AWS

AWS Overview

Multicloud Defense has created a CloudFormation template that you use when connecting an AWS account to the Multicloud Defense Controller.

To prepare cloud account for integration with Multicloud Defense Controller, there are certain steps that need to be performed in the cloud account. Below are the prerequisite steps you need to perform before connecting your AWS cloud account to Multicloud Defense Controller. This is intended to provide an overview of the operation and not intended to be performed manually. In CloudFormation section, there are details of deployments and parameters information.

Overview of steps

1. Create a cross account IAM role that's used by the Multicloud Defense Controller to manage your cloud account.
2. Create an IAM role that is assigned to the Multicloud Defense Gateway EC2 instances that run in your account.
3. Create a CloudWatch event rule that transfers the management events to the Multicloud Defense Controller.
4. Create an IAM role that is used by the above CloudWatch event rule that gives it the permissions to do the transfer of the management events.
5. Optionally create a S3 bucket in your account to store CloudTrail Events, Route53 DNS query logs and VPC Flow Logs.
6. Enable Route53 DNS Query Logging with the destination as the S3 Bucket created above and select the VPCs for which query logging must be enabled.
7. Enable CloudTrail to log all the management events to the S3 Bucket created above.
8. Enable VPC Flow Logs with destination as the S3 Bucket created above.

Connect AWS Account to Multicloud Defense Controller from The Multicloud Defense Dashboard

Multicloud Defense has created a CloudFormation template that makes it easy to connect an AWS account to the Multicloud Defense Controller.

Before you begin

Read through the following requirements before you connect an AWS account to Multicloud Defense:

- You must have requested a Multicloud Defense Controller for your CDO tenant before you begin.
- The name of the cloud storage bucket in your AWS account must be between 3-65 characters. Bucket names longer than 65 characters will result in an error during the connection process.

Note Multicloud Defense Controller version 23.10 defaults to IMDSv2 in the AWS EC2 instance when using Multicloud Defense Gateway version 23.04 or newer. For more information about the difference beween IMDSv1 and IMDSv2, see AWS documentation.

Step 1 In the CDO menu bar, click Multicloud Defense.

Step 2 Click Multicloud Defense Controller.

Step 3 In the Cloud Accounts pane, click **Add Account**.

Step 4 On the **General Information** page, select AWS from the **Account Type** list box.

Step 5 Click **Launch Stack** to download and deploy our CloudFormation template. This should open up another tab to deploy the template. Login to AWS is required.

Step 6 Acknowledge that the AWS CloudFormation might create IAM resources with custom names.

Step 7 Fill in these values:

- **AWS Account Number**: Enter the AWS account number of the account you wish to secure. This number can be found in the output value CurrentAccount of the CloudFormation Template.
- **Account Name**: Enter the name you want to give your account once it has been onboarded.
- **Description:**(Optional) Enter an account description.
- **External ID**: A random string for IAM role's trust policy. This value will be used in the controller IAM role created. You can edit or regenerate the External ID.
- **Controller IAM Role**: This is the IAM role created for theMulticloud Defense Controller during CloudFormation Template (CFT) deployment. Look for the output value MCDControllerRoleArm in CFT stack. It should be something similar to this: `arn:aws:iam::<Acc Number>:role/ciscomcdcontrollerrole`.
- **Inventory Monitor Role**: This is the IAM role created for Multicould Defense Inventory during CFT deployment. Look for the output value MCDInventoryRoleArn in CFT stack. Should be something similar to this: `arn:aws:iam::<Acc Number>:role/ciscomcdinventoryrole`.

Step 8 Click **Save and Continue**.
You are returned to the Multicloud Defense dashboard where you will see that the you have a new AWS cloud account recorded.

What to do next

Enable traffic visibility.

CloudFormation Outputs

From the **Outputs** tab, copy and paste the following informationin to a text editor:

- CurrentAccount (This is your AWS Account ID where applications run and Multicloud Defense Gateways will be deployed)
  - MCDControllerRoleArm
  - MCDGatewayRoleName
  - MCDInventoryRoleArn
  - MCDS3BucketArn
  - MCDBucketName

Roles Created by Multicloud Defense

When you onboard a cloud service account to Multicloud Defense Controller with the provided script, user roles are created within the parameters of the cloud service provider to ensure that communication between the services are protected. Depending on the cloud service provider, different roles and permissions are created.

The following roles are created when you onboard an account.

MCDControllerRole

Cross-account IAM role that allows Multicloud Defense to access your cloud account and take necessary actions, for example, Create EC2 instances, create load balancers, and change Route53 entries. The service principal is the Multicloud Defense-controller-account with an external id applied. Here is the IAM policy applied to the role (e.g controller role name used in this example is ***Multicloud Defense-controller-role***):

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "acm:DescribeCertificate",
        "aacm:ListCertificates",
        "apigateway:Get",
        "ec2:*",
        "elasticloadbalancing:*",
        "events:*",
        "globalaccelerator:*"
        "iam:ListPolicies",
```

```
        "iam:ListRoles",
        "iam:ListRoleTags",
        "logs:*",
        "route53resolver:*",
        "servicequotas:GetServiceQuota",
        "3:ListAllMyBuckets",
        "s3:ListBucket",
        "wafv2:Get*",
        "wafv2:List*",
      ],
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Action": [
        "iam:GetRole",
        "iam:ListRolePolicies",
        "iam:GetRolePolicy"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:iam::<ciscomcd-account>:role/ciscomcd-controller-role"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::<S3Bucket>/*"
    },
    {
      "Action": [
        "iam:GetRole",
        "iam:ListRolePolicies",
        "iam:GetRolePolicy",
        "iam:PassRole"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:iam::<customer- account>:role/ciscomcd_firewall_role"
    },
    {
      "Action": "iam:CreateServiceLinkedRole",
      "Effect": "Allow",
      "Resource": "arn:aws:iam::*:role/aws-service-role/*"
    }
  ]
}
```

Service Principal:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::<ciscomcd-account>:root"
        ]
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "sts:ExternalId": "ciscomcd-external-id"
        }
```

```
      }
    }
  ]
}
```

MCDGatewayRole

Role that is assigned to the Multicloud Defense Gateway (Firewall) EC2 instances. The role gives the Gateway instance capabilities to access secretsmanager where the private keys for the application are stored, ability to decrypt keys using AWS KMS if the keys are stored in KMS, and save objects like PCAPs and technical support data onto a S3 bucket. The service principal of this role is ec2.amazonaws.com. Here is the IAM policy applied to the role:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:PutObject",
        "s3:ListBucket"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::*/*"
    },
    {
      "Action": [
        "kms:Decrypt"
      ],
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Action": [
        "secretsmanager:GetSecretValue"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Tip You can download and edit the CloudFormation template to make the policy more restrictive e.g. restricting decrypt to use a specific key, or PutObject to a defined/specific S3 bucket.

MCDInventoryRole

This is the role used for dynamic inventory purposes and provides the capability for the CloudTrail events to be transferred to the Controller's AWS account. It does the following:

- Put events on the event bus in the AWS account where the Multicloud Defense Controller exists.
- Send events matching the rule to the Multicloud Defense Controller's webhook server directly from the customer's AWS account.

The Service Principal for this role is **events.amazonaws.com**. Here is the policy applied to the role:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "events:PutEvents",
      "Effect": "Allow",
      "Resource": [
        "arn:aws:events:*:<ciscomcd-account>:event-bus/default"
      ]
    }
  ]
}
```

InventoryMonitorRule

Rule that is added to the MCDInventoryRole to put all CloudTrail inventory changes to EC2 and API gateways to be copied to the event bus on the AWS account where the Multicloud Defense Controller runs. The rule is required to match on specific event patterns that occur in the customer's AWS account. Once a match occurs, the rule states that the matched event should be sent to the webhook server (API based destination) of the controller. This rule is executed using the Multicloud DefenseMCDInventoryRole created in the previous section.

Custom Event Pattern:

```
{
  "detail-type": [
    "AWS API Call via CloudTrail",
    "EC2 Instance State-change Notification"
  ],
  "source": [
    "aws.ec2",
    "aws.elasticloadbalancing",
    "aws.apigateway"
  ]
}
```

Target:

```
Event Bus in another AWS Account (mcd-account) using the MCDInventoryRole
```