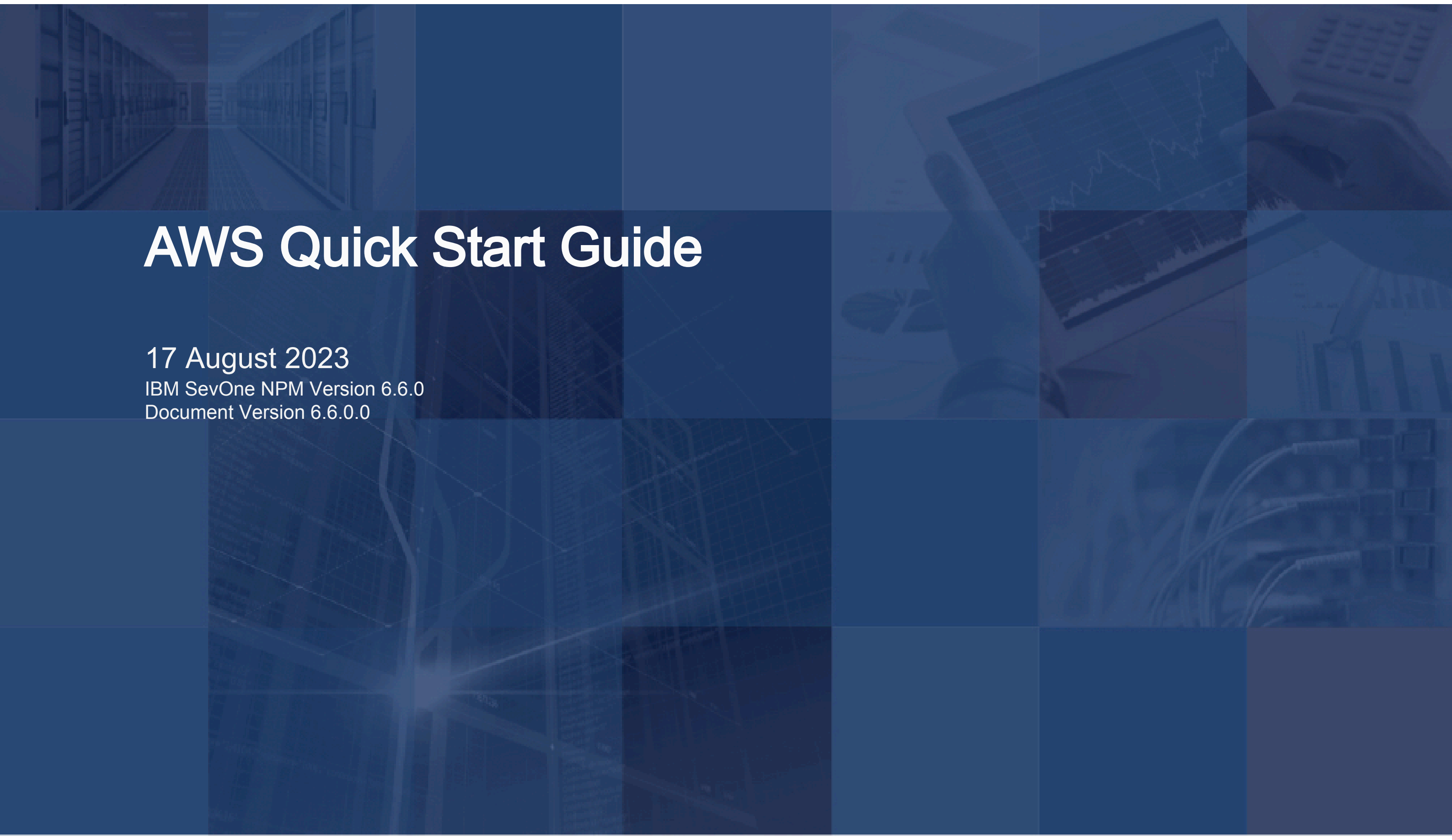


# AWS Quick Start Guide

17 August 2023

IBM SevOne NPM Version 6.6.0

Document Version 6.6.0.0

# Table of Contents

SevOne Documentation

All documentation is available from the IBM SevOne Support customer portal.

# 1 About

SevOne's AWS plugin allows you to collect devices, metadata, and metrics from the AWS environment right out-of-the-box. The plugin makes use of AWS CloudWatch metric streams for overtime data as well as for API calls for metadata enrichment. The AWS plugin allows you to monitor multiple different AWS accounts and regions.

## 2 Device Creation

SevOne NMS is configured, by default, to regularly call AWS APIs to retrieve information about the devices and then, automatically create devices for each AWS resource.

# 3 Required Infrastructure in AWS

To monitor an AWS account, the following necessary infrastructure in AWS must be configured.

- Collector Role ARN
- SQS Queue
- Metric Stream
- Firehose
- S3 Bucket

This infrastructure is used to collect metrics for your AWS environment and make them available to SevOne NMS. Ensure that this is configured before continuing with the AWS plugin.

> ⓘ AWS Infrastructure can be set up in two ways: *Terraform or AWS management console*. Terraform is the **recommended** way that we will explain in details below. Please refer to section *Appendix* > Deploy AWS Resources using AWS Console to learn how to setup via AWS management console.

# 4 Deploy with Terraform

Terraform is the recommended way to configure the AWS resources. This allows for the quickest startup time while ensuring that resources are configured exactly as intended. To deploy and configure the required resources, a set of Terraform files can be found in **/opt/SevOne-aws-collector/terraform** directory. In order to run the Terraform files to create the resources, a role has been defined to maintain a least privileged posture. Let's refer to this role as *Infrastructure Role*.

## 4.1 Create a role for Terraform to use

> ⓘ A role can be created in two ways: *Terraform or AWS management console*. Terraform is the **recommended** way that we will explain in details below. Please refer to section *Appendix >* Create a Role via AWS Console for Terraform to use to learn how to create a role via AWS management console.

1. Using **ssh**, log in to SevOne NMS appliance as **root**.

```
$ ssh root@<NMS appliance>
```

2. Change directory to **/opt/SevOne-aws-collector/terraform/envs/infrastructure_role**.
3. Update the following values in **terraform.tfvars**.

> ⚠ Variable **prefix** must be,
> - between 1 and 20 characters
> - contain only lowercase letters, digits, or hyphens
> - start and end with letters or digits

```
account_id         = <YOUR AWS ACCOUNT NUMBER>
collector_user_arn = <USER ARN THAT WILL BE ABLE TO ASSUME THE ROLE>
prefix             = <PREFIX TO UNIQUELY IDENTIFY RESOURCES>
```

**Empty 'terraform.tfvars' file published with examples in comments**

```
## Your 12 digit AWS account number
# account_id                = 012345678901

## AWS IAM User ARN that will be used to run the collector
# collector_user_arn        = "arn:aws:iam::012345678901:user/person@company.com"

## Prefix to uniquely identify resources that are created in AWS
## - Must be all lowercase due to S3 bucket naming restrictions
# prefix                    = "sevone"
```

4. Set the following environment variables.

```
export AWS_ACCESS_KEY_ID="mykey"
export AWS_SECRET_ACCESS_KEY="mysecret"
export AWS_REGION="us-east-1"
```

5. Apply the terraform files.

```
terraform init
terraform plan
terraform apply
```

⚠ Please make note of output value, **infrastructure_role_arn**, as it will be required in section Run Terraform to deploy AWS resources below.

## 4.2 Run Terraform to deploy AWS resources

1. Using **ssh**, log in to SevOne NMS appliance as **root**.

```
$ ssh root@<NMS appliance>
```

2. Change directory to **/opt/SevOne-aws-collector/terraform/envs/collector_infrastructure**.
3. Update the following values in **terraform.tfvars**.

⚠ Variable **prefix** must be,
- between 1 and 20 characters
- contain only lowercase letters, digits, or hyphens
- start and end with letters or digits

```
account_id               = [YOUR AWS ACCOUNT NUMBER]
collector_user_arn       = [USER ARN THAT WILL BE ABLE TO ASSUME ROLE]
infrastructure_role_arn = [ROLE ARN FOR INFRASTRUCTURE ROLE] (created in last step)
regions                  = [ARRAY OF REGIONS TO CREATE RESOURCES IN]
prefix                   = [PREFIX TO UNIQUELY IDENTIFY RESOURCES]
```

**Empty 'terraform.tfvars' file published with examples in comments**

```
## Your 12 digit AWS account number
# account_id              = 012345678901

## AWS IAM User ARN that will be used to run the collector
# collector_user_arn      = "arn:aws:iam::012345678901:user/person@company.com"

## AWS IAM Role ARN that will be used to install infrastructure
## - This can be found in the output of applying the `infrastructure_role` Terraform config
# infrastructure_role_arn = "arn:aws:iam::012345678901:role/sevone_infrastructure_role"

## List of regions that you want to monitor
# regions                 = ["us-east-1", "us-west-1"]

## Prefix to uniquely identify resources that are created in AWS
## - Must be all lowercase due to S3 bucket naming restrictions
```

```
# prefix                    = "sevone"
```

⚠ When adding more than one region to create the resources in, you can add it in **terraform.tfvars** file as,

**Example**

```
regions = ["us-east-1", "eu-central-1"]
```

4. Apply the terraform files.

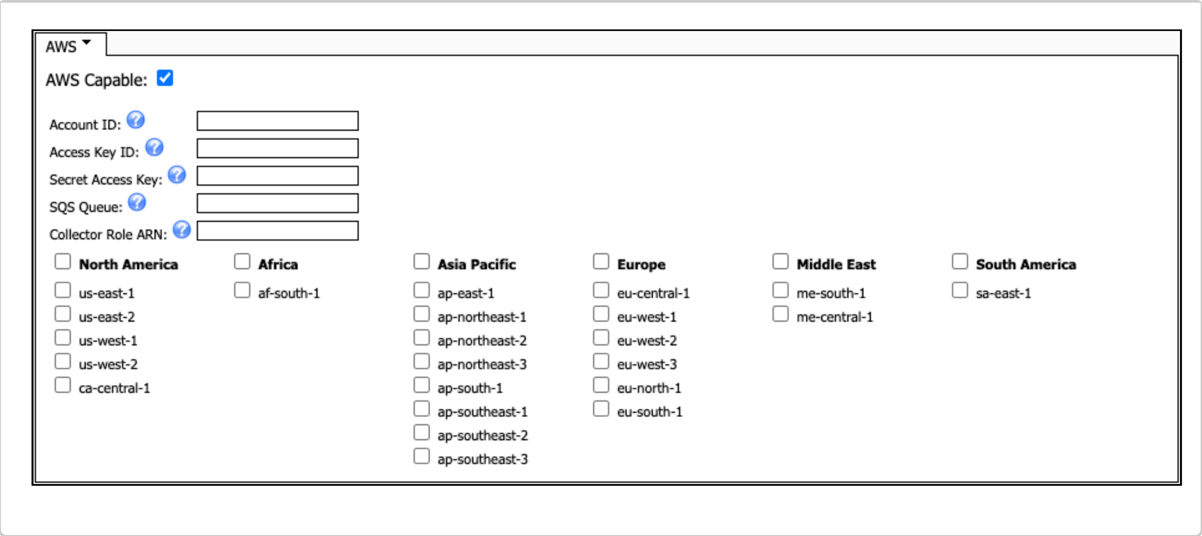
```
cd gen
terraform init
terraform plan
terraform apply
cd ..
terraform init
terraform plan
terraform apply
```

5. Details from the following output will be required to create a device in SevOne NMS.
    a. account_id
    b. collector_role_arn
    c. sqs_queue
    d. regions

# 5 Enable AWS Plugin

Execute the following steps to monitor an AWS account. This will automatically create devices and collect metrics for various AWS resources within selected regions for that account.

1. To access the Device Manager from the navigation bar, click the **Devices** menu and select **Device Manager**.
2. Either add a device with the AWS plugin or edit an existing device to enable the AWS plugin.
    - Click **Add Device** to display the **New Device** page.
    - Click the wrench icon under the **Actions** column to display the **Edit Device** page.
3. Click the plugin drop-down. By default, it is set to **SNMP**. Select **AWS**.



AWS ▾

AWS Capable: ☑

Account ID:

Access Key ID:

Secret Access Key:

SQS Queue:

Collector Role ARN:

☐ **North America**
- ☐ us-east-1
- ☐ us-east-2
- ☐ us-west-1
- ☐ us-west-2
- ☐ ca-central-1

☐ **Africa**
- ☐ af-south-1

☐ **Asia Pacific**
- ☐ ap-east-1
- ☐ ap-northeast-1
- ☐ ap-northeast-2
- ☐ ap-northeast-3
- ☐ ap-south-1
- ☐ ap-southeast-1
- ☐ ap-southeast-2
- ☐ ap-southeast-3

☐ **Europe**
- ☐ eu-central-1
- ☐ eu-west-1
- ☐ eu-west-2
- ☐ eu-west-3
- ☐ eu-north-1
- ☐ eu-south-1

☐ **Middle East**
- ☐ me-south-1
- ☐ me-central-1

☐ **South America**
- ☐ sa-east-1

4. Select the **AWS Capable** check box.
5. In the **Account ID** field, enter the ID of the account you want to monitor and collect data from.
6. In the **Access Key ID** field, enter the access key ID created for monitoring this account. For additional details, please refer to https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_access-keys.html.
7. In the **Secret Access Key** field, enter the secret access key created for monitoring this account. For additional details, please refer to https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_access-keys.html.
8. In the **SQS Queue** field, enter the SQS Queue name that AWS plugin will listen to for metric stream events. For additional details, please refer to section Required Infrastructure in AWS.
9. In the **Collector Role ARN** field, enter the ARN of the IAM role that the collector will assume. For additional details, please refer to section Required Infrastructure in AWS.
10. Select / deselect the column headers or AWS region codes you want to collect data from.
    a. Select or deselect a column header (for example, North America) to enable or disable collection for all regions underneath that header.
    b. Select or deselect an AWS region code (for example, us-east-1) to enable or disable collection for that region.
11. Click **Save As New** to save the current changes as a **New Device**, or click **Save** to confirm the changes in the **Edit Device** page.
12. When the desired changes have been saved, click the **Cancel** button to return to the **Device Manager** page.

⚠ Once the device is created, collection of AWS resources starts on SevOne NMS.

# 6 Create TopN View in SevOne NMS to view AWS Resources

SevOne NMS contains a package with AWS-specific TopN views. To import, execute the following command.

```
$ SevOne-import --file /opt/SevOne-aws-collector/topn.spk
```

⚠ The import will fail unless the AWS collector has run successfully at least once.

# 7 OOTB Reports

The following out-of-the-box (OOTB) reports are available as part of SevOne Data Insight 6.5 and above.

- AWS Direct Connect - report showing AWS Direct Connect inventory and network performance.
- AWS EC2 Report- report showing AWS EC2 inventory, CPU, disk, and network performance.
- AWS NAT Gateway - report showing AWS NAT Gateway inventory, throughput and connection statistics.
- AWS S3 Report - report showing AWS S3 inventory and bucket statistics.
- AWS Transit Gateways - report showing AWS Transit Gateway inventory, network traffic, and drops.

# 8 Update AWS Infrastructure

> ⚠ SevOne NMS v6.6.0 supports more AWS resources to be monitored than v6.5.x. If you would like to monitor these newly supported AWS resources, please follow the steps below to update your AWS Infrastructure after an SevOne NMS upgrade from v6.5.x to v6.6.0.

## 8.1 Update a role for Terraform to use

1. Using **ssh**, log in to SevOne NMS appliance as **root**.

```
$ ssh root@<NMS appliance>
```

2. Change directory to **/opt/SevOne-aws-collector/terraform/envs/infrastructure_role**.
3. Apply the terraform files.

```
terraform init
terraform plan
terraform apply
```

> ⚠ Please make note of output value, **infrastructure_role_arn**, as it will be required in section Run Terraform to deploy AWS resources below.

## 8.2 Run Terraform to deploy AWS resources

1. Using **ssh**, log in to SevOne NMS appliance as **root**.

```
$ ssh root@<NMS appliance>
```

2. Change directory to **/opt/SevOne-aws-collector/terraform/envs/collector_infrastructure**.
3. Apply the terraform files.

```
cd gen
terraform init
terraform plan
terraform apply
cd ..
terraform init
terraform plan
terraform apply
```

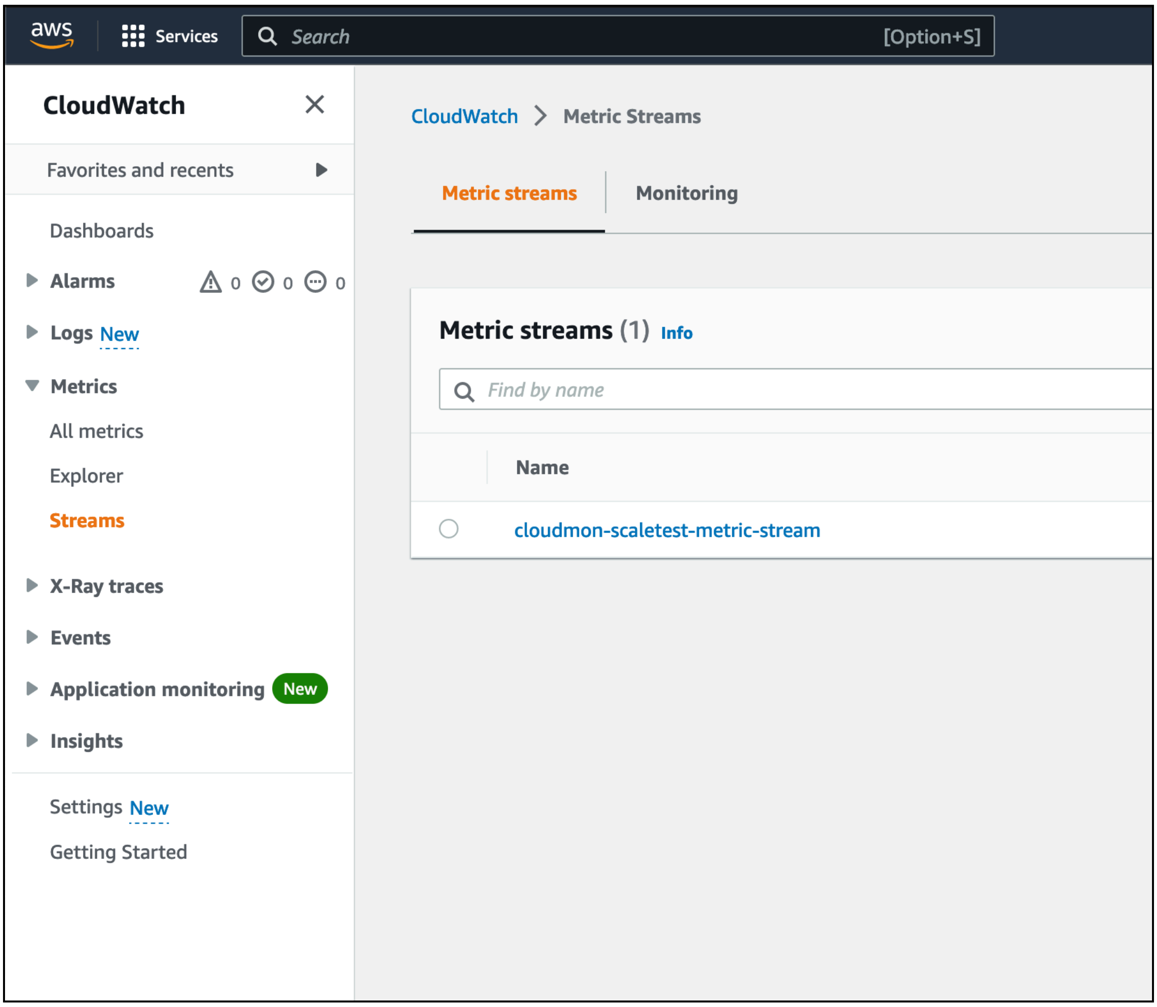
# 9 Appendix

## 9.1 Deploy AWS Resources using AWS Console

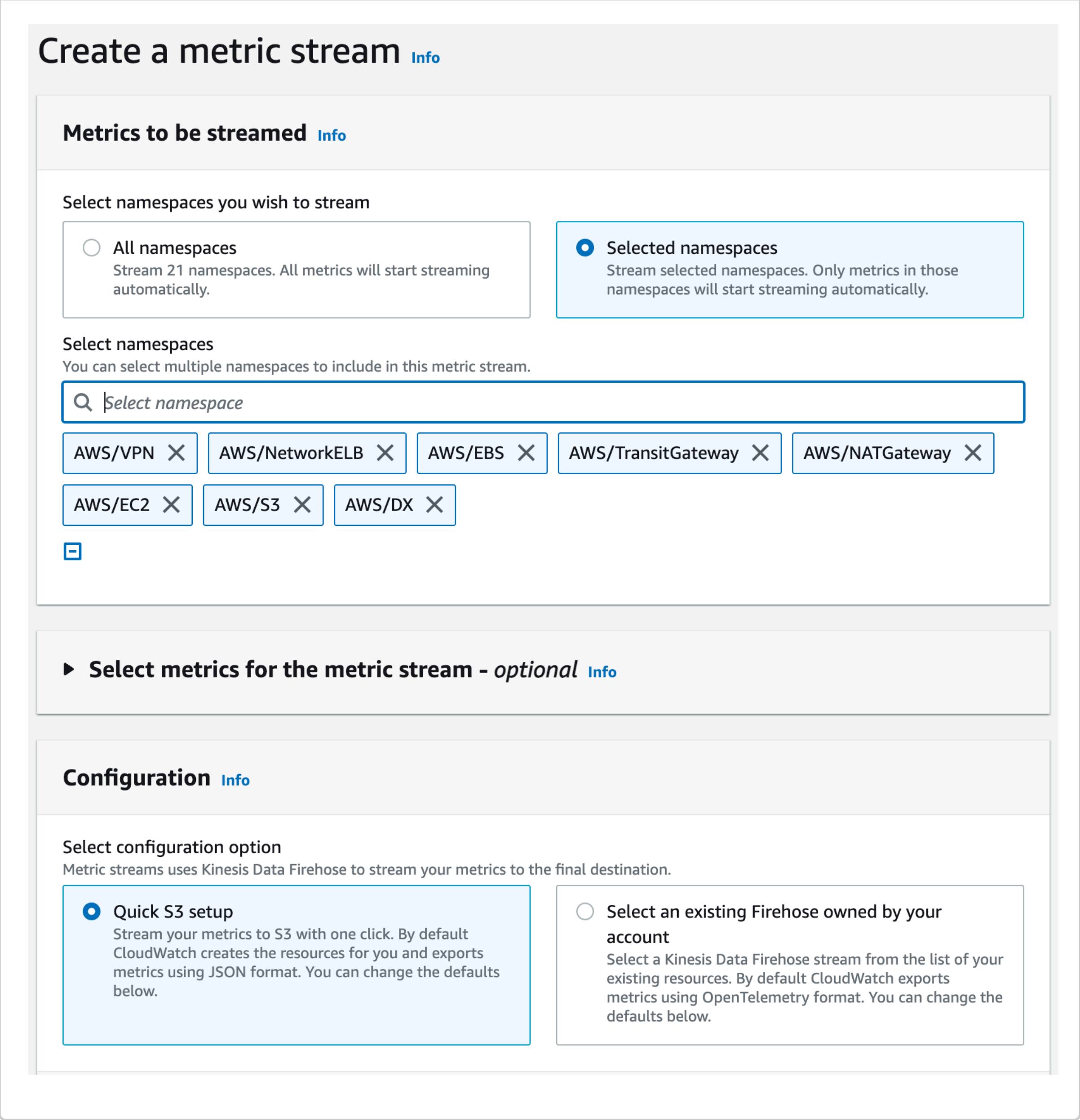
AWS console allows you to create the AWS resource using the AWS console. While it is recommended to use Terraform, this is a viable option when that is not possible.

### 9.1.1 Create a Metric Stream

1. Navigate to CloudWatch > Metric Streams.



2. Click Create metric stream to launch the wizard.
3. Select the following namespaces.
   a. AWS/EC2
   b. AWS/S3
   c. AWS/NATGateway
   d. AWS/TransitGateway
   e. AWS/DX
   f. AWS/EBS
   g. AWS/NetworkELB
   h. AWS/VPN
4. Select Quick S3 setup check box.



Create a metric stream Info

Metrics to be streamed Info

Select namespaces you wish to stream

○ All namespaces
Stream 21 namespaces. All metrics will start streaming automatically.

● Selected namespaces
Stream selected namespaces. Only metrics in those namespaces will start streaming automatically.

Select namespaces
You can select multiple namespaces to include in this metric stream.

Select namespace

AWS/VPN ✕ AWS/NetworkELB ✕ AWS/EBS ✕ AWS/TransitGateway ✕ AWS/NATGateway ✕ AWS/EC2 ✕ AWS/S3 ✕ AWS/DX ✕

▶ Select metrics for the metric stream - *optional* Info

Configuration Info

Select configuration option
Metric streams uses Kinesis Data Firehose to stream your metrics to the final destination.

● Quick S3 setup
Stream your metrics to S3 with one click. By default CloudWatch creates the resources for you and exports metrics using JSON format. You can change the defaults below.

○ Select an existing Firehose owned by your account
Select a Kinesis Data Firehose stream from the list of your existing resources. By default CloudWatch exports metrics using OpenTelemetry format. You can change the defaults below.

5. Rename the metric stream.
6. Click Create metric stream to complete the configuration.

### 9.1.2 Set up S3 Event Notifications

In https://docs.aws.amazon.com/AmazonS3/latest/userguide/ways-to-add-notification-config-to-bucket.html, follow steps 1. and 3a. to send all object create events to the SQS queue.

### 9.1.3 Create a Collector Role

Create a IAM role for the AWS plugin to use with the following policies.

#### 9.1.3.1 Policy *'aws_collector_directconnect_policy'*

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Action": [
                "directconnect:DescribeLocations",
                "directconnect:DescribeConnections"
            ],
            "Resource": "*"
        }
    ]
}
```

#### 9.1.3.2 Policy *'sevone_collector_cloudwatch_policy'*

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Action": "cloudwatch:GetMetricData",
            "Resource": "*"
        }
    ]
}
```

#### 9.1.3.3 Policy *'sevone_collector_ec2_policy'*

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:DescribeNatGateways",
                "ec2:DescribeTransitGateways",
                "ec2:DescribeVolumes",
                "ec2:DescribeVolumeStatus",
                "ec2:DescribeVpnConnections"
            ],
            "Resource": "*"
        }
    ]
}
```

### 9.1.3.4 Policy 'sevone_collector_elasticloadbalancing_policy'

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Action": [
                "elasticloadbalancing:DescribeLoadBalancers"
            ],
            "Resource": "*"
        }
    ]
}
```

### 9.1.3.5 Policy *'sevone_collector_s3_policy'*

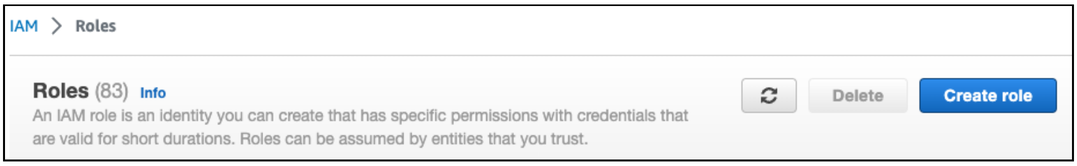
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets",
                "s3:GetObject"
            ],
            "Resource": "*"
        }
    ]
}
```

### 9.1.3.6 Policy *'sevone_collector_sqs_policy'*

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Action": [
                "sqs:ReceiveMessage",
                "sqs:GetQueueUrl",
                "sqs:DeleteMessage"
            ],
            "Resource": "*"
        }
    ]
}
```

## 9.2 Create a Role via AWS Console for Terraform to use

1. To create a new role,
   a. from **Services** menu, select **IAM**.
   b. select **Roles** from **Access management** menu.
   c. select **Create** role.

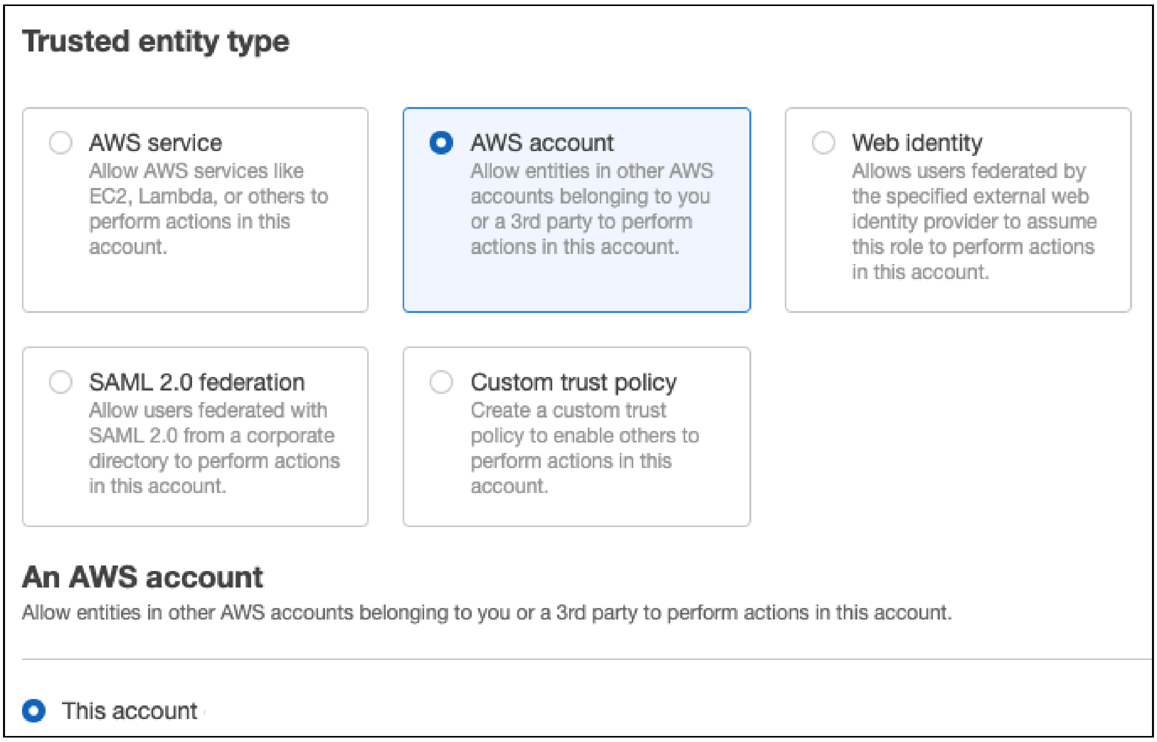


IAM > Roles

**Roles** (83) Info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

2. Allow Trust from **This account**.



**Trusted entity type**

○ AWS service
Allow AWS services like EC2, Lambda, or others to perform actions in this account.

● AWS account
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

○ Web identity
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.

○ SAML 2.0 federation
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

○ Custom trust policy
Create a custom trust policy to enable others to perform actions in this account.

**An AWS account**

Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

● This account

3. Add policies.

**Example**

```
{
    "Version": "2012-10-17",
    "Statement":[
        {
            "Action":[
                "cloudwatch:TagResource",
                "cloudwatch:ListTagsForResource"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid":""
        },
        {
         "Action":[
                "firehose:ListTagsForDeliveryStream",
                "firehose:DescribeDeliveryStream",
                "firehose:DeleteDeliveryStream",
                "firehose:CreateDeliveryStream"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": ""
        },
        {
            "Action":[
                "iam:TagRole",
                "iam:PutRolePolicy",
                "iam:PassRole",
                "iam:ListRolePolicies",
                "iam:ListInstanceProfilesForRole",
                "iam:ListAttachedRolePolicies",
                "iam:GetRolePolicy",
                "iam:GetRole",
                "iam:DeleteRolePolicy",
                "iam:DeleteRole",
                "iam:CreateRole"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": ""
        },
        {
            "Action":[
                "cloudwatch:StopMetricStreams",
                "cloudwatch:StartMetricStreams",
                "cloudwatch:PutMetricStream",
                "cloudwatch:ListMetricStreams",
                "cloudwatch:GetMetricStream",
                "cloudwatch:DeleteMetricStream"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": ""

        },
        {
            "Action":[
                "s3:PutObject",
                "s3:PutLifecycleConfiguration",
                "s3:PutBucketTagging",
                "s3:PutBucketPublicAccessBlock",
                "s3:PutBucketNotification",
                "s3:PutBucketAcl",
                "s3:ListBucket",
                "s3:ListAllMyBuckets",
                "s3:GetReplicationConfiguration",
                "s3:GetLifecycleConfiguration",
                "s3:GetEncryptionConfiguration",
                "s3:GetBucketWebsite",
                "s3:GetBucketVersioning",
                "s3:GetBucketTagging",
                "s3:GetBucketRequestPayment",
                "s3:GetBucketPublicAccessBlock",
                "s3:GetBucketPolicyStatus",
                "s3:GetBucketPolicy",
                "s3:GetBucketObjectLockConfiguration",
                "s3:GetBucketNotification",
                "s3:GetBucketLogging",
                "s3:GetBucketLocation",
                "s3:GetBucketCORS",
                "s3:GetBucketAcl",
                "s3:GetAccelerateConfiguration",
                "s3:DeleteBucket",
                "s3:CreateBucket"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": ""
        },
        {
            "Action":[
                "sqs:TagQueue",
                "sqs:SetQueueAttributes",
                "sqs:ListQueues",
                "sqs:ListQueueTags",
                "sqs:GetQueueUrl",
                "sqs:GetQueueAttributes",
                "sqs:DeleteQueue",
                "sqs:CreateQueue"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": ""
        }
    ]
}
```

4. Finish the **Creation Wizard**.
5. Allow Terraform to assume the newly created role.
    a. Browse to **IAM** > **Roles** > **Your newly created role** > copy the **ARN** value.
    b. Also, select **Trust Relationships**.
    c. Ensure that the user for Terraform has permissions to assume the role.

⚠ Replace **${ACCOUNT_ID}** and **${USERNAME}** with real values.

**Example**

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::${ACCOUNT_ID}:user/${USERNAME}",
                    "arn:aws:iam::${ACCOUNT_ID}:root"
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

For additional details on creating AWS IAM roles, please refer to https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_create.html.