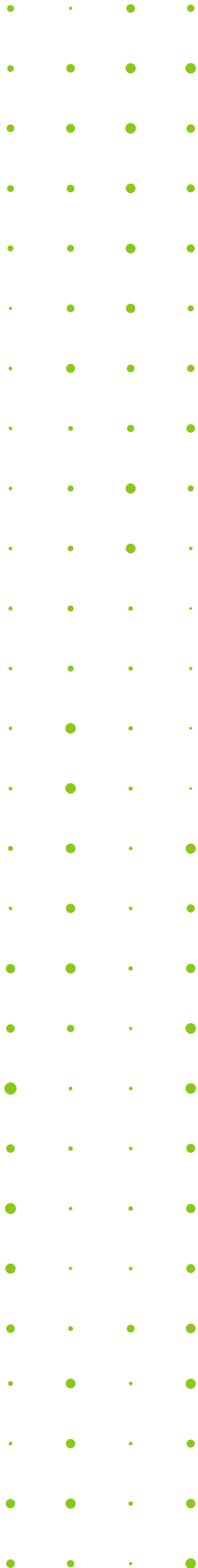


COHESITY

Cohesity AWS SaaS Connector Deployment Guide

February 14, 2024



© 2024 Cohesity, Inc. All rights reserved.

Cohesity, the Cohesity logo, SnapTree, SpanFS, DataPlatform, DataProtect, Helios, the Helios logo, DataGovern, SiteContinuity, DataHawk, and other Cohesity marks are trademarks or registered trademarks of Cohesity, Inc. in the US and/or internationally. Other company and product names may be trademarks of the respective companies with which they are associated. This material (a) is intended to provide you information about Cohesity and our business and products; (b) was believed to be true and accurate at the time it was written, but is subject to change without notice; and (c) is provided on an "AS IS" basis. Cohesity disclaims all express or implied conditions, representations, warranties of any kind.

No part of this documentation or any related software may be reproduced, stored, transmitted, or otherwise distributed in any form or by any means (electronic or otherwise) for any purpose other than the purchaser's personal use without the prior written consent of Cohesity, Inc. You may not use, modify, perform or display this documentation or any related software for any purpose except as expressly set forth in a separate written agreement executed by Cohesity, Inc., and any other use (including without limitation for the reverse engineering of such software or creating compatible software or derivative works) is prohibited, except to the extent such restrictions are prohibited by applicable law.

Published on February 14, 2024

Contents

- AWS SaaS Connector Deployment 4
 - Deployment Options 4
 - Register AWS Source and Deploy AWS SaaS Connector 35

AWS SaaS Connector Deployment

Before you [deploy your AWS SaaS Connectors](#), review the deployment options and detailed, step-by-step procedures below.

- [Deployment Options](#) 4
- [Register AWS Source and Deploy AWS SaaS Connector](#)35

Deployment Options

There are two primary deployment options for the AWS SaaS Connector:

- [VPC with Public Subnet](#)
- [VPC with Private Subnet and NAT Gateway](#)

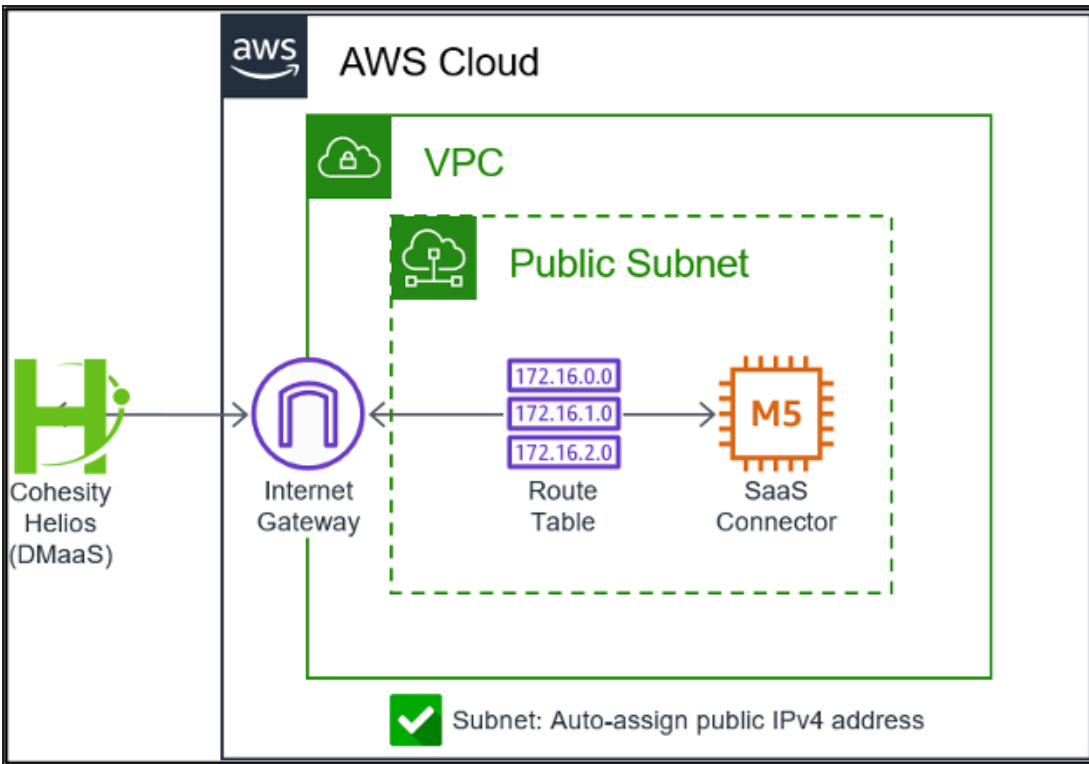
Regardless of which deployment option is used, once you have prepared the AWS configuration the steps to deploy the AWS SaaS Connector are the same. Complete the instructions for one of the deployment options.

Next > Once you have completed the AWS configuration, you are ready to [register an AWS source](#) and then [deploy the AWS SaaS Connector](#).

VPC with Public Subnet

This section provides the prerequisites and the steps to be followed for deploying the AWS SaaS connector using VPC with public subnet.

The image below shows an overview of the AWS configuration.



Prerequisites

- A VPC is deployed with a subnet.
- Configure the security group and network ACL allow traffic to traverse the network over the required ports. For a list of required ports, see [SaaS Connection Requirements](#).

Process Overview

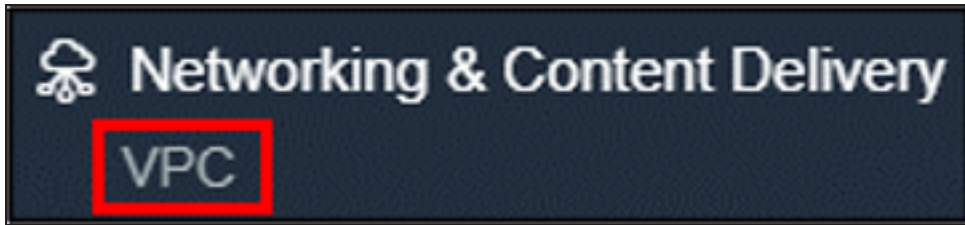
- Deployment of an [internet gateway \(IGW\)](#), which requires the provisioning of an elastic IP address (public IP address).
- Updates to the [route table](#) for the public subnet.
- Enabling [auto-assign public IPv4 address](#) for the public subnet.

Deploy Internet Gateway

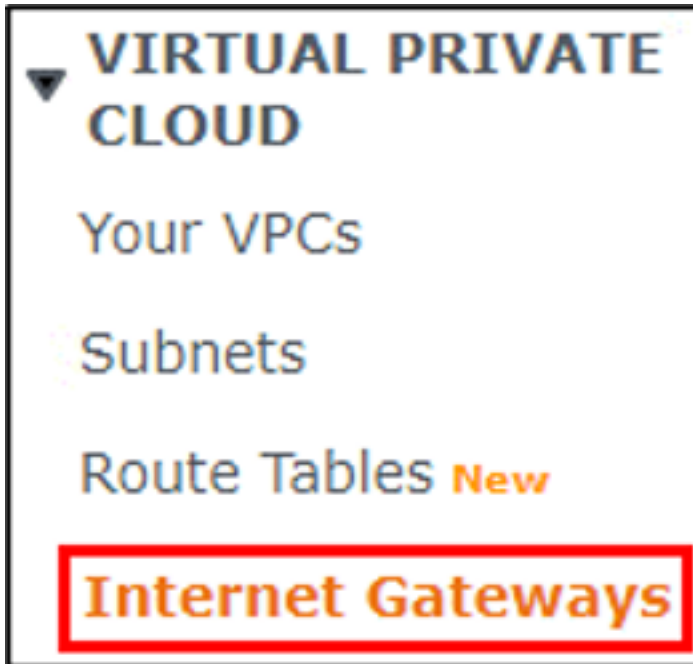
This section provides instructions to deploy an Internet gateway.

To deploy an Internet gateway:

1. Go to the **VPC** service in the AWS console.



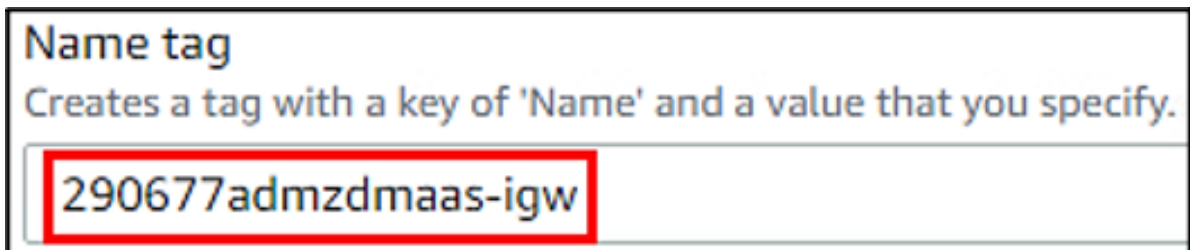
2. Under the **VIRTUAL PRIVATE CLOUD** menu, click **Internet Gateways**.



3. Click the **Create internet gateway** button.




4. Enter a **Name tag** that will allow you to easily identify this resource.



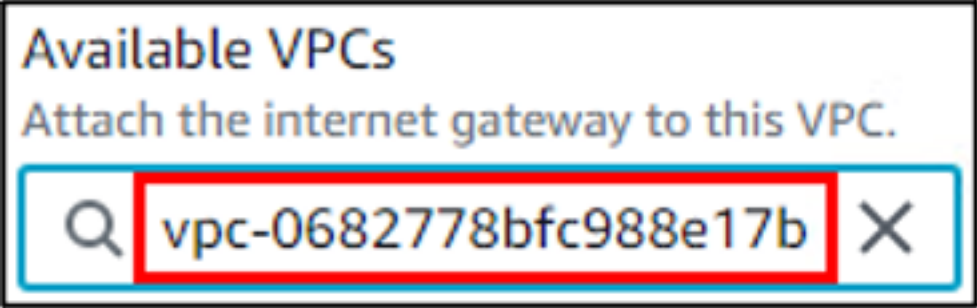
5. Click the **Create internet gateway** button.

An orange rectangular button with a black border and rounded corners, containing the text "Create internet gateway" in white, bold, sans-serif font.


6. Click the **Attach to a VPC** button that appears at the top of the console.

A white rectangular button with a green border and rounded corners, containing the text "Attach to a VPC" in blue, bold, sans-serif font.

7. Select the appropriate VPC from the list of **Available VPCs**.

A screenshot of the AWS console showing the "Available VPCs" section. The text "Attach the internet gateway to this VPC." is visible. A search bar contains the VPC ID "vpc-0682778bfc988e17b" which is highlighted with a red box. The search bar also includes a magnifying glass icon on the left and an 'X' icon on the right.

8. Click the **Attach internet gateway** button.

An orange rectangular button with a black border and rounded corners, containing the text "Attach internet gateway" in white, bold, sans-serif font.

The Internet gateway is now attached to your VPC.

Next > You're ready to [update the route table](#).

Update Route Table

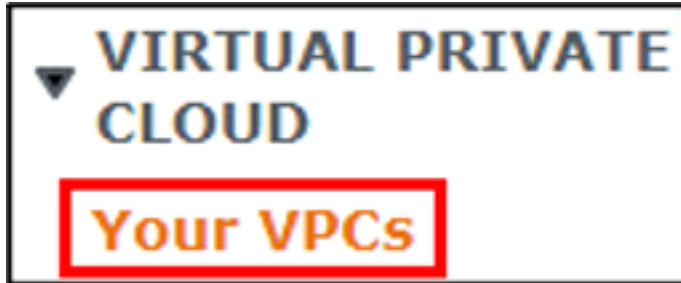
You need to update the routing table for the public subnet to route traffic to the Internet gateway. For this document, we will route all traffic that is not intended for the VPCs assigned CIDR block to the Internet. This may vary for customers that are using VPC peering, AWS Direct Connect, Transit Gateways, and more that may need to include additional routes in the route table.

To update the route table:

1. If you are not already there, go to the **VPC** service in the AWS console.



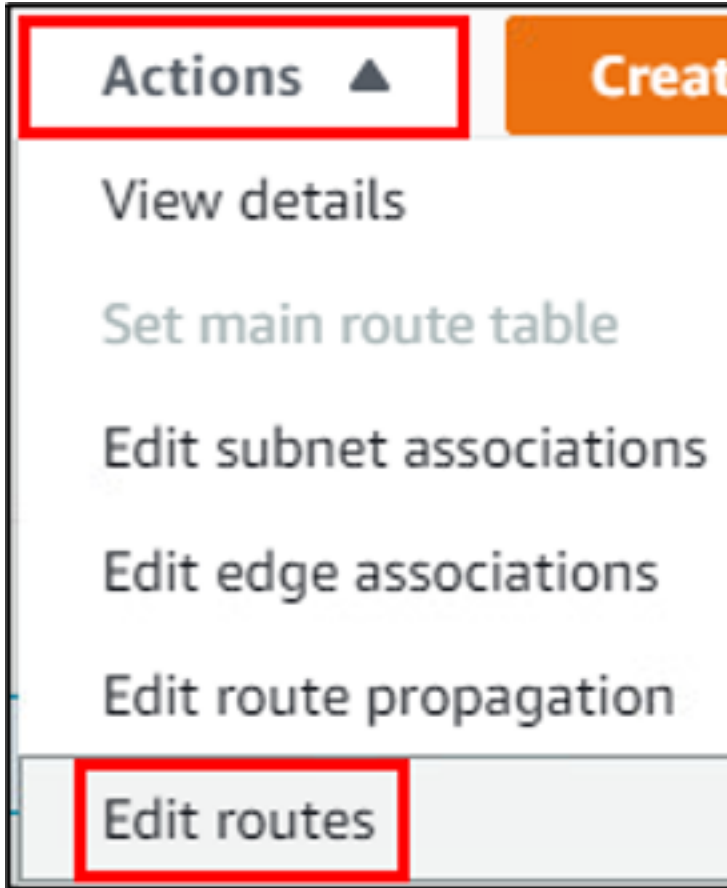
2. Under the **VIRTUAL PRIVATE CLOUD** menu, click **Your VPCs**.



3. Click the **Main route table** link for the appropriate VPC. This will take you directly to the route table for the VPC and select it (checkbox is selected by default for the route table).

Name	VPC ID	Main route table
290677admzdmaas	vpc-0682778bfc988e17b	rtb-02aa5b64d3888fe84

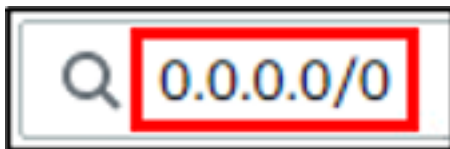
4. With the route table selected, click the **Actions** menu, then click **Edit routes**.



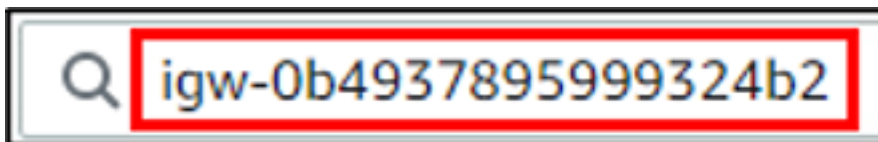
- 5. Click the **Add route** button.



- 6. For the **Destination** column, click the field and select **0.0.0.0/0** from the list.



- 7. Click the **Target** field and select **Internet Gateway** from the list, then select the Internet gateway that was created earlier in this process.



- 8. Click the **Save changes** button.



The route table for the VPC is now updated. Traffic that is not local (not within the VPC CIDR block) will now be routed to the Internet gateway.

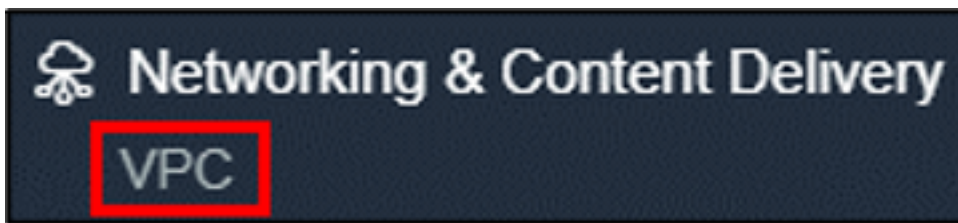
Next > You're ready to [enable auto-assign IPv4 public IP addresses](#).

Enable Auto-Assign IPv4 Public IP Address

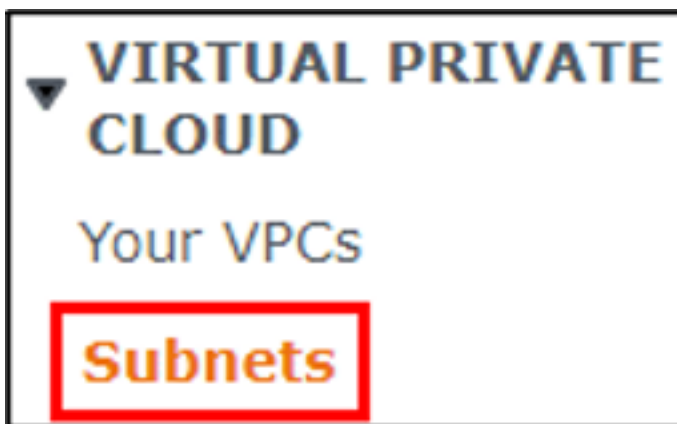
When the AWS SaaS Connector is deployed from Helios it will need a public IP address (elastic IP address). This is not assigned by the deployment code. It is necessary to enable the automatic assignment of an IPv4 public IP address by default for the VPC subnet. This is only necessary for the SaaS Connector deployment period. Once the SaaS Connector is deployed this option can be disabled. If additional SaaS Connectors are deployed in the future, or if auto-scaling is enabled, this option must be re-enabled.

To enable auto-assign IPv4 public IP addresses:

1. If you are not already there, go to the **VPC** service in the AWS console.



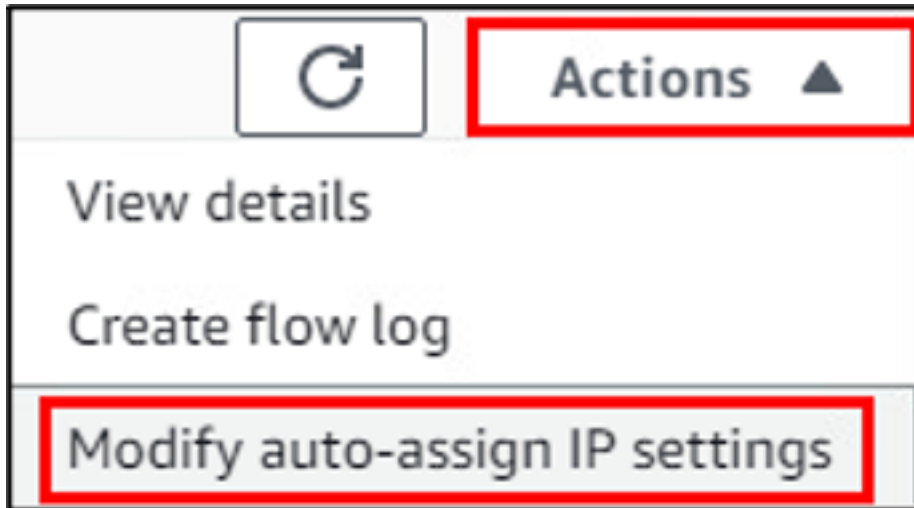
2. Under the **VIRTUAL PRIVATE CLOUD** menu, click **Subnets**.



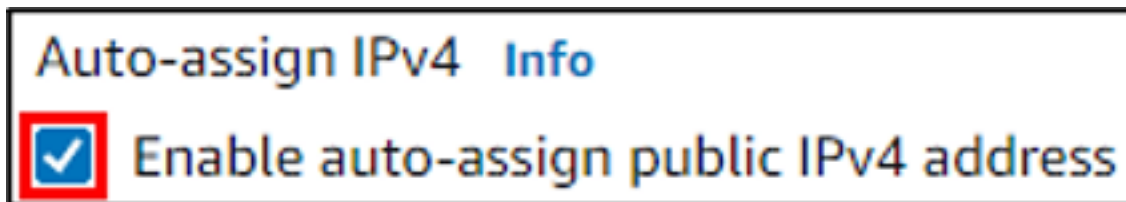
3. Click the checkbox to select the subnet for the appropriate VPC.

<input checked="" type="checkbox"/>	Name	Subnet ID	VPC
<input checked="" type="checkbox"/>	290677admzdmaas	subnet-0b160805d65b02ba2	vpc-0682778bfc988e17b 290677admzdmaas

4. With the subnet selected, click the **Actions** menu, then click **Modify auto-assign IP settings**.



5. Click the checkbox to select **Enable auto-assign public IPv4 address**.



6. Click the **Save** button.



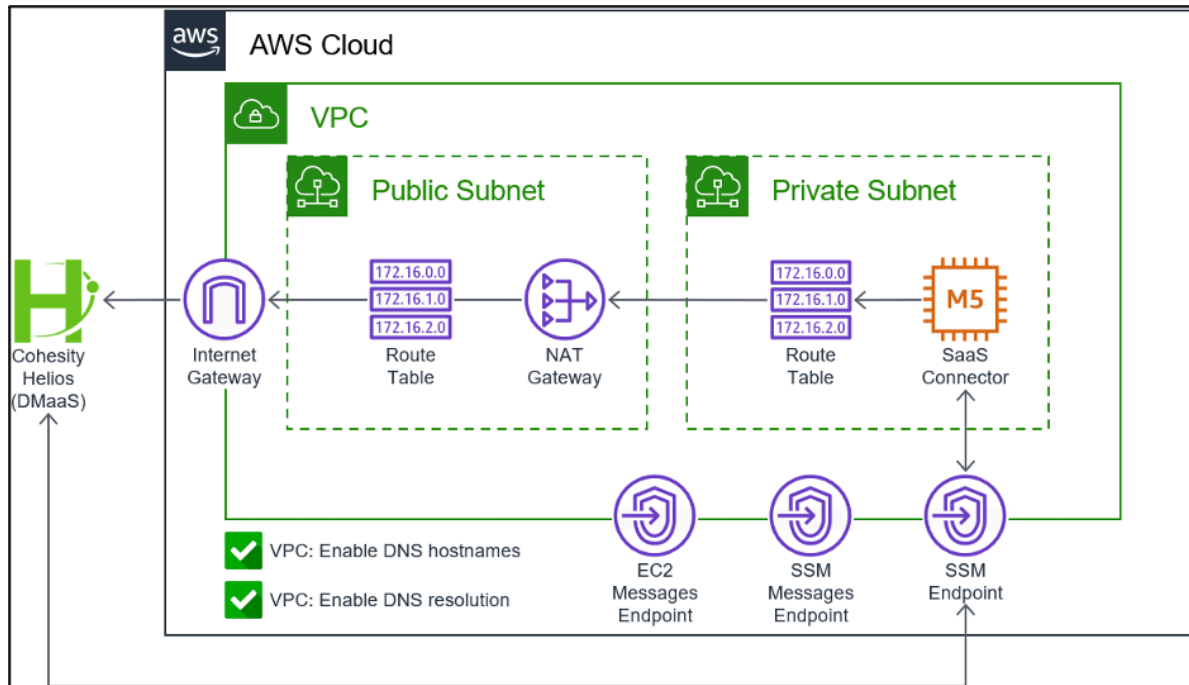
The AWS configuration is now ready for the AWS SaaS Connector deployment.

Next > You're now ready to [register an AWS source](#) and [deploy the AWS SaaS Connectors](#).

VPC with Private Subnet and NAT Gateway

This section provides the prerequisites and the steps to be followed for deploying the AWS SaaS connector using VPC with private subnet.

The image below shows an overview of the AWS configuration



Prerequisites

- A VPC is deployed with two subnets. One subnet for public and another for private.
- Configure the security group and network ACL allow traffic to traverse the network over the required ports. For a list of required ports, see [SaaS Connection Requirements](#).

Process Overview

- Deployment of an [internet gateway \(IGW\)](#), which requires the provisioning of an elastic IP address (public IP address).
- Allocating an [elastic \(public\) IP address](#) for the NAT gateway (NGW).
- Deployment of a [NAT gateway](#).
- Update the default route table to be the [public route table](#) and create a [private route table](#).
- Enabling [DNS resolution and hostnames](#) for the VPC.
- Deployment of endpoints for [SSM](#), [SSM messages](#), and [EC2 messages](#).

Subnet Configuration

If you have an existing VPC with a single subnet it is possible (likely) you have a subnet configuration with a /24 subnet that consumes all IP addresses within the VPCs assigned CIDR block. If this is the case, you will need to consider either deploying a new VPC with two subnets or delete the existing subnet and create two new subnets that split the subnet IP address assignment.

For example, a VPC has an assigned CIDR block of 172.31.41.0/24. If you want to split the available IP addresses evenly between the two subnets you can assign the public subnet the 172.31.41.0/25 range and the private subnet the 172.31.41.128/25 range.

Name	Subnet ID	VPC	IPv4 CIDR
290841admzdmaas-private	subnet-053393e7aa5128505	vpc-0f015f35434287f9d 290841admzdmaas	172.31.41.128/25
290841admzdmaas-public	subnet-08011a0796fa0dbbd	vpc-0f015f35434287f9d 290841admzdmaas	172.31.41.0/25

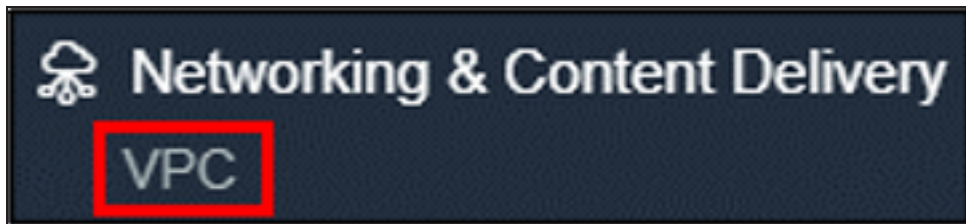
Next > You're ready to [deploy an Internet Gateway](#).

Deploy Internet Gateway

This section provides instructions to deploy an Internet gateway.

To deploy an Internet gateway:

1. Go to the **VPC** service in the AWS console.



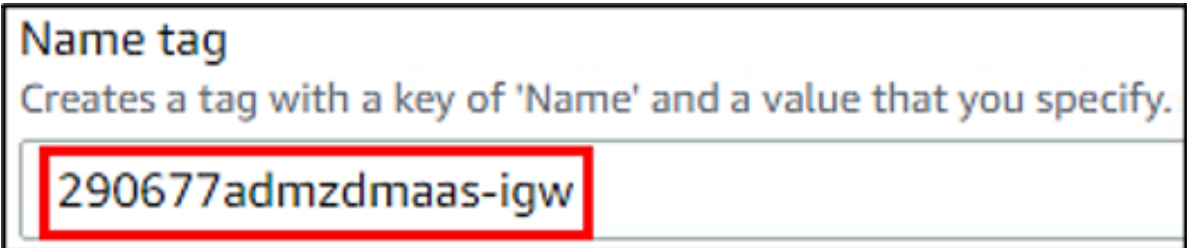
2. Under the **VIRTUAL PRIVATE CLOUD** menu, click **Internet Gateways**.



3. Click the **Create internet gateway** button.



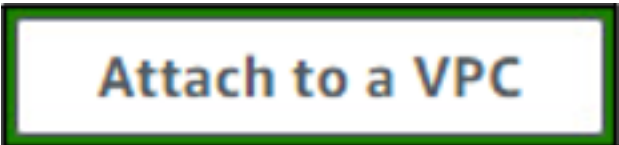
4. Enter a **Name tag** that will allow you to easily identify this resource.



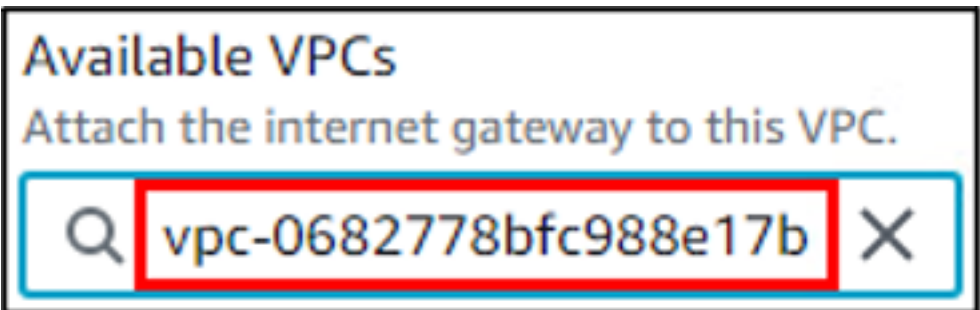
5. Click the **Create internet gateway** button.



6. Click the **Attach to a VPC** button that appears at the top of the console.



7. Select the appropriate VPC from the list of **Available VPCs**.



8. Click the **Attach internet gateway** button.



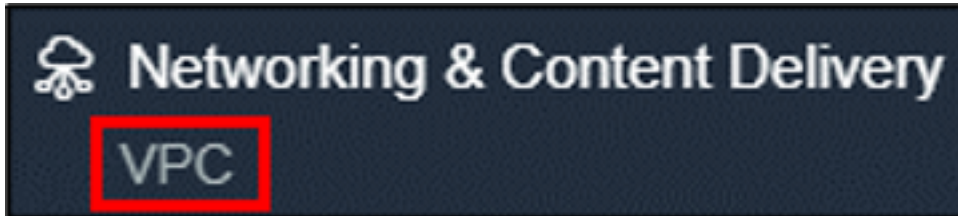
The Internet gateway is now attached to your VPC. The Internet gateway will allow traffic in the public subnet to reach the Internet, which includes the NAT gateway that will be deployed.

Next > You're ready to [allocate elastic IP address](#).

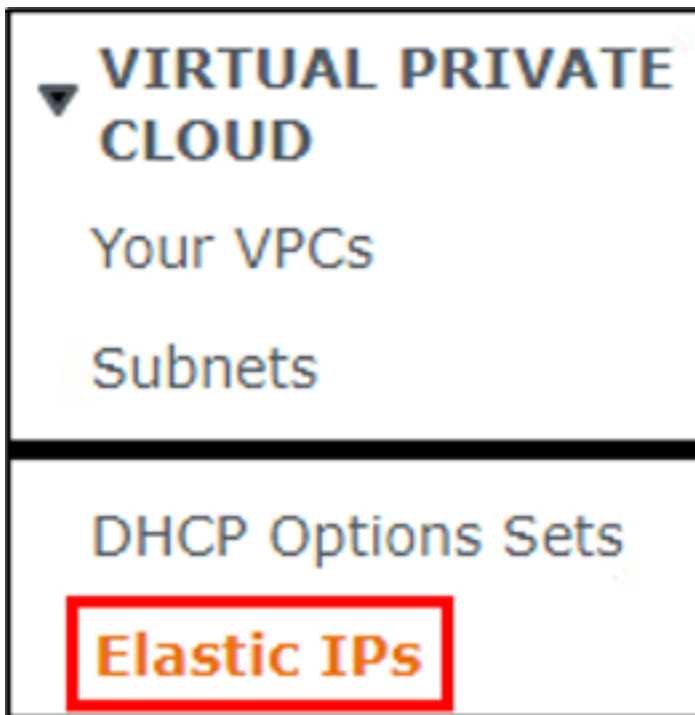
Allocate Elastic IP Address

Before a NAT Gateway can be deployed, we must first provision a public IP address:

1. If you are not already there, go to the **VPC** service in the AWS console.



2. Under the **VIRTUAL PRIVATE CLOUD** menu, click **Elastic IPs**.



3. Click the **Allocate Elastic IP** address button.



4. While not required, it is recommended that you add a name tag for the resource. For the **Key** field, enter **Name**. For the **Value** field, enter an appropriate name for the elastic IP address.

Key	Value - optional
<input type="text" value="Name"/>	<input type="text" value="290841admzdmaas-ngw-ip"/>

5. Click the **Allocate** button.



The elastic IP address is now allocated and ready to be used for the NAT gateway.

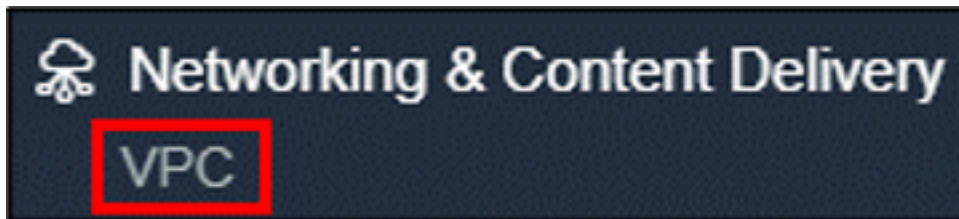
Next > You're ready to [deploy NAT Gateway](#).

Deploy NAT Gateway

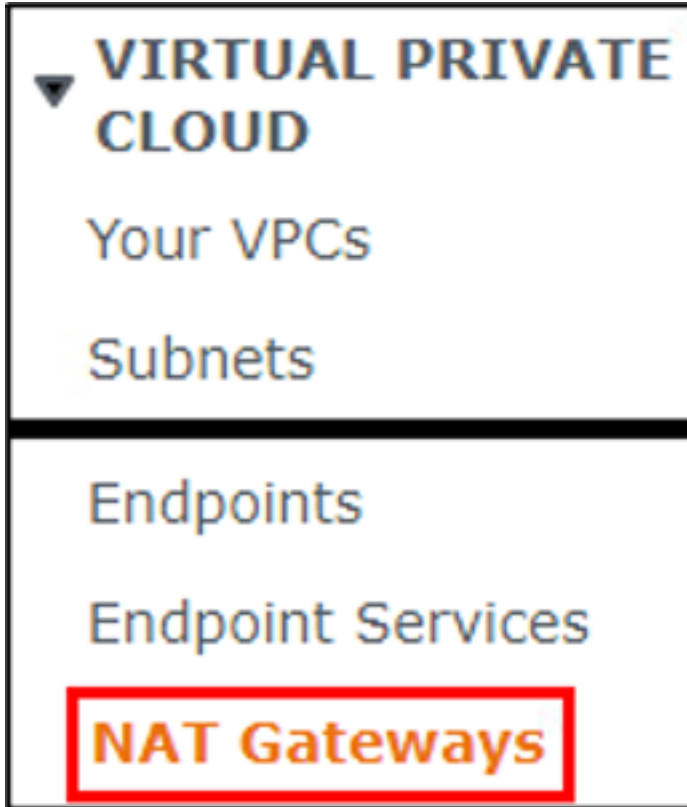
This topic provides instructions to deploy a NAT Gateway.

To deploy a NAT gateway:

1. If you are not already there, go to the **VPC** service in the AWS console.



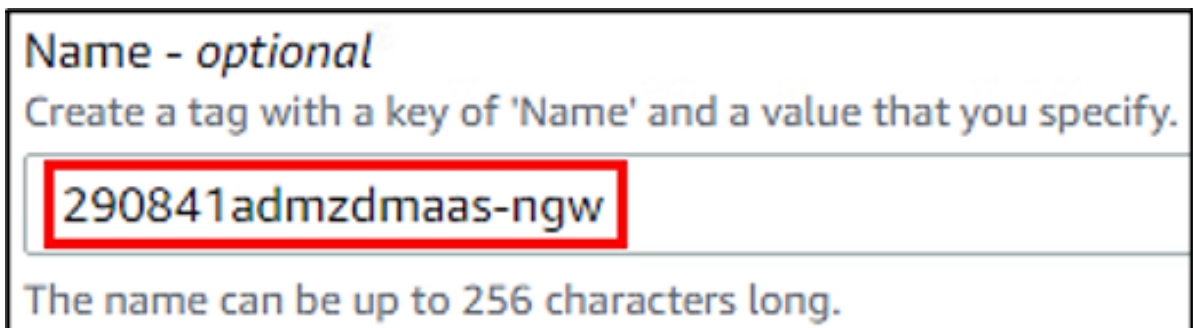
2. Under the **VIRTUAL PRIVATE CLOUD** menu, click **NAT Gateways**.



- 3. Click the **Create NAT gateway** button.



While not required, it is recommended that you add a name tag for the resource. Enter an appropriate value in the **Name** field.



- 4. For the **Subnet** field, select the public subnet for the VPC.

Subnet
Select a subnet in which to create the NAT gateway.

subnet-08011a0796fa0dbbd (290841admzdmaas-public)

- Keep the default selection for **Connectivity type**.

Connectivity type
Select a connectivity type for the NAT gateway.

Public

Private

- For the **Elastic IP allocation ID** field, select the elastic IP that was allocated earlier in this process.

Elastic IP allocation ID [Info](#)
Assign an Elastic IP address to the NAT gateway.

eipalloc-01ebc98e22a6c6561 (290841admzdmaas-ngw-ip)

- Click the **Create NAT gateway** button.

Create NAT gateway

The NAT gateway is now deployed in the public subnet and ready to use. It is critical that the NAT gateway is deployed in the public subnet as the routing table for the public subnet will be updated to route traffic from the NAT gateway to the Internet gateway. The private subnet route table will be updated to route traffic to the NAT gateway.

Next > You're ready to [update the public route table](#).

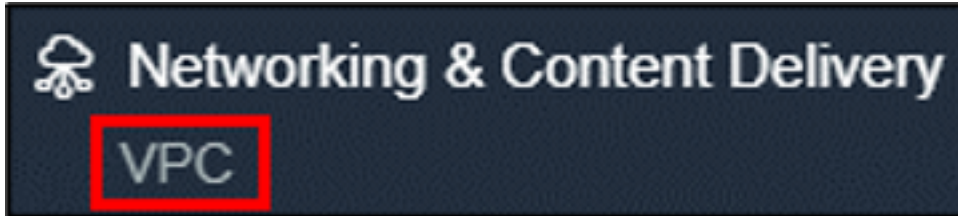
Update Public Route Table

We'll now update the routing table for the public subnet to route traffic to the Internet gateway. Since the NAT gateway is in the public subnet, this means that traffic directed to the NAT Gateway will be able to access the internet through the Internet gateway.

We'll use the default route table by updating the name of the route table and explicitly associating the route table with the public subnet.

To update the public route table:

1. If you are not already there, go to the **VPC** service in the AWS console.



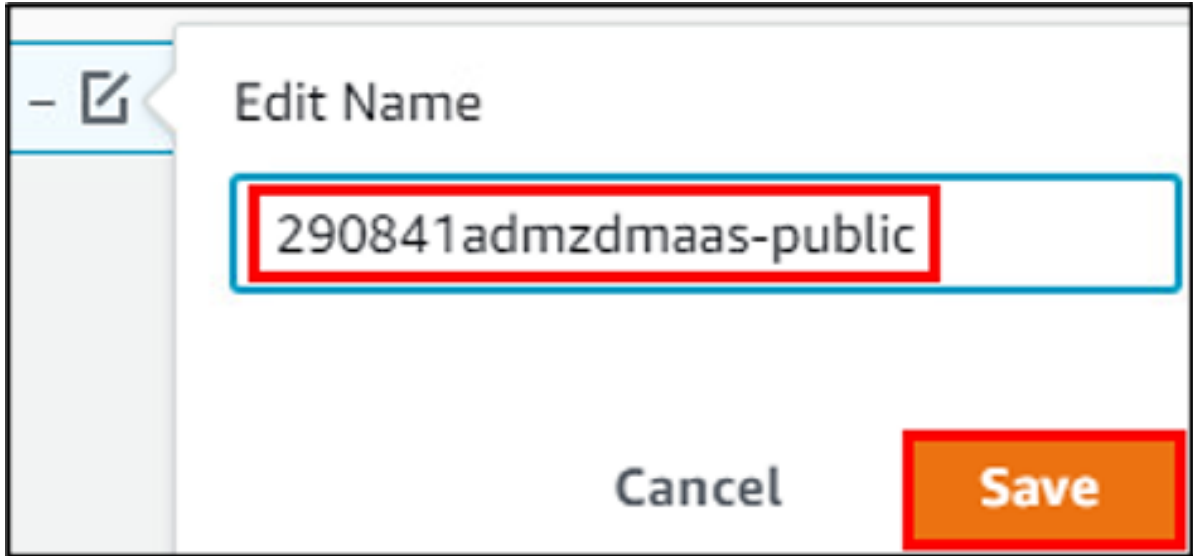
2. Under the **VIRTUAL PRIVATE CLOUD** menu, click **Your VPCs**.



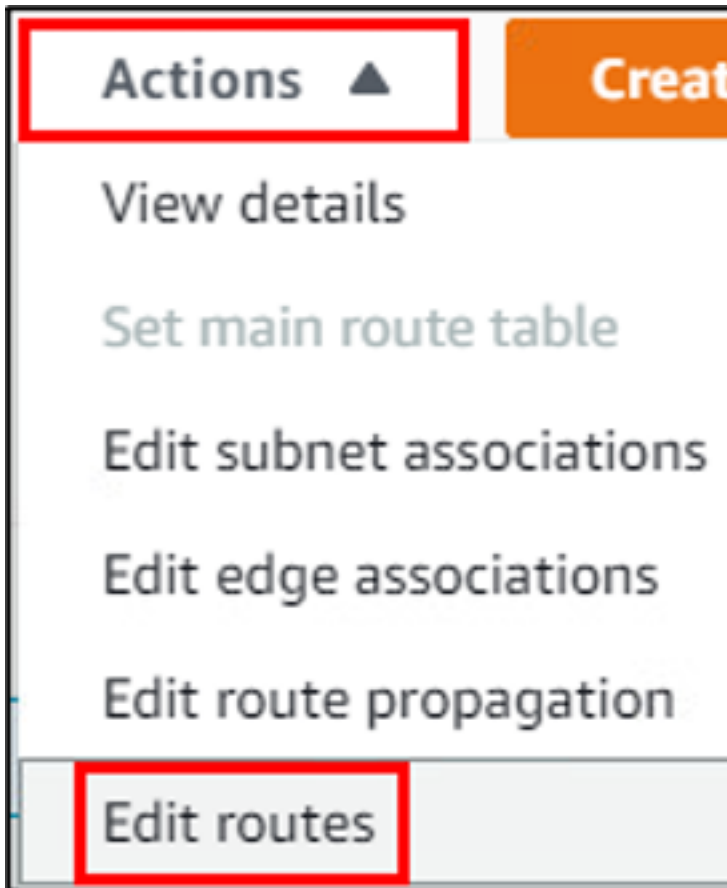
3. Click the **Main route table** link for the appropriate VPC. This will take you directly to the route table for the VPC and select it (checkbox is selected by default for the route table).

Name	VPC ID	Main route table
290677admzdmaas	vpc-0682778bfc988e17b	rtb-02aa5b64d3888fe84

4. Click the **Name** field. Enter a name that identifies the subnet as the public subnet, then click the **Save** button.



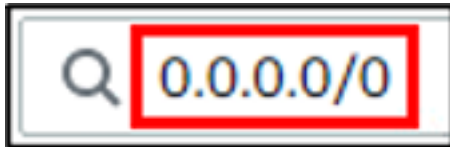
- 5. With the route table selected, click the **Actions** menu, then click **Edit routes**.



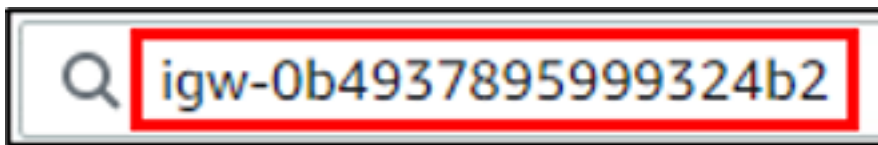
- 6. Click the **Add route** button.



- 7. For the **Destination** column, click the field and select **0.0.0.0/0** from the list.



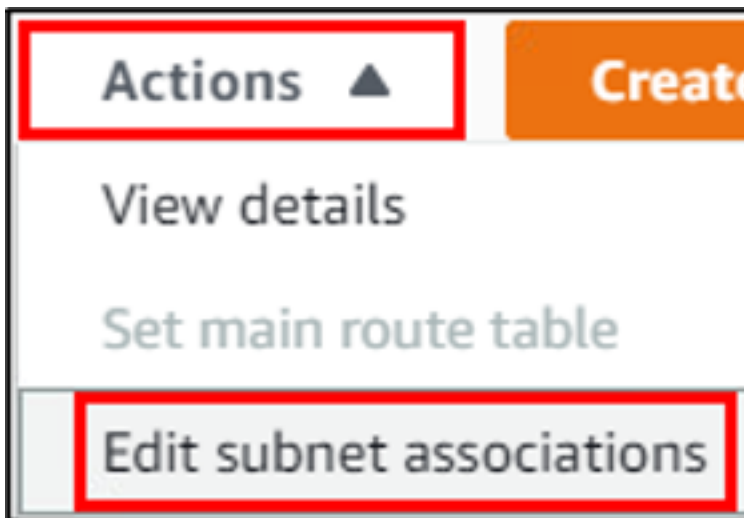
- 8. Click the **Target** field and select **Internet Gateway** from the list, then select the Internet gateway that was created earlier in this process.



- 9. Click the **Save changes** button.



- 10. With the route table selected, click the **Actions** menu, then click **Edit subnet associations**



- 11. Click the checkbox to select the public subnet for the VPC.

<input type="checkbox"/>	Name	Subnet ID	IPv4 CIDR
<input type="checkbox"/>	290841admzdmaas-private	subnet-053393e7aa5128505	172.31.41.128/25
<input checked="" type="checkbox"/>	290841admzdmaas-public	subnet-08011a0796fa0dbbd	172.31.41.0/25

12. Click the **Save associations** button.



The public route table is now updated, and the route table is explicitly associated to the public subnet. Traffic that is not local to the public subnet will now be routed to the Internet gateway.

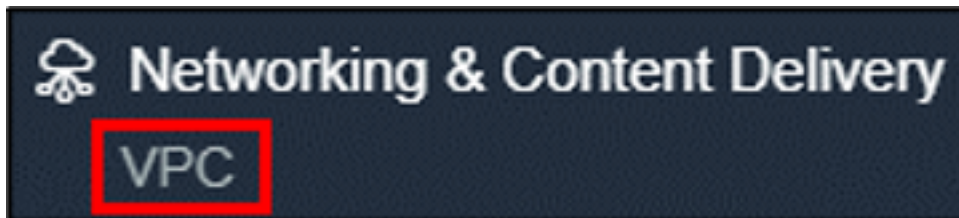
Next > You're ready to create a [private route table](#).

Create Private Route Table

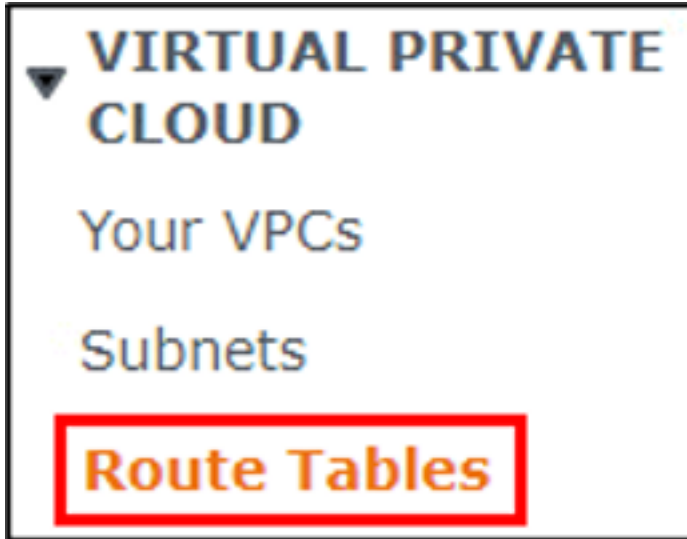
We'll now create the private route table and explicitly associate it to the private subnet.

To create the private route table:

1. If you are not already there, go to the **VPC** service in the AWS console.



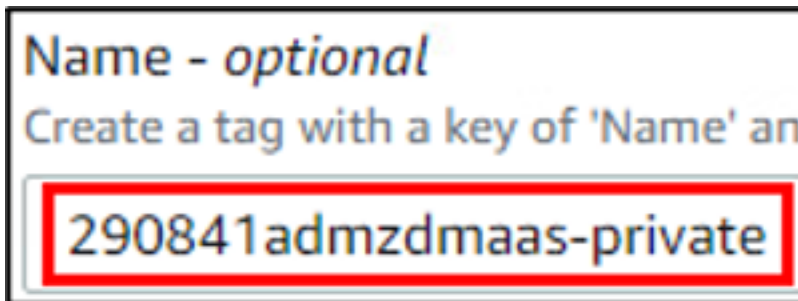
2. Under the **VIRTUAL PRIVATE CLOUD** menu, click **Route Tables**.



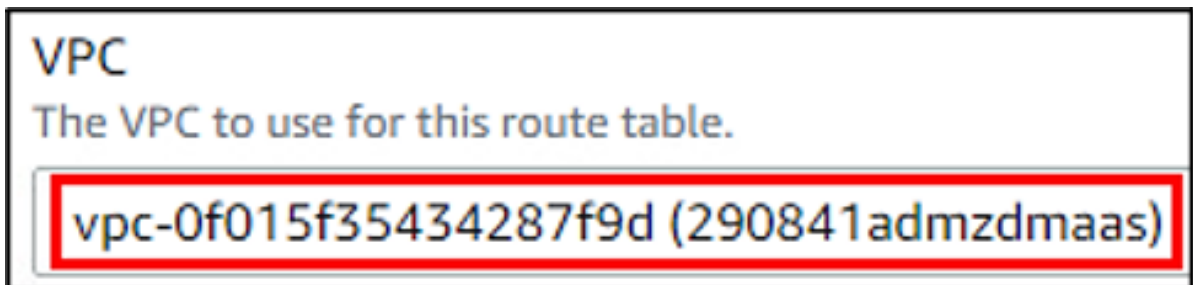
- 3. Click the **Create route table** button.



- 4. For the **Name** field, enter a name that identifies the subnet as the private subnet.



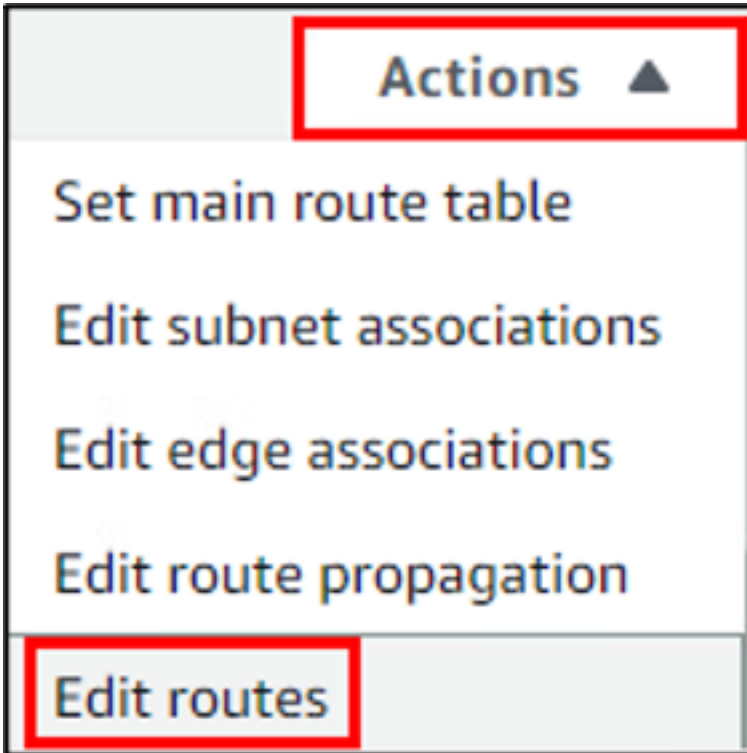
- 5. For the **VPC** field, select the appropriate VPC entry from the list.



- 6. Click the **Create route table** button.



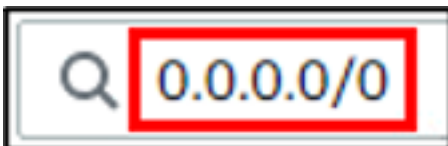
- 7. Click the **Actions** menu, then click **Edit routes**.



- 8. Click the **Add route** button.



- 9. For the **Destination** column, click the field and select **0.0.0.0/0** from the list.



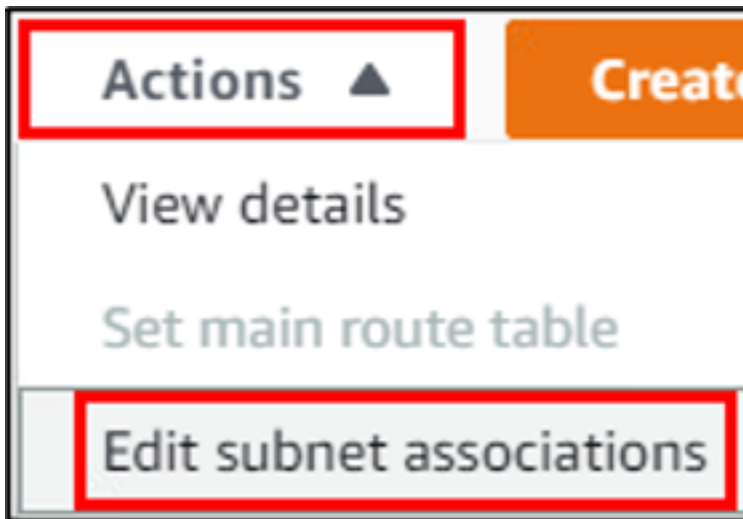
- 10. Click the **Target** field and select **NAT Gateway** from the list, then select the NAT gateway that was created earlier in this process.



- 11. Click the **Save changes** button.



- 12. Click the **Actions** menu then click **Edit subnet associations**.



- 13. Click the checkbox to select the private subnet for the VPC.

<input type="checkbox"/>	Name	Subnet ID	IPv4 CIDR
<input checked="" type="checkbox"/>	290841admzdmaas-private	subnet-053393e7aa5128505	172.31.41.128/25
<input type="checkbox"/>	290841admzdmaas-public	subnet-08011a0796fa0dbbd	172.31.41.0/25

- 14. Click the **Save associations** button.



The private route table has now been created and explicitly associated to the private subnet. Traffic that is not local to the private subnet will now be routed to the NAT gateway.

Next > You're ready to [enable DNS resolution and hostnames](#).

Enable DNS Resolution and Hostnames

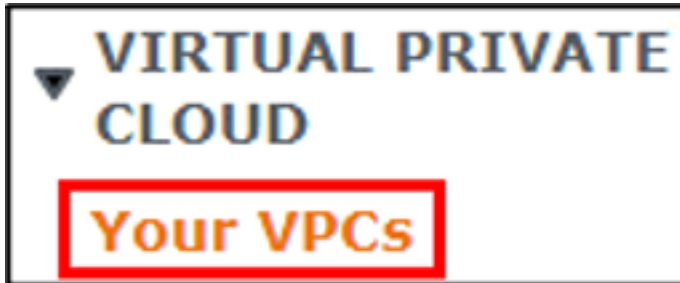
To support the deployment of endpoints for the VPC, DNS resolution and DNS hostnames must be enabled for the VPC.

To enable DNS:

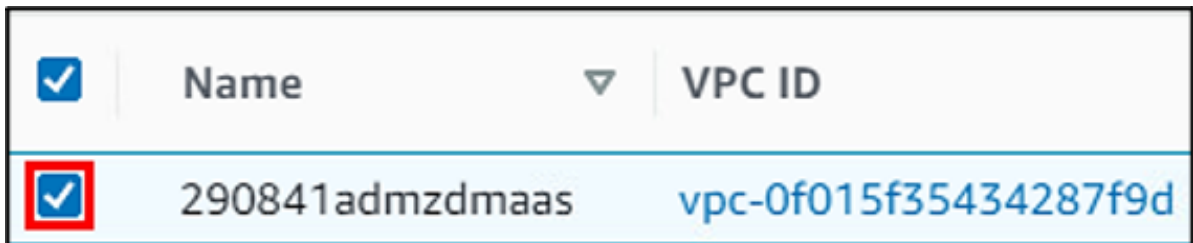
1. If you are not already there, go to the **VPC** service in the AWS console.



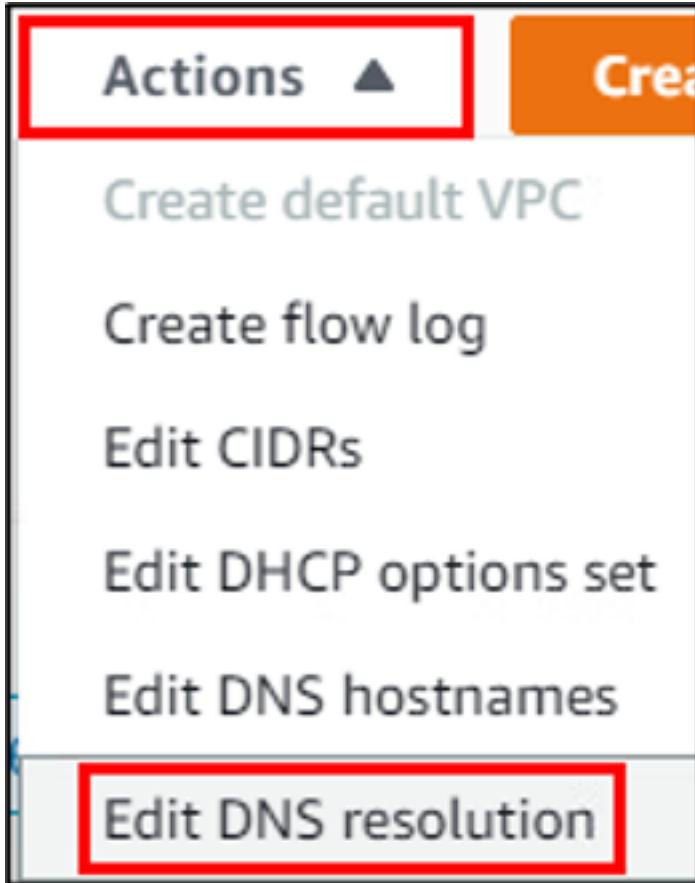
2. Under the **VIRTUAL PRIVATE CLOUD** menu, click **Your VPCs**.



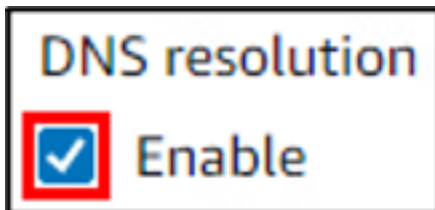
3. Click the checkbox for the appropriate VPC to select it.



4. Click the **Actions** menu, then click **Edit DNS resolution**.



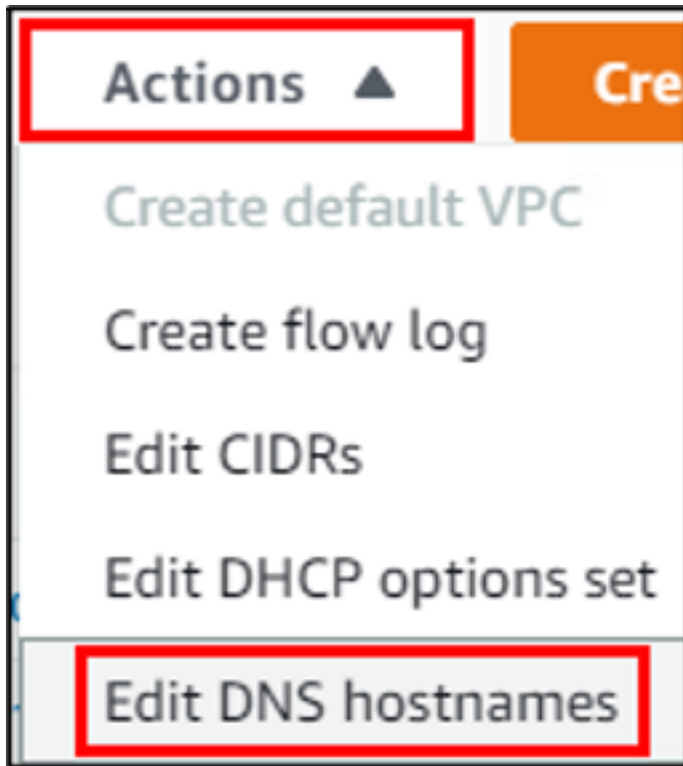
- 5. Click the **Enable** checkbox to select it for the **DNS resolution** field.



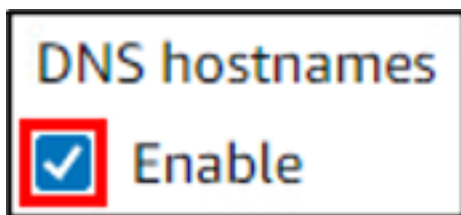
- 6. Click the **Save changes** button.



- 7. With the VPC selected, click the **Actions** menu, then click **Edit DNS hostnames**.



- 8. Click the **Enable** checkbox to select it for the **DNS hostnames** field.



- 9. Click the **Save changes** button.



DNS resolution and hostnames is now enabled for the VPC.

Next > You're ready to [deploy SSM endpoints](#).

Deploy SSM Endpoint

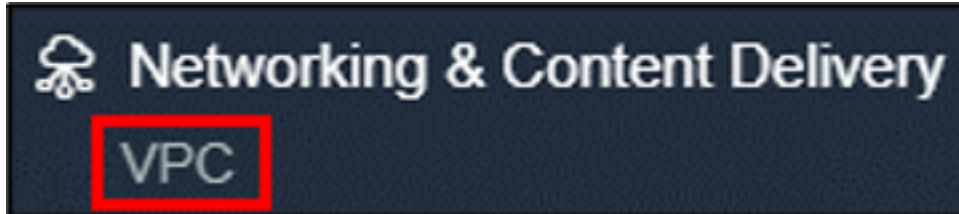
SSM is used during the AWS SaaS Connector deployment so connect to the EC2 instance and complete the VM configuration and claim process.

For information about deploying gateway endpoints for Amazon S3, see [Gateway endpoints for Amazon S3](#).

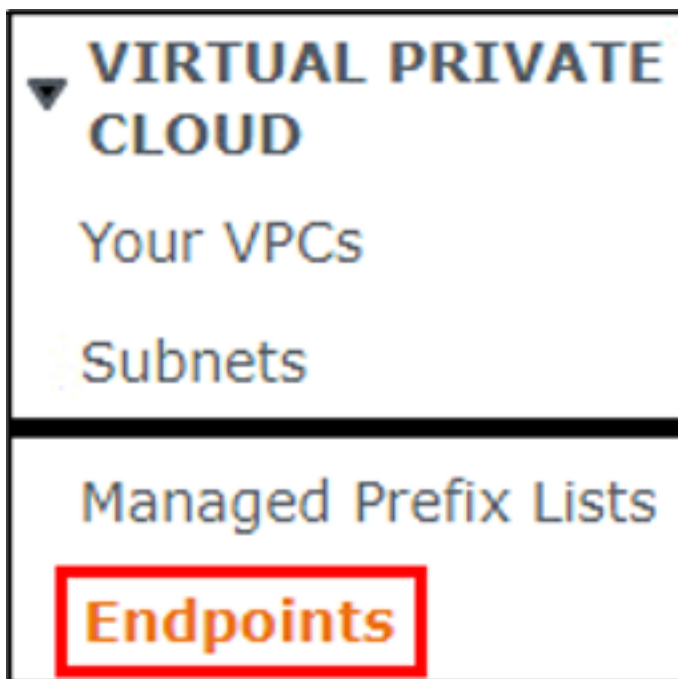
For information about deploying an interface VPC endpoint for Amazon S3, see [Access an AWS service using an interface VPC endpoint](#).

To deploy an SSM endpoint:

1. If you are not already there, go to the **VPC** service in the AWS console.



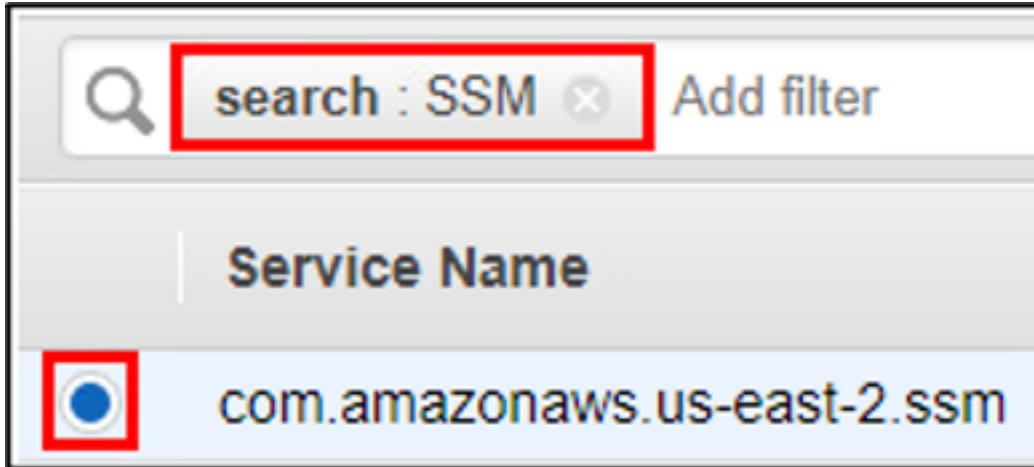
2. Under the **VIRTUAL PRIVATE CLOUD** menu, click **Endpoints**.



3. Click the **Create Endpoint** button.



4. For the **Service Name** section, enter **SSM** into the **search** field, then click the radio button for **com.amazonaws.<region>.ssm** to select it.



- 5. For the **VPC** field, select the appropriate VPC from the list.



- 6. After selecting the VPC, the associated **Availability Zone** will automatically be selected. Verify the private subnet for the VPC is selected for the **Subnet ID** field.



- 7. While not required, it is recommended to tag the resource to you to easily identify it. For the **Key** field, enter **Name**. For the **Value** field, enter a descriptive name for the resource.



- 8. Click the **Create endpoint** button.



- 9. Click the **Close** button.



The SSM endpoint has now be deployed for the private subnet.

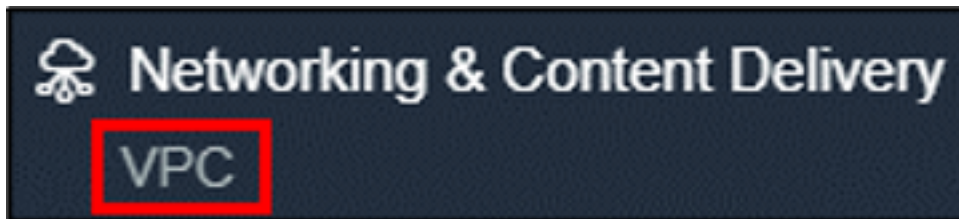
Next > You're ready to [deploy SSM messages endpoints](#).

Deploy SSM Messages Endpoint

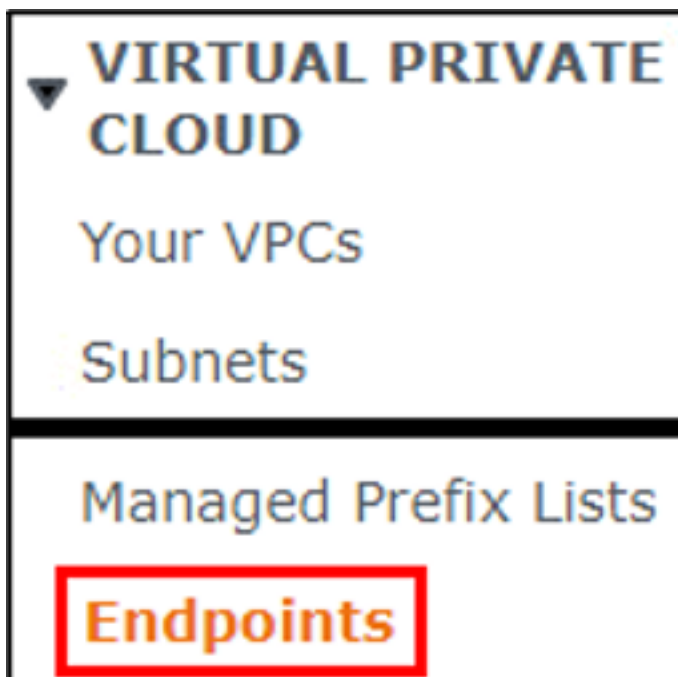
SSM is used during the AWS SaaS Connector deployment so connect to the EC2 instance and complete the VM configuration and claim process.

To deploy an SSM messages endpoint:

1. If you are not already there, go to the **VPC** service in the AWS console.



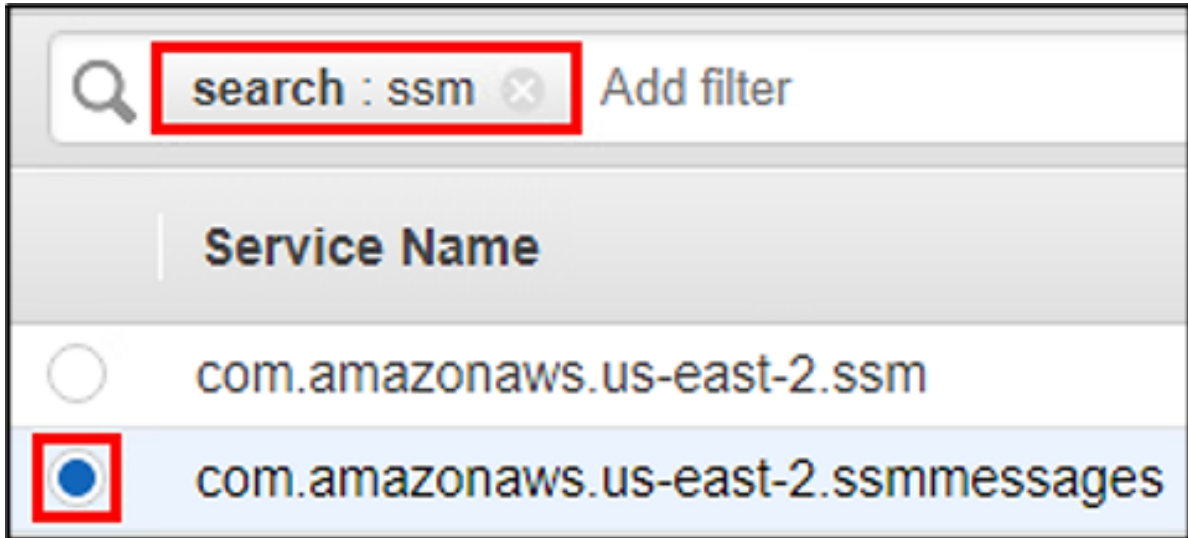
2. Under the **VIRTUAL PRIVATE CLOUD** menu, click **Endpoints**.



3. Click the **Create Endpoint** button.



- For the **Service Name** section, enter **SSM** into the search field, then click the radio button for **com.amazonaws.<region>.ssmmessages** to select it.



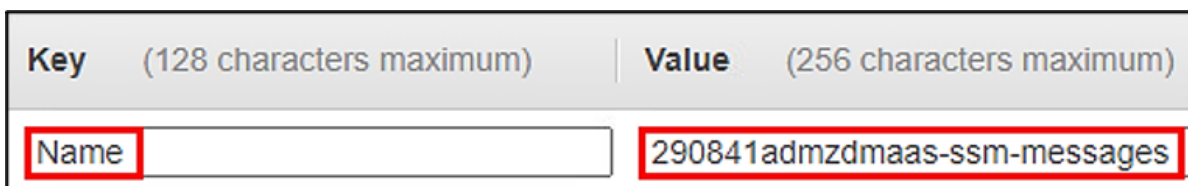
- For the **VPC** field, select the appropriate VPC from the list.



- After selecting the VPC, the associated **Availability Zone** will automatically be selected. Verify the private subnet for the VPC is selected for the **Subnet ID** field.



- While not required, it is recommended to tag the resource to you to easily identify it. For the **Key** field, enter **Name**. For the **Value** field, enter a descriptive name for the resource.



- Click the **Create endpoint** button.



9. Click the **Close** button.



The SSM messages endpoint has now be deployed for the private subnet.

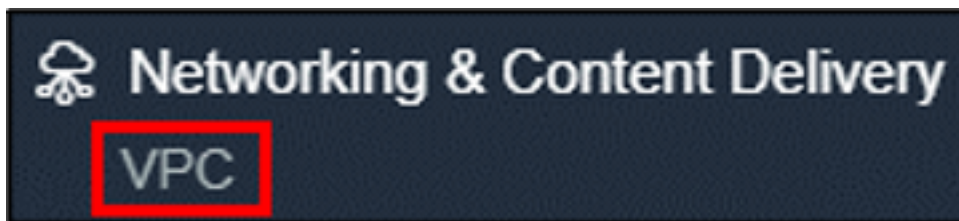
Next > You're ready to [deploy EC2 messages endpoint](#).

Deploy EC2 Messages Endpoint

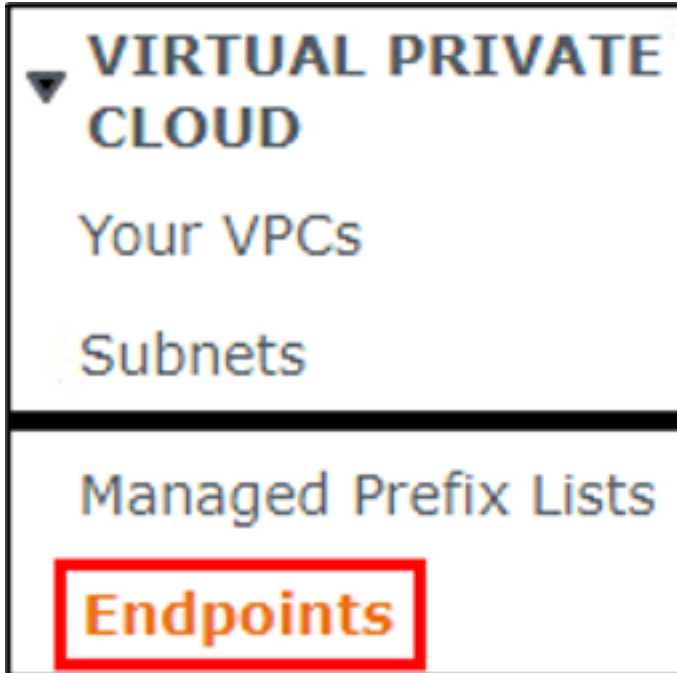
This topic provides instructions to deploy Amazon EC2 messages endpoint.

To deploy an EC2 messages endpoint:

1. If you are not already there, go to the **VPC** service in the AWS console.



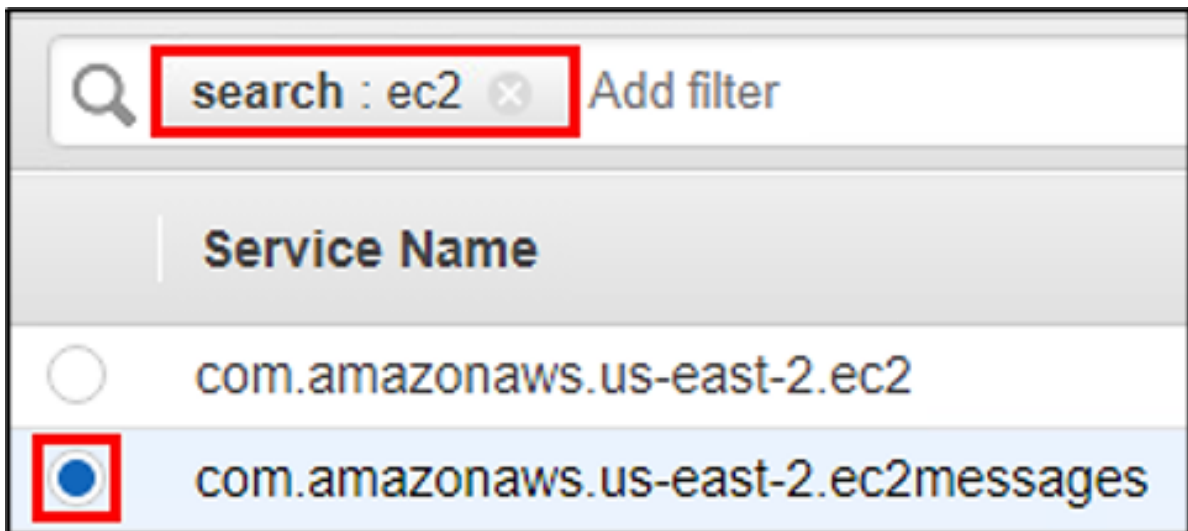
2. Under the **VIRTUAL PRIVATE CLOUD** menu, click **Endpoints**.



- 3. Click the **Create Endpoint** button.



- 4. For the **Service Name** section, enter **ec2** into the **search** field, then click the radio button for **com.amazonaws.<region>.ec2messages** to select it.



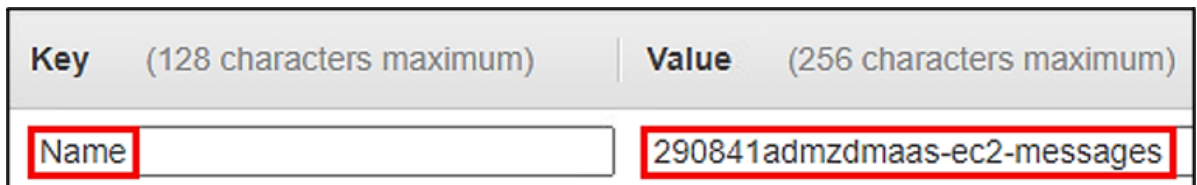
- 5. For the **VPC** field, select the appropriate VPC from the list.



- 6. After selecting the VPC, the associated **Availability Zone** will automatically be selected. Verify the private subnet for the VPC is selected for the **Subnet ID** field.



- 7. While not required, it is recommended to tag the resource to you to easily identify it. For the **Key** field, enter **Name**. For the **Value** field, enter a descriptive name for the resource.



- 8. Click the **Create endpoint** button.



- 9. Click the **Close** button.



The EC2 messages endpoint has now be deployed for the private subnet.

Next > You're now ready to [register an AWS source and deploy the AWS SaaS Connectors](#).

Register AWS Source and Deploy AWS SaaS Connector

Once the AWS configuration is prepared you will be ready to deploy the AWS SaaS Connector. This process will take roughly 15 minutes normally. Deploying the AWS SaaS Connector is a two-step process.

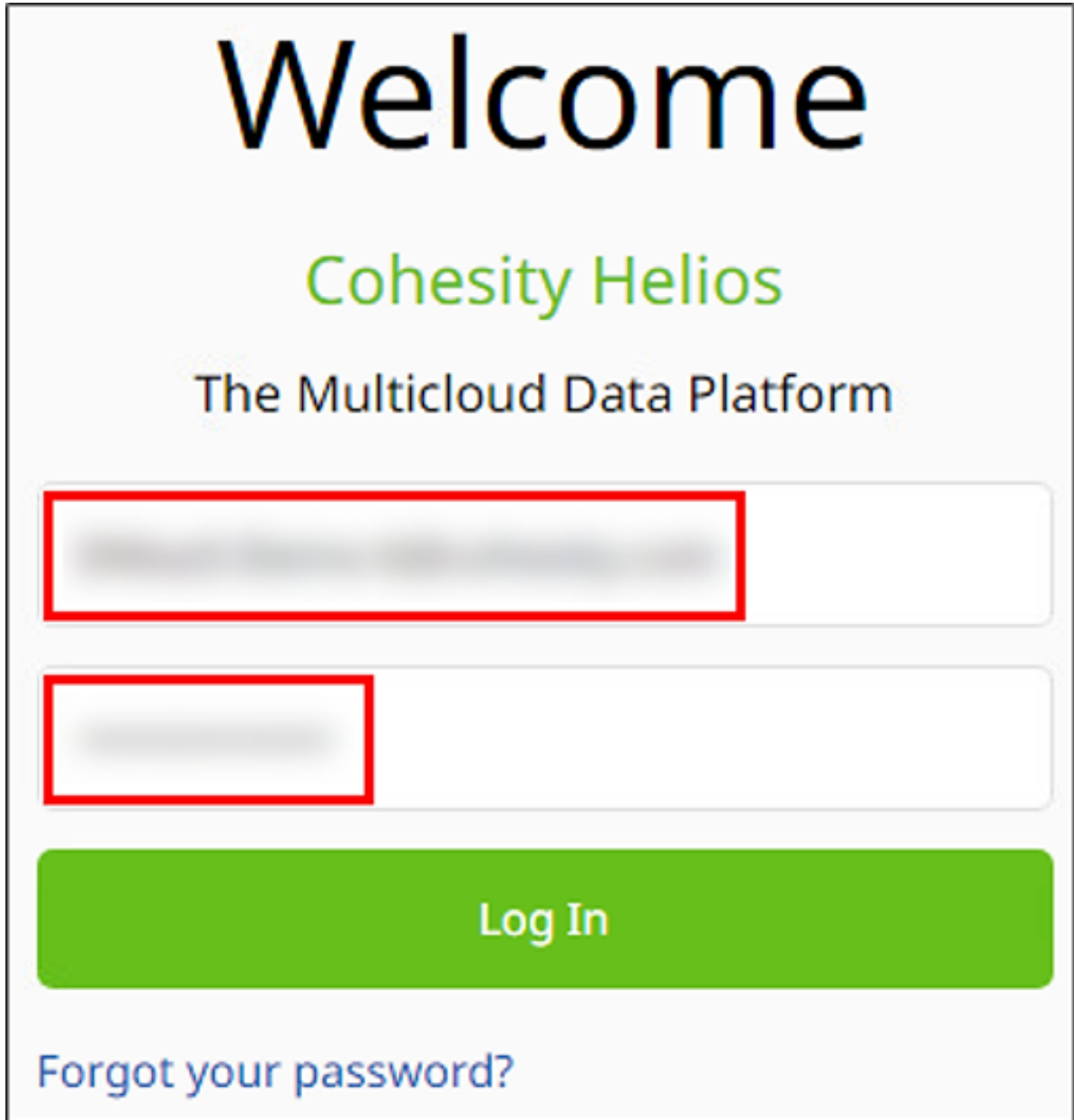
This topic covers the following:

Register AWS Source

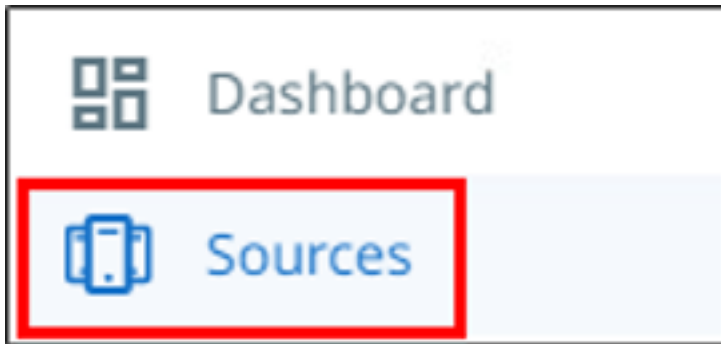
This section provides instructions to register an AWS source.

To register an AWS source:

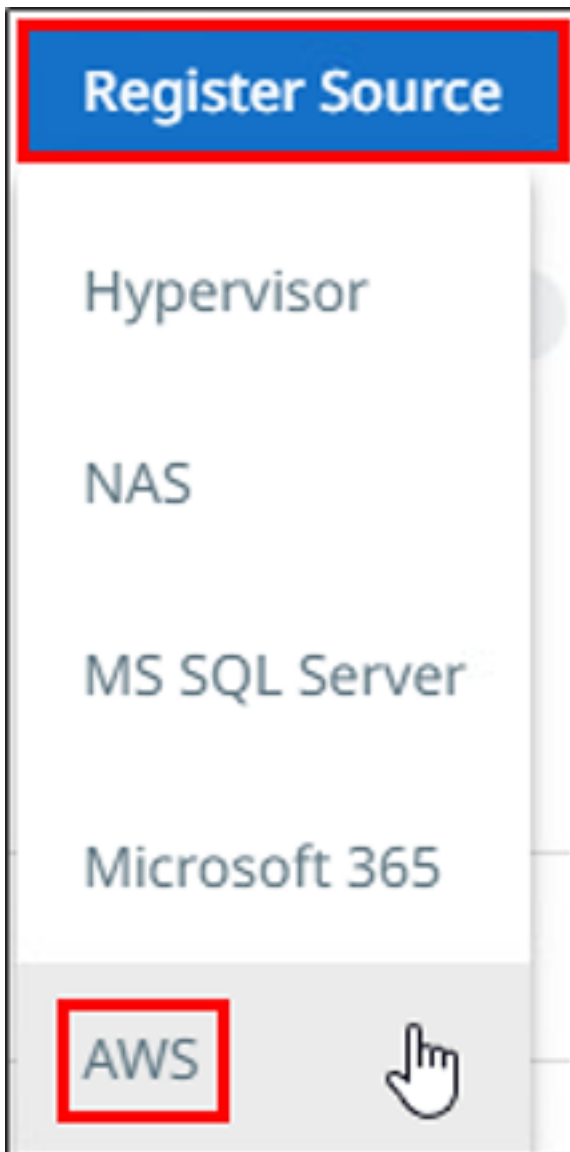
1. Log in to Helios and access the DataProtect portal.



2. Click **Sources** from the left-side menu.



3. Click the **Register Source** button, then click **AWS**.



4. Enter your AWS account ID number in the **AWS Account ID** field. Select the appropriate AWS region for the **Destination Region** field.

Enter AWS account details.

AWS Account ID Destination Region

5. The **Use this account as a backup source in DataProtect** toggle is enabled by default. Verify the **EC2** checkbox is selected. You can optionally deselect the RDS checkbox or leave it selected (the same can be done with the EC2 checkbox).

Use this account as a backup source in DataProtect

Enable this option to use Cohesity DataProtect to protect your Amazon EC2 instances and/or RDS databases

AWS Services

EC2 RDS

The **Use this account as a DR target in SiteContinuity** toggle is for use with the Cohesity SiteContinuity service. If you are deploying a SaaS Connector in support of this service, then you will need to enable this toggle.

Use this account as a DR target in SiteContinuity

Enable this option if you plan to use this AWS account as a disaster recovery (DR) target in Cohesity SiteContinuity. This ensures that the Cohesity role will have the necessary permissions to protect, failover, and failback objects from this account.

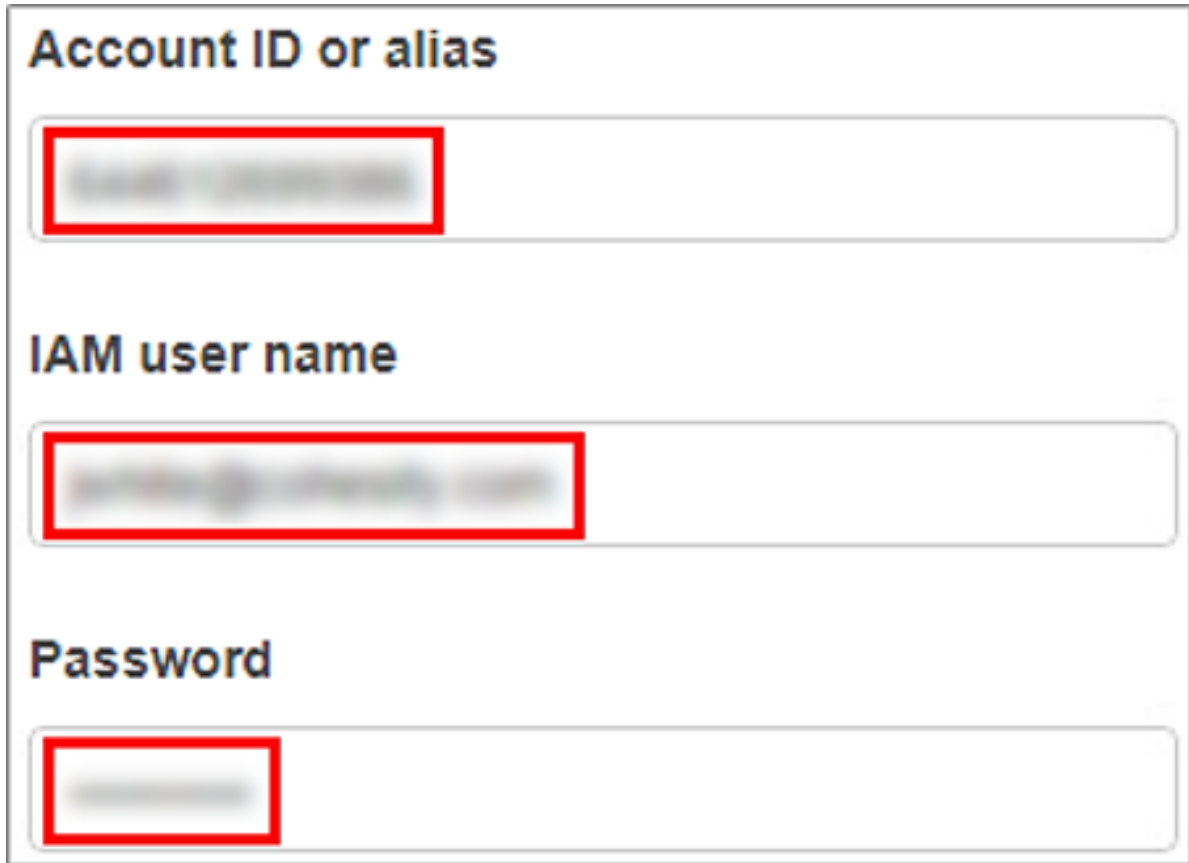
6. Click the **Next** button.



7. Click the **Download CloudFormation Template** link.

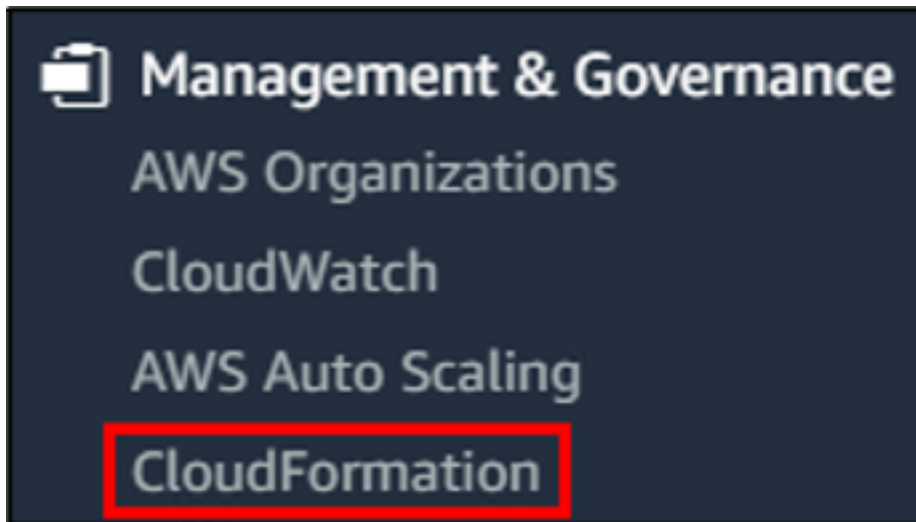


8. Open a new browser tab and log in to the AWS console.

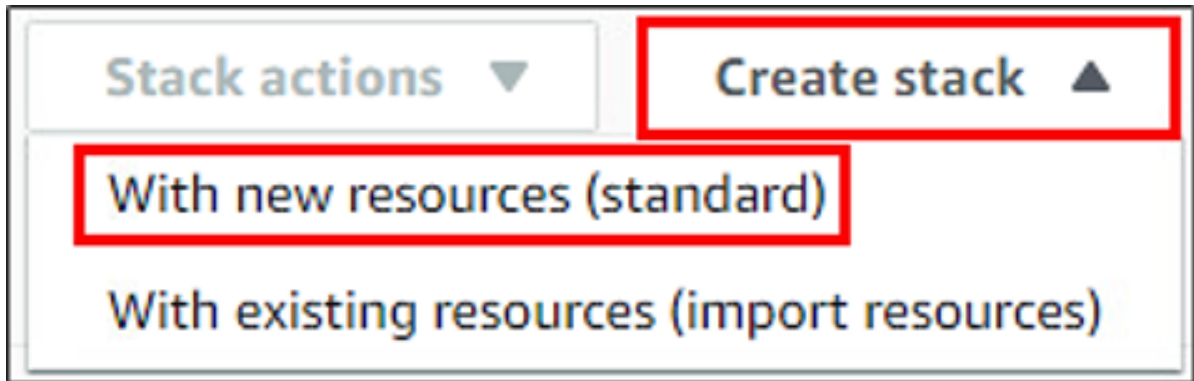


The screenshot shows the AWS console login page. It features three input fields: 'Account ID or alias', 'IAM user name', and 'Password'. Each field is highlighted with a red rectangular box. The text is in a dark blue font on a white background.

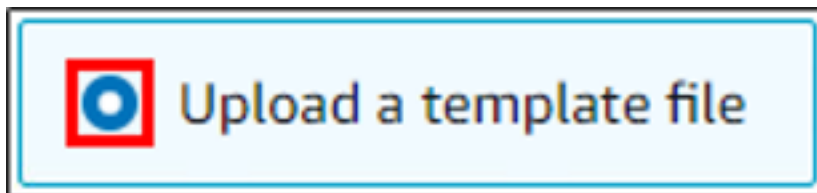
9. Navigate to the AWS CloudFormation service.



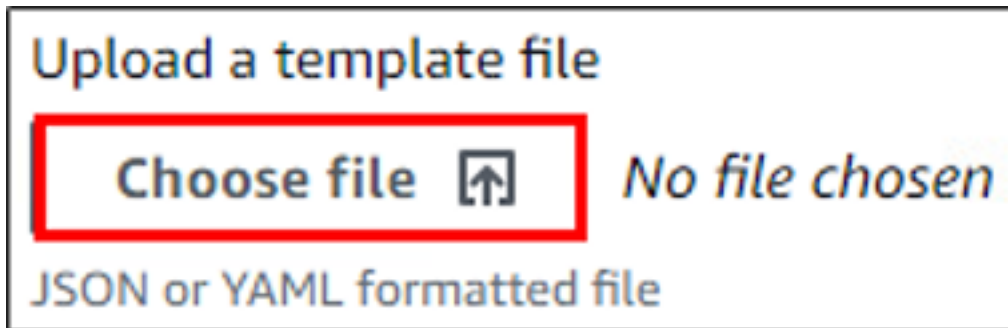
10. Click the **Create stack** button, then select **With new resources (standard)** from the list.



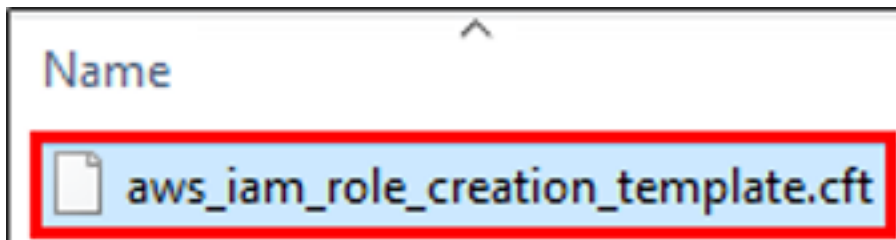
11. Click the **Upload a template file** radio button to select it.



12. Click the **Choose file** link.



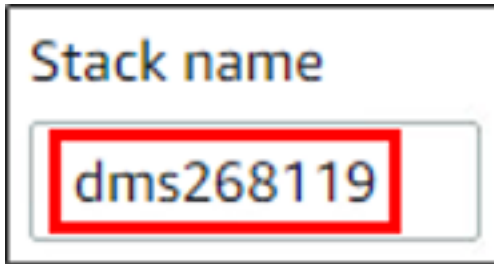
13. Navigate to the location that contains the downloaded CFT template. Select the **aws_iam_role_creation_template.cft** file.



14. Click the **Next** button.



- 15. Enter a descriptive name for the **Stack name** field.



A screenshot of a form field labeled "Stack name". The field contains the text "dms268119". The text and the field border are highlighted with a red rectangular box.

- 16. Click the **Next** button.



A screenshot of three navigation buttons: "Cancel", "Previous", and "Next". The "Next" button is highlighted with a red rectangular box.

- 17. Keep the default values for this page and click the **Next** button.



A screenshot of three navigation buttons: "Cancel", "Previous", and "Next". The "Next" button is highlighted with a red rectangular box.

- 18. Scroll to the bottom of the page. Click the **I acknowledge that AWS CloudFormation might create IAM resources with custom names** checkbox to select it.



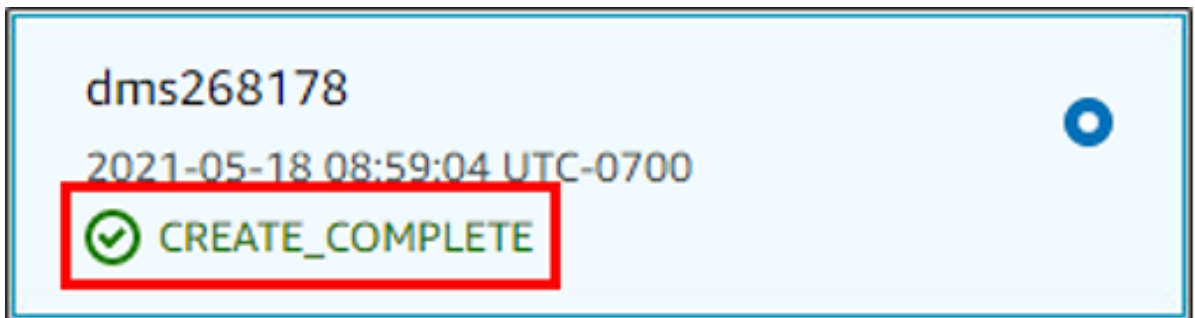
A screenshot of a checkbox with the text "I acknowledge that AWS CloudFormation might create IAM resources with custom names." The checkbox is checked and highlighted with a red rectangular box.

- 19. Click the **Create stack** button.



A screenshot of four navigation buttons: "Cancel", "Previous", "Create change set", and "Create stack". The "Create stack" button is highlighted with a red rectangular box.

- 20. Wait for the CloudFormation stack deployment to complete successfully. You can click the refresh buttons in the AWS console to update the status.



A screenshot of a stack deployment status card. The card displays the stack name "dms268178" and the timestamp "2021-05-18 08:59:04 UTC-0700". Below this, the status "CREATE_COMPLETE" is shown with a green checkmark icon. The status text and icon are highlighted with a red rectangular box.

21. After the stack is deployed, return to the Cohesity SaaS portal. Note that the AWS source registration now shows as **Account authenticated** to indicate the CloudFormation template completed the necessary AWS configuration.



22. Click the **Register** button.



The AWS source registration has been completed. In the current configuration DataProtect can support the option to protect EC2 instances (and RDS databases if selected during registration) via native AWS snapshots (EBS or RDS snapshots). This method of protection does not create Cohesity snapshots or result in egress of data from AWS.

To support Cohesity snapshots of EC2 instances, an AWS SaaS Connector must be deployed.

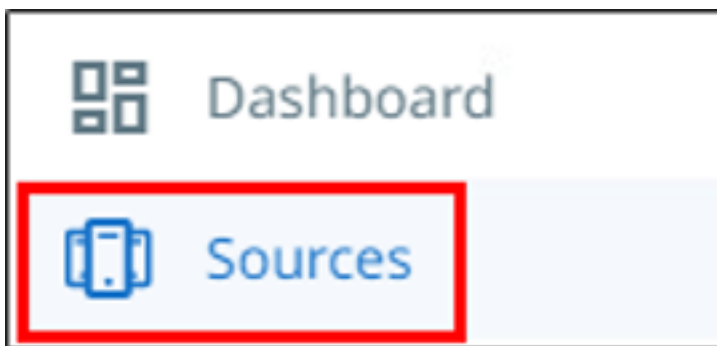
Next > You're ready to [deploy an AWS SaaS connector](#).

Deploy AWS SaaS Connector

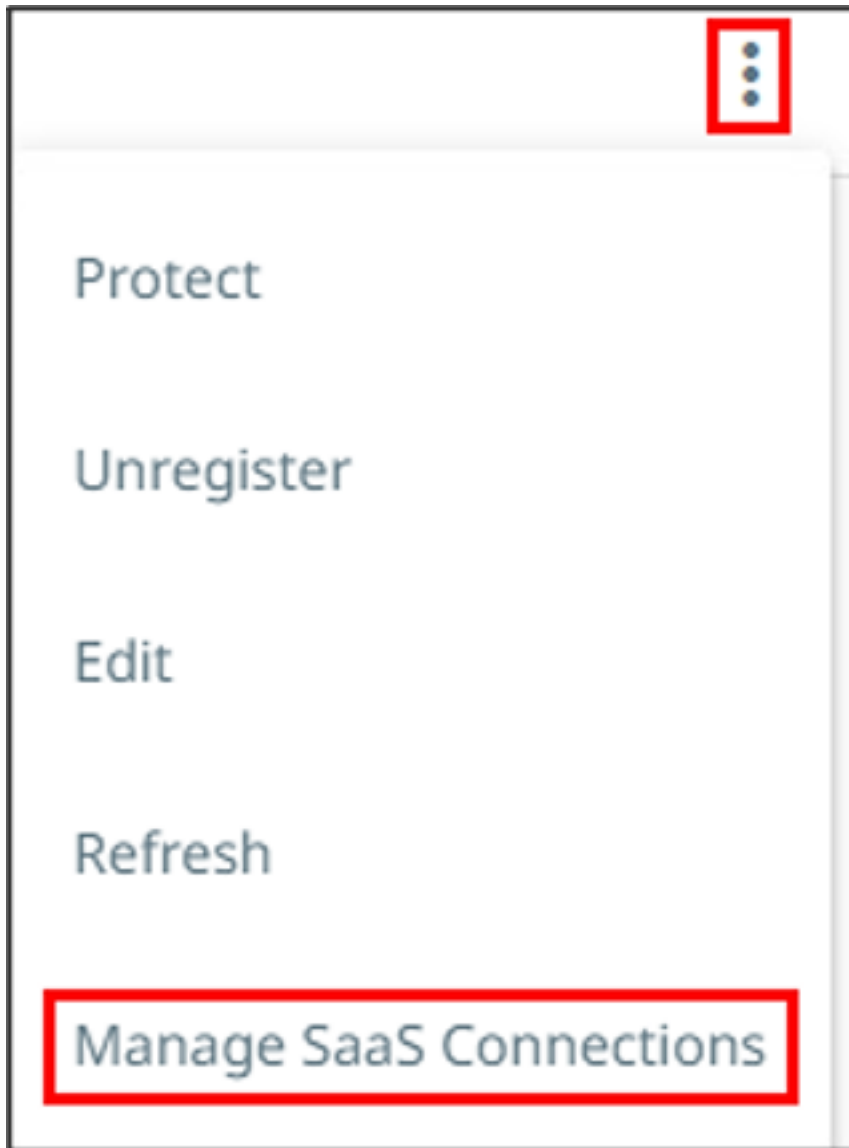
This topic provides instructions to deploy an AWS SaaS connector.

To deploy an AWS SaaS Connector:

1. From the Cohesity SaaS portal, if you are not already there, go to the **Sources** page.



2. Hover over the AWS source entry, click the **three dots (ellipses)** located to the right of the entry, then click **Manage SaaS Connections**.

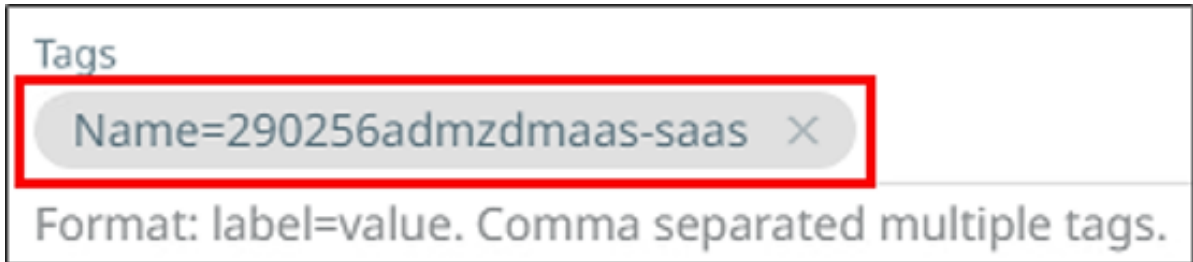


- 3. Select the appropriate AWS region for the **Region** field. A minimum of one connector must be deployed and is recommended to start. Additional connectors can be deployed if desired. For the **Subnet** field, select the private subnet that has been configured for the VPC. For the **Network Security Groups** field, select the appropriate network security group for the VPC.

Region	Number of Connectors
us-east-2	1
Subnet	Network Security Groups
290256admzdmaas (subnet-0fda41...	default (sg-07122e42764d91593)

Note: It is critical that you select the correct subnet for the VPC. If you are using the [VPC with Private Subnet and NAT Gateway](#) configuration, you must select the private subnet for the VPC. If you are using the [VPC with Public Subnet](#) configuration, you must select the public subnet for the VPC.

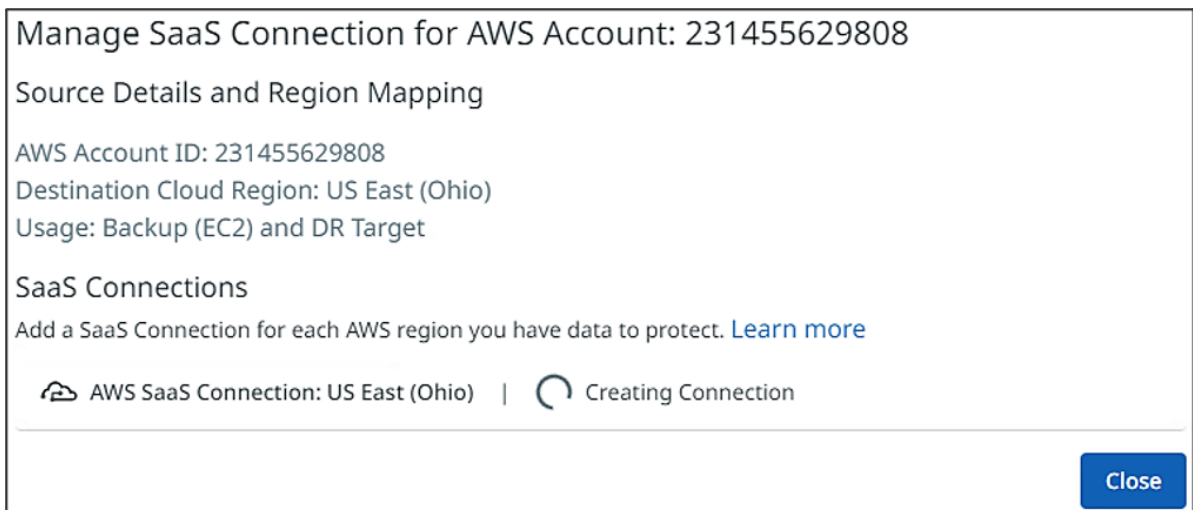
- While not required, it is recommended that you add a tag for the resource. For the **Tags** field, enter **Name=<descriptive_name>**. This name tag will be used for the SaaS Connector EC2 instance that will be deployed.



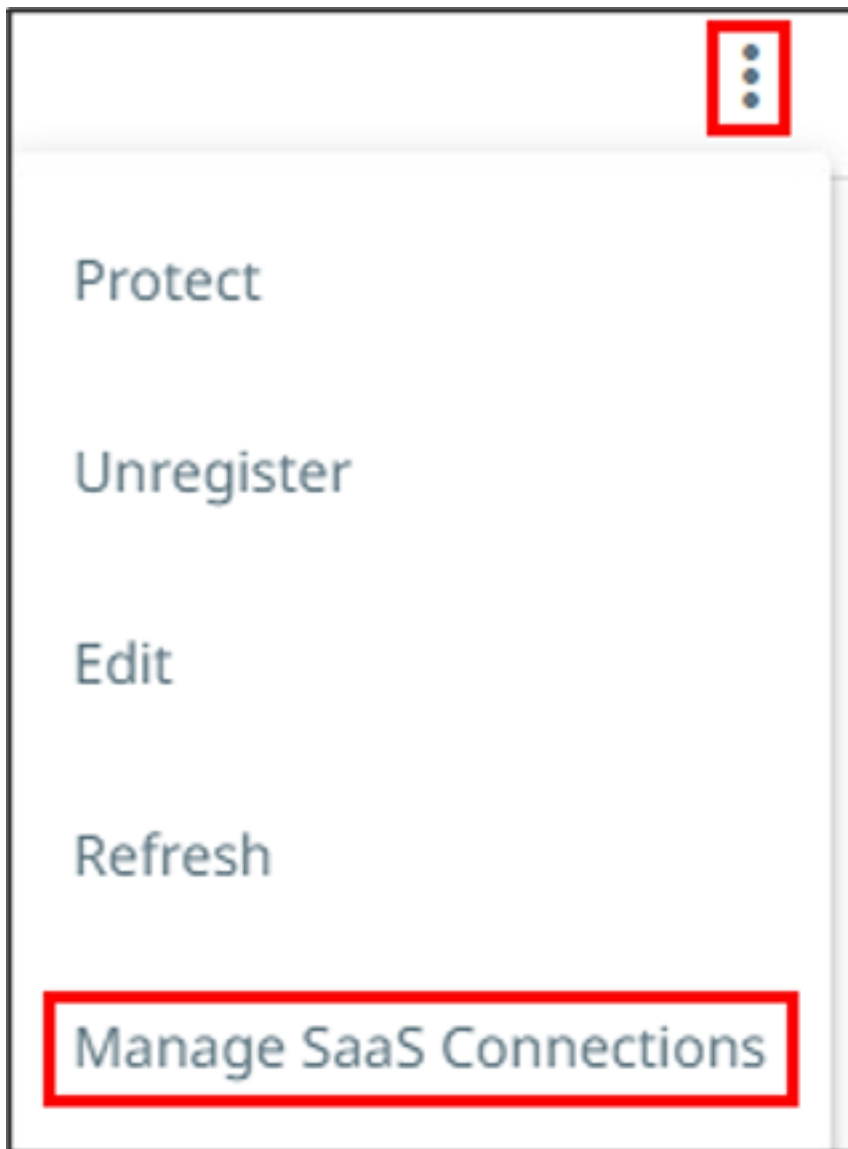
- Click the **Update** button.



- The **Configure SaaS Connection for AWS Account** window will display the status **Creating Connection**. You can monitor the progress of the AWS SaaS Connector deployment. Please be patient as this can take roughly 10-15 minutes to complete normally.



You can close this window if desired. To return to this window and check on the status simply click the three dots (ellipses) located to the right of the entry, then click **Manage SaaS Connections**.



Please be patient as this process normally takes 15-20 minutes to complete. If this last for longer than 30 minutes, something has gone wrong and you should contact [Cohesity support](#).

During this time, you can also go to the AWS console EC2 service. Go to Instances and observe the current state of the AWS SaaS Connector instance as it is deployed. While this can be helpful, the only information that can be learned from this is if the EC2 instance has been successfully launched by AWS. Any further information can only be learned in the Cohesity SaaS portal.

Name	Instance ID	Instance state	Instance type	Status check
290841admzdmaas-saas	i-02ea22bd958c96361	Running	m5.xlarge	2/2 checks passed

You will need to return to the Cohesity SaaS portal to wait for the process to complete.

Note: The connection status for the AWS SaaS Connector may temporarily show **Not Connected** in red font. This is normal to see when the deployment process is completing. The status indicates the deployment and configuration of the SaaS Connector is complete and the SaaS Connector claim process is in progress.

- When complete, the status **Connected** with a green check mark will be displayed.


Manage SaaS Connection for AWS Account: 231455629808

Source Details and Region Mapping

AWS Account ID: 231455629808
 Destination Cloud Region: US East (Ohio)
 Usage: Backup (EC2) and DR Target

SaaS Connections

Add a SaaS Connection for each AWS region you have data to protect. [Learn more](#)

 AWS SaaS Connection: US East (Ohio) | ✔ Connected

- Click the **Close** button.



The AWS SaaS Connector has been successfully deployed. You can now proceed with protection of EC2 instances in AWS using the **Cohesity snapshot** option.

Next > Your new AWS SaaS Connection is now available to use when you [protect your AWS EC2 instances](#).