fabric

Audience

How you use AWS Identity and Access Management (IAM) differs based on your role:

- **Service user** - request permissions from your administrator if you cannot access features (see Troubleshooting RTB Fabric identity and access)
- **Service administrator** - determine user access and submit permission requests (see How RTB Fabric works with IAM)
- **IAM administrator** - write policies to manage access (see Identity-based policy examples for RTB Fabric)

Authenticating with identities

Authentication is how you sign in to AWS using your identity credentials. You must be authenticated as the AWS account root user, an IAM user, or by assuming an IAM role.

You can sign in as a federated identity using credentials from an identity source like AWS IAM Identity Center (IAM Identity Center), single sign-on authentication, or Google/Facebook credentials. For more information about signing in, see How to sign in to your AWS account in the *AWS Sign-In User Guide*.

For programmatic access, AWS provides an SDK and CLI to cryptographically sign requests. For more information, see AWS Signature Version 4 for API requests in the *IAM User Guide*.

AWS account root user

When you create an AWS account, you begin with one sign-in identity called the AWS account *root user* that has complete access to all AWS services and resources. We strongly recommend that you don't use the root user for everyday tasks. For tasks that require root user credentials, see Tasks that require root user credentials in the *IAM User Guide*.

Federated identity

As a best practice, require human users to use federation with an identity provider to access AWS services using temporary credentials.

A *federated identity* is a user from your enterprise directory, web identity provider, or AWS Directory Service that accesses AWS services using credentials from an identity source. Federated identities assume roles that provide temporary credentials.

For centralized access management, we recommend AWS IAM Identity Center. For more information, see What is IAM Identity Center? in the *AWS IAM Identity Center User Guide*.

IAM users and groups

An *IAM user* is an identity with specific permissions for a single person or application. We recommend using temporary credentials instead of IAM users with long-term credentials. For more information, see Require human users to use federation with an identity provider to access AWS using temporary credentials in the *IAM User Guide*.

An *IAM group* specifies a collection of IAM users and makes permissions easier to manage for large sets of users. For more information, see Use cases for IAM users in the *IAM User Guide*.

IAM roles

An *IAM role* is an identity with specific permissions that provides temporary credentials. You can assume a role by switching from a user to an IAM role (console) or by calling an AWS CLI or AWS API operation. For more information, see Methods to assume a role in the *IAM User Guide*.

IAM roles are useful for federated user access, temporary IAM user permissions, cross-account access, cross-service access, and applications running on Amazon EC2. For more information, see Cross account resource access in IAM in the *IAM User Guide*.

Managing access using policies

You control access in AWS by creating policies and attaching them to AWS identities or resources. A policy defines permissions when associated with an identity or resource. AWS evaluates these policies when a principal makes a request. Most policies are stored in AWS as JSON documents. For more information about JSON policy documents, see Overview of JSON policies in the *IAM User Guide*.

Using policies, administrators specify who has access to what by defining which **principal** can perform **actions** on what **resources**, and under what **conditions**.

By default, users and roles have no permissions. An IAM administrator creates IAM policies and adds them to roles, which users can then assume. IAM policies define permissions regardless of the method used to perform the operation.

Identity-based policies

Identity-based policies are JSON permissions policy documents that you attach to an identity (user, group, or role). These policies control what actions identities can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Define custom IAM permissions with customer managed policies in the *IAM User Guide*.

Identity-based policies can be *inline policies* (embedded directly into a single identity) or *managed policies* (standalone policies attached to multiple identities). To learn how to choose between managed and inline policies, see Choose between managed policies and inline policies in the *IAM User Guide*.

Resource-based policies

Resource-based policies are JSON policy documents that you attach to a resource. Examples include IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-

based policies, service administrators can use them to control access to a specific resource. You must specify a principal in a resource-based policy.

Resource-based policies are inline policies that are located in that service. You can't use AWS managed policies from IAM in a resource-based policy.

Other policy types

AWS supports additional policy types that can set the maximum permissions granted by more common policy types:

- **Permissions boundaries** – Set the maximum permissions that an identity-based policy can grant to an IAM entity. For more information, see Permissions boundaries for IAM entities in the *IAM User Guide*.
- **Service control policies (SCPs)** – Specify the maximum permissions for an organization or organizational unit in AWS Organizations. For more information, see Service control policies in the *AWS Organizations User Guide*.
- **Resource control policies (RCPs)** – Set the maximum available permissions for resources in your accounts. For more information, see Resource control policies (RCPs) in the *AWS Organizations User Guide*.
- **Session policies** – Advanced policies passed as a parameter when creating a temporary session for a role or federated user. For more information, see Session policies in the *IAM User Guide*.

Multiple policy types

When multiple types of policies apply to a request, the resulting permissions are more complicated to understand. To learn how AWS determines whether to allow a request when multiple policy types are involved, see Policy evaluation logic in the *IAM User Guide*.

How RTB Fabric works with IAM

Before you use IAM to manage access to RTB Fabric, learn what IAM features are available to use with RTB Fabric.

IAM feature	RTB Fabric support
Identity-based policies	Yes

IAM feature	RTB Fabric support
Resource-based policies	No
Policy actions	Yes
Policy resources	Yes
Policy condition keys	Yes
ACLs	No
ABAC (tags in policies)	No
Temporary credentials	Yes
Principal permissions	Yes
Service roles	Yes
Service-linked roles	Yes

To get a high-level view of how RTB Fabric and other AWS services work with most IAM features, see AWS services that work with IAM in the *IAM User Guide*.

Identity-based policies for RTB Fabric

Supports identity-based policies: Yes

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Define custom IAM permissions with customer managed policies in the *IAM User Guide*.

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions under which actions are allowed or denied. To learn about all of the elements that you can use in a JSON policy, see IAM JSON policy elements reference in the *IAM User Guide*.

Identity-based policy examples for RTB Fabric

To view examples of RTB Fabric identity-based policies, see Identity-based policy examples for RTB Fabric.

Resource-based policies within RTB Fabric

Supports resource-based policies: No

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must specify a principal in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

To enable cross-account access, you can specify an entire account or IAM entities in another account as the principal in a resource-based policy. For more information, see Cross account resource access in IAM in the *IAM User Guide*.

Policy actions for RTB Fabric

Supports policy actions: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Action` element of a JSON policy describes the actions that you can use to allow or deny access in a policy. Include actions in a policy to grant permissions to perform the associated operation.

To see a list of RTB Fabric actions, see Actions, resources, and condition keys for RTB Fabric in the *Service Authorization Reference*.

Policy actions in RTB Fabric use the following prefix before the action:

```
rtbfabric
```

To specify multiple actions in a single statement, separate them with commas.

```
"Action": [
```

```
      "rtbfabric:action1",
      "rtbfabric:action2"
         ]
```

To view examples of RTB Fabric identity-based policies, see Identity-based policy examples for RTB Fabric.

Policy resources for RTB Fabric

Supports policy resources: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Resource` JSON policy element specifies the object or objects to which the action applies. As a best practice, specify a resource using its Amazon Resource Name (ARN). For actions that don't support resource-level permissions, use a wildcard (*) to indicate that the statement applies to all resources.

```
"Resource": "*"
```

To see a list of RTB Fabric resource types and their ARNs, see Actions, resources, and condition keys for RTB Fabric in the *Service Authorization Reference*. To learn with which actions you can specify the ARN of each resource, see Actions, resources, and condition keys for RTB Fabric.

To view examples of RTB Fabric identity-based policies, see Identity-based policy examples for RTB Fabric.

Policy condition keys for RTB Fabric

Supports service-specific policy condition keys: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Condition` element specifies when statements execute based on defined criteria. You can create conditional expressions that use condition operators, such as equals or less than, to match

the condition in the policy with values in the request. To see all AWS global condition keys, see AWS global condition context keys in the *IAM User Guide*.

To see a list of RTB Fabric condition keys, see Actions, resources, and condition keys for RTB Fabric in the *Service Authorization Reference*. To learn with which actions and resources you can use a condition key, see Actions, resources, and condition keys for RTB Fabric.

To view examples of RTB Fabric identity-based policies, see Identity-based policy examples for RTB Fabric.

ACLs in RTB Fabric

Supports ACLs: No

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format.

ABAC with RTB Fabric

Supports ABAC (tags in policies): No

Attribute-based access control (ABAC) is an authorization strategy that defines permissions based on attributes called tags. You can attach tags to IAM entities and AWS resources, then design ABAC policies to allow operations when the principal's tag matches the tag on the resource.

To control access based on tags, you provide tag information in the condition element of a policy using the `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, or `aws:TagKeys` condition keys.

If a service supports all three condition keys for every resource type, then the value is **Yes** for the service. If a service supports all three condition keys for only some resource types, then the value is **Partial**.

For more information about ABAC, see Define permissions with ABAC authorization in the *IAM User Guide*. To view a tutorial with steps for setting up ABAC, see Use attribute-based access control (ABAC) in the *IAM User Guide*.

Using temporary credentials with RTB Fabric

Supports temporary credentials: Yes

Temporary credentials provide short-term access to AWS resources and are automatically created when you use federation or switch roles. AWS recommends that you dynamically generate temporary credentials instead of using long-term access keys. For more information, see Temporary security credentials in IAM and AWS services that work with IAM in the *IAM User Guide*.

Cross-service principal permissions for RTB Fabric

Supports forward access sessions (FAS): Yes

Forward access sessions (FAS) use the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. For policy details when making FAS requests, see Forward access sessions.

Service roles for RTB Fabric

Supports service roles: Yes

A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see Create a role to delegate permissions to an AWS service in the *IAM User Guide*.

> **Warning**
>
> Changing the permissions for a service role might break RTB Fabric functionality. Edit service roles only when RTB Fabric provides guidance to do so.

Service-linked roles for RTB Fabric

Supports service-linked roles: Yes

A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.

For details about creating or managing service-linked roles, see AWS services that work with IAM. Find a service in the table that includes a `Yes` in the **Service-linked role** column. Choose the **Yes** link to view the service-linked role documentation for that service.

Identity-based policy examples for RTB Fabric

By default, users and roles don't have permission to create or modify RTB Fabric resources. To grant users permission to perform actions on the resources that they need, an IAM administrator can create IAM policies.

To learn how to create an IAM identity-based policy by using these example JSON policy documents, see Create IAM policies (console) in the *IAM User Guide*.

For details about actions and resource types defined by RTB Fabric, including the format of the ARNs for each of the resource types, see Actions, Resources, and Condition Keys for AWS RTB Fabric in the *Service Authorization Reference*.

Topics

Policy best practices

Identity-based policies determine whether someone can create, access, or delete RTB Fabric resources in your account. These actions can incur costs for your AWS account. When you create or edit identity-based policies, follow these guidelines and recommendations:

- **Get started with AWS managed policies and move toward least-privilege permissions** – To get started granting permissions to your users and workloads, use the *AWS managed policies* that grant permissions for many common use cases. They are available in your AWS account. We recommend that you reduce permissions further by defining AWS customer managed policies that are specific to your use cases. For more information, see AWS managed policies or AWS managed policies for job functions in the *IAM User Guide*.
- **Apply least-privilege permissions** – When you set permissions with IAM policies, grant only the permissions required to perform a task. You do this by defining the actions that can be taken on specific resources under specific conditions, also known as *least-privilege permissions*. For more

information about using IAM to apply permissions, see Policies and permissions in IAM in the *IAM User Guide*.

- **Use conditions in IAM policies to further restrict access** – You can add a condition to your policies to limit access to actions and resources. For example, you can write a policy condition to specify that all requests must be sent using SSL. You can also use conditions to grant access to service actions if they are used through a specific AWS service, such as AWS CloudFormation. For more information, see IAM JSON policy elements: Condition in the *IAM User Guide*.
- **Use IAM Access Analyzer to validate your IAM policies to ensure secure and functional permissions** – IAM Access Analyzer validates new and existing policies so that the policies adhere to the IAM policy language (JSON) and IAM best practices. IAM Access Analyzer provides more than 100 policy checks and actionable recommendations to help you author secure and functional policies. For more information, see Validate policies with IAM Access Analyzer in the *IAM User Guide*.
- **Require multi-factor authentication (MFA)** – If you have a scenario that requires IAM users or a root user in your AWS account, turn on MFA for additional security. To require MFA when API operations are called, add MFA conditions to your policies. For more information, see Secure API access with MFA in the *IAM User Guide*.

For more information about best practices in IAM, see Security best practices in IAM in the *IAM User Guide*.

Using the RTB Fabric console

To access the AWS RTB Fabric console, you must have a minimum set of permissions. These permissions must allow you to list and view details about the RTB Fabric resources in your AWS account. If you create an identity-based policy that is more restrictive than the minimum required permissions, the console won't function as intended for entities (users or roles) with that policy.

You don't need to allow minimum console permissions for users that are making calls only to the AWS CLI or the AWS API. Instead, allow access to only the actions that match the API operation that they're trying to perform.

To ensure that users and roles can still use the RTB Fabric console, also attach the RTB Fabric `ReadOnly` permissions to the entities. For more information, see Adding permissions to a user in the *IAM User Guide*.

Users who need console access require the following permissions:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "rtbfabric:GetRequesterGateway",
                "rtbfabric:GetResponderGateway",
                "rtbfabric:ListRequesterGateways",
                "rtbfabric:ListResponderGateways",
                "rtbfabric:GetLink",
                "rtbfabric:ListLinks",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeSubnets",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeVpcs"
            ],
            "Resource": "*"
        }
    ]
}
```

Allow users to view their own permissions

This example shows how you might create a policy that allows IAM users to view the inline and managed policies that are attached to their user identity. This policy includes permissions to complete this action on the console or programmatically using the AWS CLI or AWS API.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
```

```
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Basic RTB Fabric permissions

This example shows a policy that allows basic RTB Fabric operations including creating, viewing, and managing RTB applications and links.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "rtbfabric:CreateRequesterGateway",
                "rtbfabric:CreateResponderGateway",
                "rtbfabric:DeleteRequesterGateway",
                "rtbfabric:DeleteResponderGateway",
                "rtbfabric:GetRequesterGateway",
                "rtbfabric:GetResponderGateway",
                "rtbfabric:ListRequesterGateways",
                "rtbfabric:ListResponderGateways",
                "rtbfabric:CreateLink",
                "rtbfabric:DeleteLink",
                "rtbfabric:GetLink",
                "rtbfabric:ListLinks",
                "rtbfabric:AcceptLink",
                "rtbfabric:RejectLink"
            ],
            "Resource": [
                "arn:aws:rtbfabric:*:*:gateway/*",
                "arn:aws:rtbfabric:*:*:link/*"
            ]
        }
    ]
}
```

This policy grants permissions to perform common RTB Fabric operations on RTB applications and links in any region within your AWS account.

RTB Fabric administrator permissions

This example shows a policy that allows full administrative access to RTB Fabric, including the ability to view network interfaces managed by the service. For additional security, consider scoping the CloudWatch Get actions to specific metric resources rather than using wildcard (*) resources, depending on your monitoring requirements.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "rtbfabric:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeSubnets",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeVpcs"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestedRegion": "${aws:PrincipalTag/RTBFabricRegion}"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:ListMetrics"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": "rtbfabric"
                }
```

```
            }
        }
    ]
}
```

This policy grants full RTB Fabric permissions and allows viewing of related AWS resources like network interfaces and CloudWatch metrics that RTB Fabric manages. The EC2 describe actions are scoped to regions specified in the principal's RTBFabricRegion tag for additional security.

RTB Fabric read-only permissions

This example shows a policy that allows read-only access to RTB Fabric resources and related AWS resources.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "rtbfabric:GetRequesterGateway",
                "rtbfabric:GetResponderGateway",
                "rtbfabric:ListRequesterGateways",
                "rtbfabric:ListResponderGateways",
                "rtbfabric:GetLink",
                "rtbfabric:ListLinks"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeSubnets",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeVpcs"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetMetricStatistics",
```

```
                "cloudwatch:ListMetrics"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": "rtbfabric"
                }
            }
        }
    ]
}
```

This policy grants read-only access to RTB Fabric resources and allows viewing CloudWatch metrics published by the service.

AWS managed policies for AWS RTB Fabric

AWS RTB Fabric uses AWS managed policies and service-linked roles to securely access AWS services on your behalf. AWS managed policies are standalone policies created and maintained by AWS that provide permissions for common use cases. A service-linked role is a unique type of IAM role that is linked directly to RTB Fabric and uses these managed policies to include all the permissions that the service requires to call other AWS services on your behalf.

For information about the service-linked role that RTB Fabric creates, see Using service-linked roles for RTB Fabric.

RTBFabricServiceRolePolicy

The RTBFabricServiceRolePolicy managed policy allows RTB Fabric to manage network interfaces and publish CloudWatch metrics on your behalf. This policy provides the necessary permissions for RTB Fabric to create, modify, and delete network interfaces with proper tagging controls, as well as to publish custom metrics to CloudWatch.

This policy grants the following permissions:

- *Amazon EC2 network interface management* – Allows creating network interfaces in specified subnets and security groups, with conditional permissions to create tagged network interfaces and manage network interface permissions.

- *Amazon EC2 network interface operations* – Allows deleting and detaching network interfaces that are tagged with RTBFabricManaged=true, ensuring operations are limited to RTB Fabric-managed resources.
- *Amazon EC2 tagging* – Allows creating tags on network interfaces during the CreateNetworkInterface action to properly identify RTB Fabric-managed resources.
- *Amazon EC2 describe operations* – Allows describing availability zones, network interfaces, subnets, VPCs, and security groups to gather necessary information for network interface management.
- *Amazon CloudWatch metrics* – Allows publishing custom metrics to the AWS/RTBFabric namespace for monitoring and observability purposes.

To view more details about the policy, including the latest version of the JSON policy document, see RTBFabricServiceRolePolicy in the *AWS Managed Policy Reference Guide*.

RTB Fabric updates to AWS managed policies

View details about updates to AWS managed policies for RTB Fabric since this service began tracking these changes. For automatic alerts about changes to this page, subscribe to the RSS feed on the RTB Fabric Document history page.

Change	Description	Date
RTBFabricServiceRolePolicy – Policy updated	RTB Fabric updated the CloudWatch namespace from `rtbfabric` to `AWS/RTBFabric` for publishing custom metrics.	October 16, 2025
RTBFabricServiceRolePolicy – New policy	RTB Fabric added a new managed policy that allows RTB Fabric to manage network interfaces and publish CloudWatch metrics on your behalf.	August 19, 2025

Change	Description	Date
RTB Fabric started tracking changes	RTB Fabric started tracking changes for its AWS managed policies.	March 1, 2021

Troubleshooting RTB Fabric identity and access

Use the following information to help you diagnose and fix common issues that you might encounter when working with RTB Fabric and IAM.

Topics

- I am not authorized to perform an action in RTB Fabric
- I am not authorized to perform iam:PassRole
- I want to allow people outside of my AWS account to access my RTB Fabric resources

I am not authorized to perform an action in RTB Fabric

If you receive an error that you're not authorized to perform an action, your policies must be updated to allow you to perform the action.

The following example error occurs when the `mateojackson` IAM user tries to use the console to view details about a fictional *`my-example-widget`* resource but doesn't have the fictional `rtbfabric:`*`GetWidget`* permissions.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 rtbfabric:GetWidget on resource: my-example-widget
```

In this case, the policy for the `mateojackson` user must be updated to allow access to the *`my-example-widget`* resource by using the `rtbfabric:`*`GetWidget`* action.

If you need help, contact your AWS administrator. Your administrator is the person who provided you with your sign-in credentials.

I am not authorized to perform iam:PassRole

If you receive an error that you're not authorized to perform the `iam:PassRole` action, your policies must be updated to allow you to pass a role to RTB Fabric.

Some AWS services allow you to pass an existing role to that service instead of creating a new service role or service-linked role. To do this, you must have permissions to pass the role to the service.

The following example error occurs when an IAM user named `marymajor` tries to use the console to perform an action in RTB Fabric. However, the action requires the service to have permissions that are granted by a service role. Mary does not have permissions to pass the role to the service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In this case, Mary's policies must be updated to allow her to perform the `iam:PassRole` action.

If you need help, contact your AWS administrator. Your administrator is the person who provided you with your sign-in credentials.

I want to allow people outside of my AWS account to access my RTB Fabric resources

You can create a role that users in other accounts or people outside of your organization can use to access your resources. You can specify who is trusted to assume the role. For services that support resource-based policies or access control lists (ACLs), you can use those policies to grant people access to your resources.

To learn more, consult the following:

- To learn whether RTB Fabric supports these features, see How RTB Fabric works with IAM.
- To learn how to provide access to your resources across AWS accounts that you own, see Providing access to an IAM user in another AWS account that you own in the *IAM User Guide*.
- To learn how to provide access to your resources to third-party AWS accounts, see Providing access to AWS accounts owned by third parties in the *IAM User Guide*.
- To learn how to provide access through identity federation, see Providing access to externally authenticated users (identity federation) in the *IAM User Guide*.
- To learn the difference between using roles and resource-based policies for cross-account access, see Cross account resource access in IAM in the *IAM User Guide*.

Using service-linked roles for RTB Fabric

AWS RTB Fabric uses AWS Identity and Access Management (IAM) service-linked roles. A service-linked role is a unique type of IAM role that is linked directly to RTB Fabric. Service-linked roles are predefined by RTB Fabric and include all the permissions that the service requires to call other AWS services on your behalf.

A service-linked role makes setting up RTB Fabric easier because you don't have to manually add the necessary permissions. RTB Fabric defines the permissions of its service-linked roles, and unless defined otherwise, only RTB Fabric can assume its roles. The defined permissions include the trust policy and the permissions policy, and that permissions policy cannot be attached to any other IAM entity.

You can delete a service-linked role only after first deleting their related resources. This protects your RTB Fabric resources because you can't inadvertently remove permission to access the resources.

For information about other services that support service-linked roles, see AWS services that work with IAM and look for the services that have **Yes** in the **Service-linked roles** column. Choose a **Yes** with a link to view the service-linked role documentation for that service.

Service-linked role permissions for RTB Fabric

RTB Fabric uses the service-linked role named `AWSServiceRoleForRTBFabric` – A service-linked role required for AWS RTB Fabric to access your network interface resources and deliver metrics. `AWSServiceRoleForRTBFabric` uses managed policy RTBFabricServiceRolePolicy.

The AWSServiceRoleForRTBFabric service-linked role trusts the following services to assume the role:

- `rtbfabric.amazonaws.com`

The role permissions policy allows RTB Fabric to complete the following actions on the specified resources:

- Action: `ec2:CreateNetworkInterface` on subnets and security groups
- Action: `ec2:CreateNetworkInterface` on network interfaces with the `RTBFabricManaged:true` tag

- Action: `ec2:CreateNetworkInterfacePermission` on network interfaces tagged with `RTBFabricManaged:true`
- Action: `ec2:DeleteNetworkInterface` and `ec2:DetachNetworkInterface` on network interfaces tagged with `RTBFabricManaged:true`
- Action: `ec2:CreateTags` on network interfaces during creation
- Action: `ec2:Describe*` on EC2 resources for network interface management
- Action: `cloudwatch:PutMetricData` to the `AWS/RTBFabric` namespace

The complete permissions policy for this role is available in the AWS Managed Policy Reference. For information about policy updates, see AWS managed policy updates.

You must configure permissions to allow your users, groups, or roles to create, edit, or delete a service-linked role. For more information, see Service-linked role permissions in the *IAM User Guide*.

Creating a service-linked role for RTB Fabric

You don't need to manually create a service-linked role. When you create an RTB application for the first time (CreateRequesterRtbApp or CreateResponderRtbApp) in the AWS Management Console, the AWS CLI, or the AWS API, RTB Fabric creates the service-linked role for you.

If you delete this service-linked role, and then need to create it again, you can use the same process to recreate the role in your account. When you create an RTB application for the first time (CreateRequesterRtbApp or CreateResponderRtbApp), RTB Fabric creates the service-linked role for you again.

You can also use the IAM console to create a service-linked role with the **RTB Fabric** use case. In the AWS CLI or the AWS API, create a service-linked role with the `rtbfabric.amazonaws.com` service name. For more information, see Creating a service-linked role in the *IAM User Guide*. If you delete this service-linked role, you can use this same process to create the role again.

Editing a service-linked role for RTB Fabric

RTB Fabric does not allow you to edit the AWSServiceRoleForRTBFabric service-linked role. After you create a service-linked role, you cannot change the name of the role because various entities might reference the role. However, you can edit the description of the role using IAM. For more information, see Editing a service-linked role in the *IAM User Guide*.

Deleting a service-linked role for RTB Fabric

If you no longer need to use a feature or service that requires a service-linked role, we recommend that you delete that role. That way you don't have an unused entity that is not actively monitored or maintained. However, you must clean up the resources for your service-linked role before you can manually delete it.

> **Note**
>
> If the RTB Fabric service is using the role when you try to delete the resources, then the deletion might fail. If that happens, wait for a few minutes and try the operation again.

To delete RTB Fabric resources used by the AWSServiceRoleForRTBFabric

1. Delete all RTB applications in your account. You must delete both requester and responder gateways before you can delete the service-linked role. You can use either the console or the CLI:

   a. **Console method:** For instructions on deleting requester gateways, see Deleting a requester gateway. For instructions on deleting responder gateways, see Deleting a responder gateway.

   b. **CLI method:** Use the `DeleteRequesterGateway` or `DeleteResponderGateway` API to delete RTB gateways. Replace the example gateway ID with your gateway ID:

   ```
   aws rtbfabric delete-requester-gateway --gateway-id rtb-gw-abc123xyz789
   ```

   The response returns a status of `DELETING`:

   ```
   {
   "gatewayId": "rtb-gw-abc123xyz789",
   "status": "DELETING"
   }
   ```

   Repeat this command for each RTB application in your account.

2. After deleting all RTB applications, wait for RTB Fabric to automatically clean up the network interfaces tagged with `RTBFabricManaged:true`. This process can take up to 20 minutes.

3. Verify that no RTB applications or RTB Fabric-managed network interfaces remain in your account:

   a. Open the RTB Fabric console and verify that no RTB applications are listed.
   b. Open the Network Interfaces page of the Amazon EC2 console.
   c. In the search box, enter `tag:RTBFabricManaged:true` to filter for RTB Fabric-managed network interfaces.
   d. Verify that no network interfaces appear in the results.

> **Note**
>
> RTBFabric only deletes the network interface if no other RTBFabric configuration is using that network interface. If you have multiple RTBFabric configurations using the same subnet and security group combination, the network interface will remain until all configurations are removed.

> **Note**
>
> RTBFabric relies on the service-linked role permissions to delete network interfaces. Do not delete the AWSServiceRoleForRTBFabric role before RTBFabric completes the network interface cleanup, or the cleanup may fail.

To manually delete the service-linked role using IAM

Use the IAM console, the AWS CLI, or the AWS API to delete the AWSServiceRoleForRTBFabric service-linked role. For more information, see Deleting a service-linked role in the *IAM User Guide*.

Supported Regions for RTB Fabric service-linked roles

RTB Fabric supports using service-linked roles in all of the Regions where the service is available. For more information, see AWS Regions and endpoints.

Data protection in AWS RTB Fabric

The AWS shared responsibility model applies to data protection in AWS RTB Fabric. As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the

AWS Cloud. You are responsible for maintaining control over your content that is hosted on this infrastructure. You are also responsible for the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see the Data Privacy FAQ. For information about data protection in Europe, see the AWS Shared Responsibility Model and GDPR blog post on the *AWS Security Blog*.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual users with AWS IAM Identity Center or AWS Identity and Access Management (IAM). That way, each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We require TLS 1.2 and recommend TLS 1.3.
- Set up API and user activity logging with AWS CloudTrail. For information about using CloudTrail trails to capture AWS activities, see Working with CloudTrail trails in the *AWS CloudTrail User Guide*.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing sensitive data that is stored in Amazon S3.
- If you require FIPS 140-3 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see Federal Information Processing Standard (FIPS) 140-3.

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form text fields such as a **Name** field. This includes when you work with RTB Fabric or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into tags or free-form text fields used for names may be used for billing or diagnostic logs. If you provide a URL to an external server, we strongly recommend that you do not include credentials information in the URL to validate your request to that server.

Data encryption

RTB Fabric supports encrypting data in transit using TLS 1.2 or higher when you configure TLS for RTB communications between applications. All API calls to RTB Fabric are encrypted using HTTPS endpoints.

RTB Fabric supports HTTP/1.1 with gzip compression for all RTB traffic to optimize bandwidth usage and improve performance. Compression is automatically applied when supported by both endpoints, reducing the size of bid requests and responses during transmission.

Encryption in transit

When you configure TLS encryption, RTB traffic between requester and responder applications is encrypted using TLS 1.2+ protocols. This ensures that bid requests, responses, and other RTB communications are protected during transmission across networks when TLS is enabled.

Encryption at rest

RTB Fabric encrypts configuration data and application metadata at rest using AWS managed encryption keys. This includes RTB application configurations, link settings, and operational logs. AWS does not access or read your operational logs, maintaining the confidentiality of your RTB data and business operations.

Incident response for AWS RTB Fabric

TBD

Compliance validation for AWS RTB Fabric

RTB Fabric inherits compliance certifications from the AWS platform and follows AWS security best practices. The service is built on AWS infrastructure that has been designed and verified in accordance with industry standards and best practices.

For information about whether a specific compliance program is in scope for AWS services, see AWS services in scope by compliance program. For general information, see AWS compliance programs.

You can download third-party audit reports using AWS Artifact. For more information, see Downloading reports in AWS Artifact.

Your compliance responsibility when using RTB Fabric is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. AWS provides resources to help with compliance including security and compliance quick start guides and whitepapers.

Monitoring RTB Fabric

Monitoring is an important part of maintaining the reliability, availability, and performance of RTB Fabric and your other AWS solutions. AWS provides the following monitoring tools to watch RTB Fabric, report when something is wrong, and take automatic actions when appropriate:

- *Amazon CloudWatch* monitors your AWS resources and the applications you run on AWS in real time. You can collect and track metrics, create customized dashboards, and set alarms that notify you or take actions when a specified metric reaches a threshold that you specify. For example, you can have CloudWatch track CPU usage or other metrics of your Amazon EC2 instances and automatically launch new instances when needed. For more information, see the Amazon CloudWatch User Guide.
- *Amazon CloudWatch Logs* enables you to monitor, store, and access your log files from Amazon EC2 instances, CloudTrail, and other sources. CloudWatch Logs can monitor information in the log files and notify you when certain thresholds are met. You can also archive your log data in highly durable storage. For more information, see the Amazon CloudWatch Logs User Guide.
- *AWS CloudTrail* captures API calls and related events made by or on behalf of your AWS account and delivers the log files to an Amazon S3 bucket that you specify. You can identify which users and accounts called AWS, the source IP address from which the calls were made, and when the calls occurred. For more information, see the AWS CloudTrail User Guide.

Amazon EventBridge is a serverless event bus service that makes it easy to connect your applications with data from a variety of sources. EventBridge delivers a stream of real-time data from your own applications, Software-as-a-Service (SaaS) applications, and AWS services and routes that data to targets such as Lambda. This enables you to monitor events that happen in services, and build event-driven architectures. For more information, see the Amazon EventBridge User Guide.

Monitoring RTB Fabric with Amazon CloudWatch

You can monitor RTB Fabric using CloudWatch, which collects raw data and processes it into readable, near real-time metrics. These statistics are kept for 15 months, so that you can access historical information and gain a better perspective on how your web application or service is performing. You can also set alarms that watch for certain thresholds, and send notifications or take actions when those thresholds are met. For more information, see the Amazon CloudWatch User Guide.

> **Note**
>
> RTB Fabric CloudWatch metrics are available in the following regions only: US East (N. Virginia), US West (Oregon), Europe (Ireland), Europe (Frankfurt), Asia Pacific (Singapore), and Asia Pacific (Tokyo).

For RTB Fabric, you can monitor request volume, latency, HTTP status codes, and infrastructure metrics to track the performance and health of your RTB gateways and links.

The RTB Fabric service reports metrics in the `AWS/RTBFabric` namespace.

Configuring RTB Fabric logs with Amazon CloudWatch Logs

You can configure RTB Fabric to send application logs to Amazon CloudWatch Logs using log delivery. Logging is not enabled by default and requires setup.

Topics

Setting up log delivery for RTB Fabric

To enable logging for RTB Fabric, you need to create a log delivery source, destination, and delivery configuration. Only links can be registered as log sources, and only APPLICATION_LOGS log type is supported.

To set up log delivery for RTB Fabric

1. Create a log delivery source for your link. The resource ARN must specify a link within a gateway:

```
aws logs put-delivery-source \
  --name rtbfabric-delivery \
  --resource-arn arn:aws:rtbfabric:us-east-1:545746263314:gateway/rtb-gw-
m8x4n2p9q7r5s1t6u3v8w0y2z/link/link-a9b7c5d3e1f4g8h2i6j0k4l7m \
  --log-type APPLICATION_LOGS
```

2. Create a log delivery destination (such as an Amazon S3 bucket or CloudWatch log group).
3. Create the delivery configuration to connect the source and destination.

For detailed information about log delivery setup, see Configure standard logging in the *Amazon CloudFront Developer Guide* for a similar implementation pattern.

Required permissions for log delivery

To set up log delivery for RTB Fabric, you need the following IAM permissions:

```
{
    "Sid": "AllowLogDeliveryCreation",
    "Effect": "Allow",
    "Action": [
        "logs:PutDeliverySource",
        "logs:PutDeliveryDestination",
        "logs:CreateDelivery"
    ],
    "Resource": "*"
}
```

Additionally, you need service-level permissions for the specific link resource:

```
{
    "Sid": "ServiceLevelAccessForLogDelivery",
    "Effect": "Allow",
    "Action": [
        "rtbfabric:AllowVendedLogDeliveryForResource"
    ],
    "Resource": "arn:aws:rtbfabric:us-east-1:545746263314:gateway/rtb-gw-
m8x4n2p9q7r5s1t6u3v8w0y2z/link/link-a9b7c5d3e1f4g8h2i6j0k4l7m"
}
```

You can harden the resource permissions by specifying exact ARNs instead of using wildcards, and add additional actions like delete operations as needed.

Configuring log sampling rates

You can configure log sampling rates when creating or accepting links to control the volume of logs generated. This helps manage costs and focus on the most relevant log data.

Example of setting log sampling rates when accepting a link:

```
aws rtbfabric accept-link \
    --link-id link-brhta7fllwkwlb3l7gbpofkn \
    --gateway-id rtb-gw-d43re9jdjmkw2r08e6psphe37 \
    --log-settings '{
        "applicationLogs": {
            "sampling": {
                "errorLog": 100.0,
                "filterLog": 100.0
            }
        }
    }'
```

The sampling rates are specified as percentages (0.0 to 100.0) where:

- `errorLog` – Percentage of error logs to capture
- `filterLog` – Percentage of filter logs to capture

You can also configure sampling rates when creating links using the `CreateLink` operation with similar log-settings parameters.

RTB Fabric metrics

RTB Fabric publishes the following metrics to CloudWatch.

Metric	Description
`total-request-count`	The total number of requests received by the service. Valid Dimensions: Link Valid Statistics: Sum Units: Count
`success-request-count`	The number of successful requests processed by the service. Valid Dimensions: Link Valid Statistics: Sum

Metric	Description
	Units: Count
`failure-request-count`	The number of failed requests. Valid Dimensions: Link Valid Statistics: Sum Units: Count
`request-status-count`	The number of requests broken down by HTTP status codes. Valid Dimensions: HttpStatusCode, Link Valid Statistics: Sum Units: Count
`forwarding-latency`	The time taken to forward requests. Valid Dimensions: Link, Statistic (P90, P95, P99) Valid Statistics: Average, Maximum, Minimum Units: Milliseconds
`total-latency`	The end-to-end request processing time. Valid Dimensions: Link, Statistic (P90, P95, P99) Valid Statistics: Average, Maximum, Minimum Units: Milliseconds
`target-ip-count`	The number of target IP addresses. Valid Statistics: Sum, Average Units: Count

RTB Fabric dimensions

The following dimensions are supported for RTB Fabric metrics.

Dimension	Description
`Link`	The unique identifier for the link between RTB gateways.
`HttpStatusCode`	The HTTP status code returned by the service (for example, 200, 404, 500). Not available for all metrics.
`Statistic`	The statistical measure for latency metrics (P90, P95, P99). Not available for all metrics.

Creating CloudWatch alarms for RTB Fabric

You can create CloudWatch alarms to monitor RTB Fabric metrics and automatically notify you when metric values cross specified thresholds. This helps you proactively respond to issues with your RTB gateways and links.

Common alarms you might want to create include:

- *High failure rate* – Monitor the `failure-request-count` metric to detect when error rates exceed acceptable thresholds. Calculate success rate using (total - failure) / total.
- *High latency* – Monitor the `total-latency` or `forwarding-latency` metrics to detect performance degradation. Subtract the two to see broker processing time.
- *Low request volume* – Monitor the `total-request-count` metric to detect unexpected drops in traffic.
- *HTTP error rates* – Monitor the `request-status-count` metric filtered by HTTP status codes (4xx, 5xx) to detect client or server errors.

For information about creating CloudWatch alarms, see Creating Amazon CloudWatch alarms in the *Amazon CloudWatch User Guide*.

Creating CloudWatch dashboards for RTB Fabric

You can create CloudWatch dashboards to visualize RTB Fabric metrics and monitor the health and performance of your RTB gateways and links in real time.

Consider creating dashboard widgets for:

- *Request volume trends* – Display `total-request-count`, `success-request-count`, and `failure-request-count` metrics over time.
- *Latency performance* – Show `total-latency` and `forwarding-latency` metrics with P90, P95, and P99 statistics.
- *Error rate monitoring* – Track `request-status-count` metrics broken down by HTTP status codes.
- *Infrastructure health* – Monitor `target-ip-count` to track the availability of target endpoints.

For information about creating CloudWatch dashboards, see Creating a CloudWatch dashboard in the *Amazon CloudWatch User Guide*.

Logging AWS RTB Fabric API calls using AWS CloudTrail

AWS RTB Fabric is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service. CloudTrail captures all API calls for RTB Fabric as events. The calls captured include calls from the RTB Fabric console and code calls to the RTB Fabric API operations. Using the information collected by CloudTrail, you can determine the request that was made to RTB Fabric, the IP address from which the request was made, when it was made, and additional details.

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root user or user credentials.
- Whether the request was made on behalf of an IAM Identity Center user.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

CloudTrail is active in your AWS account when you create the account and you automatically have access to the CloudTrail **Event history**. The CloudTrail **Event history** provides a viewable, searchable, downloadable, and immutable record of the past 90 days of recorded management events in an AWS Region. For more information, see Working with CloudTrail Event history in the *AWS CloudTrail User Guide*. There are no CloudTrail charges for viewing the **Event history**.

For an ongoing record of events in your AWS account past 90 days, create a trail or a CloudTrail Lake event data store.

CloudTrail trails

A *trail* enables CloudTrail to deliver log files to an Amazon S3 bucket. All trails created using the AWS Management Console are multi-Region. You can create a single-Region or a multi-Region trail by using the AWS CLI. Creating a multi-Region trail is recommended because you capture activity in all AWS Regions in your account. If you create a single-Region trail, you can view only the events logged in the trail's AWS Region. For more information about trails, see Creating a trail for your AWS account and Creating a trail for an organization in the *AWS CloudTrail User Guide*.

You can deliver one copy of your ongoing management events to your Amazon S3 bucket at no charge from CloudTrail by creating a trail, however, there are Amazon S3 storage charges. For more information about CloudTrail pricing, see AWS CloudTrail Pricing. For information about Amazon S3 pricing, see Amazon S3 Pricing.

CloudTrail Lake event data stores

CloudTrail Lake lets you run SQL-based queries on your events. CloudTrail Lake converts existing events in row-based JSON format to Apache ORC format. ORC is a columnar storage format that is optimized for fast retrieval of data. Events are aggregated into *event data stores*, which are immutable collections of events based on criteria that you select by applying advanced event selectors. The selectors that you apply to an event data store control which events persist and are available for you to query. For more information about CloudTrail Lake, see Working with AWS CloudTrail Lake in the *AWS CloudTrail User Guide*.

CloudTrail Lake event data stores and queries incur costs. When you create an event data store, you choose the pricing option you want to use for the event data store. The pricing option determines the cost for ingesting and storing events, and the default and maximum retention period for the event data store. For more information about CloudTrail pricing, see AWS CloudTrail Pricing.

RTB Fabric management events in CloudTrail

Management events provide information about management operations that are performed on resources in your AWS account. These are also known as control plane operations. By default, CloudTrail logs management events.

AWS RTB Fabric logs all RTB Fabric control plane operations as management events. For a list of the AWS RTB Fabric control plane operations that RTB Fabric logs to CloudTrail, see the AWS RTB Fabric API Reference.

RTB Fabric event examples

An event represents a single request from any source and includes information about the requested API operation, the date and time of the operation, request parameters, and so on. CloudTrail log files aren't an ordered stack trace of the public API calls, so events don't appear in any specific order.

The following example shows a CloudTrail event that demonstrates the `AcceptLink` operation.


```
{
      "eventVersion": "1.09",
      "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AAAABACDEFFGHIJ3KLM5N:IntegrationTest",
        "arn": "arn:aws:sts::123456789012:assumed-role/TestRole/IntegrationTest",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
          "sessionIssuer": {
            "type": "Role",
            "principalId": "AAAABACDEFFGHIJ3KLM5N",
            "arn": "arn:aws:iam::123456789012:role/TestRole",
            "accountId": "123456789012",
            "userName": "TestRole"
          },
          "attributes": {
            "creationDate": "2025-10-01T22:16:35Z",
            "mfaAuthenticated": "false"
          }
        }
      },
      "eventTime": "2025-10-01T22:17:29Z",
```

```
      "eventSource": "rtbfabric.amazonaws.com",
      "eventName": "AcceptLink",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "10.0.0.1",
      "userAgent": "aws-sdk-java/2.34.3 md/io#sync md/http#Apache ua/2.1 api/
Some#2.34.x os/Linux#5.10.242-219.961.amzn2int.x86_64 lang/java#17.0.16 md/OpenJDK_64-
Bit_Server_VM#17.0.16+8-LTS md/vendor#Amazon.com_Inc. md/en_US md/kotlin/2.0.21-
release-482 cfg/auth-source#stsrole m/D,N",
      "requestParameters": {
        "rtbGatewayId": "rtb-gw-responder456",
        "linkId": "link-12345678",
        "attributes": {
          "customerProvidedId": "accepted-link-123"
        },
        "logSettings": {
          "applicationLogs": {
            "sampling": {
              "errorLog": 0,
              "filterLog": 0
            }
          }
        }
      },
      "responseElements": {
        "rtbGatewayId": "rtb-gw-responder456",
        "peerRtbGatewayId": "rtb-gw-requester123",
        "linkId": "link-12345678",
        "createdTimestamp": 1695734400,
        "attributes": {
          "customerProvidedId": "accepted-link-123"
        },
        "state": "ACTIVATING",
        "updatedTimestamp": 1695734500,
        "direction": "INBOUND"
      },
      "requestID": "ba5b8aa9-30a5-4a65-88eb-8e8c9d644d48",
      "eventID": "200c17c9-40c9-4f2c-b24f-864c3a4db0b9",
      "readOnly": false,
      "eventType": "AwsApiCall",
      "managementEvent": true,
      "recipientAccountId": "123456789012",
      "eventCategory": "Management"
}
```

For information about CloudTrail record contents, see CloudTrail record contents in the *AWS CloudTrail User Guide*.

Quotas for AWS RTB Fabric

Your AWS account has default quotas, formerly referred to as limits, for each AWS service. Unless otherwise noted, each quota is Region-specific. You can request increases for RTB Fabric quotas.

To view the quotas for RTB Fabric, open the Service Quotas console. In the navigation pane, choose **AWS services** and select **RTB Fabric**.

To request a quota increase, see Requesting a Quota Increase in the *Service Quotas User Guide*. If the quota is not yet available in Service Quotas, use the limit increase form.

Your AWS account has the following quotas related to RTB Fabric.

Resource quotas

The following quotas apply to RTB Fabric resources in your account.

Resource	Default quota	Adjustable	Description
Number of gateways	2	Yes	The maximum number of requester and responder gateways combined that you can create in your account.
Links per gateway	2	Yes	The maximum number of links that can be created from a single gateway.
Modules per flow	2	Yes	The maximum number of modules that can be configure d in a single flow.
Subnets per gateway	1	Yes	The maximum number of subnets

Resource	Default quota	Adjustable	Description
			that can be configure d per gateway.
External outbound links supported	No	Yes	Indicates whether gateways can be configured with outbound external links. Supported values: 0 or 1.
External inbound links supported	No	Yes	Indicates whether gateways can be configured with inbound external links. Supported values: 0 or 1.

Throughput quotas

The following quotas apply to RTB Fabric throughput and performance.

Resource	Default quota	Adjustable	Description
Transactions per second (TPS) per link	1,000	Yes	The maximum number of transacti ons per second that can be processed through a single link.
Transactions per second (TPS) per inbound or outbound external link	1,000	Yes	The maximum number of transacti ons per second that can be processed through an inbound

Resource	Default quota	Adjustable	Description
			or outbound external link.
HTTP request timeout	1 second	Yes	The maximum time RTB Fabric waits for a response from RTB applications before timing out the request. Contact your AWS account manager to request increases.

API request quotas

The following quotas apply to RTB Fabric API requests. These quotas are not adjustable.

Resource	Default quota	Description
API requests per second per account per Region	10	The maximum number of API requests per second for control plane operations in each supported Region in your account.

Quota increase considerations

When requesting quota increases for RTB Fabric, consider the following:

- **Account verification** – Quota increases may require account verification to ensure legitimate use for real-time bidding applications.
- **Regional capacity** – Quota increases are subject to available capacity in the requested Region.
- **Performance impact** – Higher quotas may affect latency and performance characteristics. Test thoroughly after quota increases.

- **Cost implications** – Higher quotas may result in increased costs. Review pricing before requesting increases.

For strategic partnerships or enterprise-level requirements, contact your AWS account team for assistance with quota planning and optimization.

AWS RTB Fabric Glossary

Application

A customer-built software system that connects to RTB Fabric gateways to send or receive real-time bidding requests. Applications are external to RTB Fabric and can be requester applications that send bid requests or responder applications that receive and process bid requests. Applications connect to RTB Fabric through gateways, which handle the routing and processing of requests between applications.

Availability Zone (AZ)

Distinct locations within an AWS Region that are engineered to be isolated from failures in other AZs.

Customer

User of the service. Includes Demand-Side Platforms (DSPs) and Supply-Side Platforms (SSPs).

Demand-Side Platform (DSP)

An adtech system that allows advertisers to buy ad inventory from multiple ad exchanges through one interface. Also known as a buyer. The service helps DSPs to participate effectively in real-time bidding auctions with a high-performance, low-latency infrastructure.

Flow

Part of a link configuration that defines how RTB requests are processed. Created with the `UpdateLinkModuleFlow`. Flows can include RTB modules configured for specific behaviors such as additional filtering and enrichment actions. Default flows are created with links, but you can update a flow with additional modules.

Gateway

RTB Fabric infrastructure components that serve as connection points for customer applications. *Requester gateways* receive requests from requester applications and forward them through links

to responder gateways. *Responder gateways* receive requests from requester gateways and forward them to responder applications, then return responses through the same pathway.

Link

The core component of RTB Fabric that establishes secure, bidirectional communication channels between gateways. Links provide authenticated communication, traffic routing and load balancing, performance monitoring, and security controls. Links can include processing logic through modules.

Requester

A customer application that sends bid requests to RTB Fabric through a requester gateway.

Resources

Gateways, links, flows, and RTB modules that comprise the RTB Fabric infrastructure. Customer applications are external to RTB Fabric and connect to these infrastructure resources.

Responder

A customer application that receives bid requests from RTB Fabric through a responder gateway and returns bid responses.

RTB module

A component that performs specific operations on RTB requests, such as filtering or enrichment. RTB modules can be configured by Supply-Side Platforms (SSPs) or Demand-Side Platforms (DSPs) in their flows, although they are not required to configure a flow. RTB Fabric provides built-in modules (QPS rate limiter, OpenRTB filter, error masker) available at no additional charge. Modules are configured using the `UpdateLinkModuleFlow` API operation.

Supply-Side Platform (SSP)

An adtech system that helps publishers manage and sell their ad inventory through programmatic auctions to Demand-Side Platforms (DSPs) and advertising agencies. Also known as a seller. The service helps SSPs with very low latency requirements process high volumes of real-time bid requests and responses.

Virtual private cloud (VPC)

A logically isolated network in the AWS Cloud. This virtual network resembles a traditional network that you'd operate in your own data center, with the benefits of using the scalable infrastructure of AWS. For more information, see Amazon Virtual Private Cloud (Amazon VPC).

Document history for the AWS RTB Fabric User Guide

The following table describes the documentation releases for RTB Fabric.

Change	Description	Date
RTB Fabric updates to AWS managed policies - Policy updated	RTB Fabric updated the `RTBFabricServiceRolePolicy` managed policy to change the CloudWatch namespace from `rtbfabric` to `AWS/RTBFabric` for publishing custom metrics. For information, see RTB Fabric updates to AWS managed policies.	October 22, 2025
RTB Fabric updates to AWS managed policies - New policy	RTB Fabric has released a new managed policy `RTBFabricServiceRolePolicy` that allows RTB Fabric to manage network interfaces and publish CloudWatch metrics on your behalf. For information, see RTB Fabric updates to AWS managed policies.	October 22, 2025
Initial release	Initial release of the RTB Fabric User Guide	October 22, 2025