Troubleshoot Connecting Your Account

Manually Onboard an Account

In cases where onboarding a cloud service provider account to Multicloud Defense with the methods provided in Account Onboarding, you may need to onboard your account manually. Use the following options as an alternative.

Manually Onboard a GCP Project

GCP Overview

GCP Project and GCP Folders

Multicloud Defense currently supports both GCP projects and GCP folders although these components are supported separately. Note the following limitations and exceptions for both of these options.

A GCP project has to potential to contain GCP resources like virtual machines, storage buckets, databases, and more. It can be used to create, enable, and use all Google Cloud services.

- Projects can be onbarded with terraform, manual onboarding, and scripted onboarding.
- Projects are ideal for environments that require orchestration, including discovery and investigation.
- You can interact with each project indvidually through the Multicloud Defense dashboard.

As of Version 23.10 you can connect a GCP folder with terraform. A GCP folder contains projects, other folders, or a combination of both. Organization resources can use folders to group projects under the organization resource node in a hierarchy.

- Folders that do not have the `roles/compute.admin` permission enabled are considered empty and are not used.
- Projects associated with onboarded folders are used for asset and traffic discovery only.

- Projects associated with onboarded folders do not accommodate orchestrating service VPC or gateway creation.
- Permissions made to folders from the GCP console must be made at the folder level. As such, Multicloud Defense actions are also made at the folder level.

If you want to onboard a GCP folder, see Terraform Repository.

Overview Procedure

The following is an overview of how to connect your GCP project. An shell **script** is provided by Multicloud Defense and facilitates an easy connective process as part of a wizard. The script automates the following steps so you don't have to:

1. Create two service accounts.
2. Enable the following APIs (Compute Engine, Secret Manager).
3. Create the two following VPCs (management, datapath).
4. Create firewall rules to allow traffic to the Multicloud Defense Gateway (app traffic) in the datapath VPC.
5. Create firewall rules to allow management traffic from Multicloud Defense Gateway to the Multicloud Defense Controller in the management VPC.

If you find that the script does not work, or if you need to manually change your settings, these actions can be executed using the GCP cloud console web UI, or using the gcloud CLI. See the alternative method of connecting your project Manually Onboard a GCP Project.

Service Accounts

Multicloud Defense requires two service accounts created in your GCP project:

- **multicloud defense-controller**: This account is used by theMulticloud Defense Controller to access your GCP project to create resources (Multicloud Defense Gateways), load balancers for Multicloud Defense Gateways, and read information about the VPCs, Subnets, Security Group tags etc.
- **multicloud defense-gateway**: This account is assigned to the Multicloud Defense Gateways (Compute VM instances). The account provides access to the secret manager (private keys for TLS decryption) and storage.

You can create these service accounts in one of two ways: by using the service available in the UI or by using the the cloud service provider's CLI.

Create Multicloud Defense Controller Service Account Using GCP Cloud Console

The Multicloud Defense Controller service account is used by the Multicloud Defense Controller to access and manage resources in your GCP project. You must create the account and generate a key. The key is added to the Controller as part of Account onboarding to the Controller.

Procedure

Step 1 Open **IAM** in your GCP project.

Step 2 Click **Service Accounts**.

Step 3 Create **Service Account**.

Step 4 Provide a name and ID (e.g multicloud defense-controller) and click **Create**.

Step 5 Add **Compute Admin** and **Service Account User roles**.

Step 6 Click **Continue**.

Step 7 Click **Done**.

Note There is no requirement to add any users.

Step 8 Click on the newly created account, scroll down to **Keys** and in the dropdown for **Add Key** and select **Create New Key**.

Step 9 Choose JSON (default option) and click **Create**.

Step 10 A file is downloaded to your computer. Save this file.

Create a Multicloud Defense Firewall Service Account Using the GCP Cloud Console

The multicloud defense firewall service account is used by the Multicloud Defense Gateway instances running inside your GCP project. The Gateways may need to access the private keys stored in the SecretManager for TLS decryption and access storage to store PCAP files etc. (if configured by the user). Also, the Gateways many need Log Writer permissions to send logs from Multicloud Defense Gateway to the GCP logging instance (if configured by the user).

Below are two (2) methods of creating this service account.

Procedure

Step 1 Open **IAM** in your GCP project.

Step 2 Click **Service Accounts**.

Step 3 Create **Service Account**.

Step 4 Provide a name and ID (e.g multicloud defense-firewall) and click **Create**.

Step 5 Add **Secret Manager**, **Secret Accessor** and **Logs Writer roles**.

Step 6 Click **Continue**.

Step 7 Click **Done**.

Note There is no requirement to add any users.

Enable API

You can enable the API for communication between Multicloud Defense Controller and your GCP account with either GCP console or the cloud service provider's CLI.

Enable API-Using the GCP Cloud Console

Enable the APIs in your project/account so that the Multicloud Defense Controller can create Multicloud Defense Gateways (Virtual Machines, Load Balancers).

Procedure

Step 1 Search for **Compute Engine API** in the searchbar.
Step 2 Click **Enable**.
Step 3 Search for **Secret Manager API** in the searchbar.
Step 4 Click **Enable**.
Step 5 Search for **Identity and Access Management (IAM) API** in the searchbar.
Step 6 Click **Enable**.
Step 7 Search for **Cloud Resource Manager API** in the searchbar.
Step 8 Click **Enable**.

VPC Setup

Multicloud Defense Gateway instances can be deployed in edge or hub mode. In edge mode, the gateway instances run in the same VPC as your applications. This document focuses on preparing you to deploy the Multicloud Defense Gateway deployment in edge mode.

VPC and Subnets

When deploying the Multicloud Defense Gateway, the Multicloud Defense Controller will prompt for the **management** and **datapath** VPC information. Multicloud Defense Gateway instances require two network interfaces. In GCP, the network interfaces of a VM instance need to be in different VPCs unlike other cloud providers where they can be in just different subnets. If you already have a VPC where the application is running, you have the **datapath** VPC and the subnet. You must create another VPC (or use an another existing VPC) for management purposes. You can either use the auto-created subnets or create them manually.

The datapath vpc is the VPC where your applications are running and will be referred to as such in the following sections

In each of the VPCs, Multicloud Defense requires one subnet for datapath and one subnet for management.

The **management** subnet is a public subnet that must be associated with the route table that has a default route to the Internet. The Multicloud Defense Gateway instance has an interface attached to this subnet that it uses to communicate with the Multicloud Defense Controller. This interface is used for policy pushes and other management and telemetry activities between the Multicloud Defense Controller and the Multicloud Defense Gateway instances. Customer application traffic **does not** flow through this interface and subnet. The interface is associated with the **multicloud defense- management** network tag (or any tag based on your team requirements), which is described in the network tags section below.

The **datapath** subnet is a public subnet that must be associated with the route table that has a default route to the Internet. The Multicloud Defense Controller creates a network load balancer (NLB) in this subnet. In addition, a Multicloud Defense Gateway instance has an interface attached to this subnet. The customer applications traffic **flows** through this interface. A security policy is applied to the traffic ingressing through

this interface. The interface is associated with the **multicloud defense-datapath** network tag (or any tag based on your team requirements), which is described in the network tags section below.

Sample VPC and Subnets using CLI

Use the following commands as an example when executing your own commands to create VPCs for your GCP account. Open the Google Cloud Shell windows for these particular commands:

Procedure

Step 1 Create VPC **apps** and subnet **apps-us-east1**

Step 2 Create VPC `multicloud defense-mgmt` and subnet `multicloud defense-mgmt-us-east1`:

Step 3 Create at least two Firewall rules for VPC `multicloud defense-mgmt` with target-tags as `multicloud defense-mgmt`:

a. Egress rule to allow all the outbound traffic:

b. Ingress rule to allow SSH into the firewall instances:

Step 4 Create at least three Firewall rules for VPC **apps**. Use the following as examples:

a. One egress rule to allow all the outbound traffic with target-tags as `multicloud defense-datapath`:

b. One ingress rule to allow HTTP and HTTPS into the gateway instances through the non-load balancerwith target-tags as `multicloud defense-datapath`:

c. Once egress rule to allow all the outbound traffic with target-tags as `app-instance`:

d. One ingress rule to allow `tcp:8000` with `target-tags` as `app-instance`:

```
gcloud config set project <project-name> # incase the project is not set in the gcloud cli shell
gcloud compute networks create apps --subnet-mode custom
gcloud compute networks subnets create apps-us-east1 --network apps --range 10.0.0.0/24 --region
us-east1
gcloud compute networks create ciscomcd-mgmt --subnet-mode custom
gcloud compute networks subnets create ciscomcd-mgmt-us-east1 --network ciscomcd-mgmt --range
172.16.0.0/24 --region us-east1
gcloud compute firewall-rules create ciscomcd-mgmt-out --direction EGRESS --network ciscomcd-mgmt \
    --target-tags ciscomcd-mgmt --allow tcp,udp
gcloud compute firewall-rules create ciscomcd-mgmt-in --direction INGRESS --network ciscomcd-mgmt \
    --target-tags ciscomcd-mgmt --allow tcp:22
gcloud compute firewall-rules create ciscomcd-datapath-out --direction EGRESS --network apps \
    --target-tags ciscomcd-datapath --allow tcp,udp
gcloud compute firewall-rules create ciscomcd-datapath-in --direction INGRESS --network apps \
    --target-tags ciscomcd-datapath --allow tcp:80,tcp:443
gcloud compute firewall-rules create app-instance-out --direction EGRESS --network apps \
    --target-tags app-instance --allow tcp,udp
gcloud compute firewall-rules create app-instance-in --direction INGRESS --network apps \
    --target-tags app-instance --allow tcp:8000,tcp:22
```

Once you run the above commands, you can create a VM instance in the **apps** VPC and launch a test web application on port 8000.

```
gcloud compute instances create app-instance1 \
  --zone=us-east1-b \
  --image-project=ubuntu-os-cloud \
```

```
  --image-family=ubuntu-2004-lts \
  --network apps \
  --subnet=apps-us-east1 \
  --tags=app-instance
gcloud compute ssh app-instance1 --zone us-east1-b
echo hello world > index.html
python3 -m http.server 8000
```

Network Tags (for GCP Gateways)

The management and datapath network tags are associated with the respective interfaces on the Multicloud Defense Gateway instance, as described in the subnets section above.

Create a gateway rule in the **management** VPC and associate that with **multicloud defense-management** network tag. This must allow all outbound traffic that makes the gateway instance communicate with the controller. Optionally, for inbound rules, enable port 22 (SSH) to allow SSH access to the gateway instance. SSH is **not required** for the Multicloud Defense firewall to function properly.

Create a gateway rule in the **datapath** VPC and associate that with **multicloud defense-datapath** network tag. This must allow the traffic to the Multicloud Defense Gateway for all the services that you enable (are going to enable).

For example, if an application is running on port 3000 and is proxied by the Multicloud Defense Gateway on port 443, port 443 must be opened on the multicloud defense-datapath network security tag.

Gateway Creation

Using the Multicloud Defense Gateway creation page use the following parameters:

1. Datapath VPC: **apps**.
2. Datapath Network Tag: **multicloud defense-datapath**.
3. Management VPC: **multicloud defense-mgmt**.
4. Management Network Tag: **multicloud defense-mgmt**.
5. Use **us-east1-b** zone.
6. Management Subnet: **multicloud defense-mgmt-us-east1**.
7. Datapath Subnet: **apps-us-east1**.

You can create subnets in other regions to test the Multicloud Defense Gateway in multi-availability zone mode.

Manually Onboard an Azure Subscription

If you cannot directly connect an Azure subscription with the script provided in the Multicloud Defense Controller dashboard, use the procedures below to manually connect your subscription

(Optional) User-assigned Managed Identity for Key Vault and Blob Storage access

Multicloud Defense Gateways can optionally integrate with Azure Key Vault to retrieve TLS certificates and with Blob Storage for saving PCAP (packet capture) files. User-assigned managed identities are used to grant access to these services.

In the Azure portal, navigate to **Managed Identities** to create an identity.

Alternatively in Azure Cloud Shell, run the following command:

```
az identity create -g <RESOURCE GROUP> -n <USER ASSIGNED IDENTITY NAME>
```

For information on creating TLS certificate secrets in Azure Key Vault, see Azure Key Vault.

Register Application in Microsoft Entra ID

Use the following procedure to register the Multicloud Defense application in your Entra ID.

Procedure

Step 1 From your Azure portal, navigate to **Microsoft Entra ID**.

Step 2 Select **App registrations**.

Step 3 Click **New registration**.

Step 4 Provide a name to reference the new app registration e.g. Multicloud Defense Controller In the *Supported account types* choose the second option *Accounts in any organizational directory*.

Step 5 Choose the option appropriate to your organization. Note that the **Redirect URI** is not needed for the creation of the App registration.

Step 6 Click **Register**.

Step 7 In the left navigation bar under the newly created application, click **Certificates & secrets**.

Step 8 Click **+ New client secret**, and then enter the required information in the *Add a client secret* dialog

- **Description** - Add a description (e.g multicloud defense-controller-secret1)
- **Expires** - Choose **Never**. You can also make this selection at your convenience. You will need to create new secrets when the current one expires)

Step 9 Click **Add**. The client secret is populated under the **Value** column.

Step 10 Copy the **Client secret** into a notepad, as this is shown only once and is never displayed again.

Step 11 In the left navigation bar click **Overview**.

Step 12 Copy the **Application (client) ID** and **Directory (tenant) ID** into a notepad.

Create a custom role to assign to the Application

The CloudFormation template creates the following role:

- **Custom Role** - The custom role gives the application permissions to read inventory information and create resources (e.g., VMs, load balancers, etc.) The custom role can be created in multiple ways.

Create a **custom role** that will be assigned to the application created for the Multicloud Defense Controller. The custom role gives the application permissions to read inventory information and create resources (e.g., VMs, load balancers, etc.) The custom role can be created in multiple ways.

Procedure

Step 1 Navigate to **Subscription** and click **Access Control (IAM)**.

Step 2 Click on **Roles** and on the top menu bar navigate to click **+Add** > **Add Custom Role**.

Step 3 Give a name to the custom role (e.g., `multicloud defense-controller-role`).

Step 4 Keep clicking **Next** until you get to the JSON editing screen.

Step 5 Click **Edit** on the screen and in the JSON text, under the **permissions > actions** section, copy and paste the following content between the square brackets (no need to maintain the indentation):

```
"Microsoft.ApiManagement/service/*",
"Microsoft.Compute/disks/*",
"Microsoft.Compute/images/read",
"Microsoft.Compute/sshPublicKeys/read",
"Microsoft.Compute/virtualMachines/*",
"Microsoft.ManagedIdentity/userAssignedIdentities/read",
"Microsoft.ManagedIdentity/userAssignedIdentities/assign/action",
"Microsoft.Network/loadBalancers/*",
"Microsoft.Network/natGateways/*",
"Microsoft.Network/networkinterfaces/*",
"Microsoft.Network/networkSecurityGroups/*",
"Microsoft.Network/publicIPAddresses/*",
"Microsoft.Network/routeTables/*",
"Microsoft.Network/virtualNetworks/*",
"Microsoft.Network/virtualNetworks/subnets/*",
"Microsoft.Resources/subscriptions/resourcegroups/*",
"Microsoft.Storage/storageAccounts/blobServices/*",
"Microsoft.Storage/storageAccounts/listkeys/action",
"Microsoft.Network/networkWatchers/*",
"Microsoft.Network/applicationSecurityGroups/*",
"Microsoft.Compute/diskEncryptionSets/read",
"Microsoft.Insights/Metrics/Read"
```

Step 6 **Optional** - If you plan to use multiple subscriptions with Multicloud Defense, you must edit the JSON at `assignableScopes` to add another subscription line or change it to ***** (star) so the custom role can be used with all subscriptions.

Step 7 Click **Save** at the top of the text box.

Step 8 Click **Review + Create** and create the role.

Step 9 Once the custom role is created return to **Access Control (IAM).**

Step 10 On the top menu bar, click **Add** > **Add role assignment**.

Step 11 In the **Role** dropdown, select the custom role created above.

Step 12 In the **Assign access to** dropdown leave it as the default (Azure AD user, group, service principal).

Step 13 In the **Select** text box, type in the name of the application created earlier (e.g. `multicloud defensecontrollerapp`) and click **Save**.

Step 14 In the **Subscription** page, click on the **Overview** in the left menu bar and copy the subscription ID to the notepad.

Required Values For Multicloud Defense Controller Onboarding

Make sure you have the following information before proceeding further:

- Subscription ID (*from subscription overview page*)
- Directory (Tenant) ID (*from the Azure AD app overview page*)
- Application (client) ID (*from the Azure AD app overview page*)
- Client Secret (*Copied when the Client secret was created*)

Accept Marketplace Terms

Multicloud Defense Controller creates Gateway instances using a Multicloud Defense virtual machine (VM) image from the Azure marketplace. The Terms and Conditions must be accepted for each subscription. Open the Azure cloud shell from the Azure portal website (on the top menubar towards the right side). Choose or switch to bash shell and execute the following command (replace the subscription-id with your subscription id copied in the previous section):

```
az vm image terms accept --publisher valtix --offer datapath --plan valtix_dp_image
--subscription subscription-id
```

Graceful Termination of Connections

Multicloud Defense Gateway can choose to terminate an established flow for multiple reasons such as:

- Termination based on the policy. For example, FQDN filtering can only be applied after the flow is established.
- IDS/IPS can deem any packet in the flow that is sent by either the client or the server to be unsafe and can choose to terminate an established flow.
- Proxy service on the Multicloud Defense Gateway decides to terminate the flow after the flow is established.
- When one of the timers in the Multicloud Defense Gateway TCP stack decides that the flow is no longer active or alive.
- Flow termination during certain configuration changes such as PRS updates, gateway setting changes and so on.
- Flow termination when the gateway is decommissioned (controller initiated - disable/upgrade/scale-in).

Currently, when a Multicloud Defense Gateway chooses to terminate an established flow for any of the above reasons, it does so without informing the client and the server about the termination (except if there is FQDN Filtering with Reset on Deny turned on). This causes the client and server to rely on TCP or application timeouts to detect the loss of connection, causing application outages.

For TCP flows, Multicloud Defense Gateway introduces a graceful termination mechanism which causes the gateway to send a TCP Reset to the client (initiator) when the gateway stops the flow. This should enable the client TCP stack to terminate the connection quickly, enabling the applications to attempt to re-establish the interrupted flow, thereby minimizing traffic disruption. This applies to all kinds of flows - forwarded, forward proxied, and reverse proxied, that are handled by the Multicloud Defense Gateway.

Also, when a Multicloud Defense Gateway data plane goes down unexpectedly (due to a software issue), this reset mechanism does not apply. Clients will continue to rely on application timeouts to recover.

Troubleshooting

To find flows that are terminated with a TCP Reset by the Multicloud Defense Gateway, download the traffic summary (from the controller) as a CSV and search for *RESET*. It will be the last connection state for the ingress flow. Connections that are terminated naturally will not have this state as the last state. For non-TCP flows, the last connection state is always *AGED OUT*.

Terraform Onboarding Scripts for Cloud Accounts

You an use the terraform script to onboard your cloud service provider account instead of using the onboarding wizard or the manual process.

About Terraform

Multicloud Defense customers can use the **Terraform Provider** to: **discover** - onboard public cloud accounts, gain continuous asset visibility and detect indicators of compromise (IoC); **deploy** - Multicloud Defense Gateways to protect ingress, egress and east-west traffic; and **defend** - with multi-cloud (AWS, Azure, GCP, OCI) dynamic policies with continuously discovered cloud assets.

Attention As of Multicloud Defense Controller version 23.10, you can connect a GCP folder as well as a GCP project using the terraform provider. See Terraform Repository, on page 10 for more information.

The Multicloud Defense terraform provider is a “Verified” provider available from the terraform registry. Customers can now use the terraform provider for Multicloud Defense to bake security into their operations, i.e. on-board their cloud accounts into Multicloud Defense, deploy Multicloud Defense Gateways and specify security policies to protect against ingress attacks from the Internet (WAF, IDS/IPS, Geo-IP), stop exfiltration on egress traffic (TLS decryption, IDS/IPS, AV, DLP, FQDN/URL filtering), and prevent east-west attacks between VPCs/VNets. The security policies can be specified based on cloud asset tags (e.g., “dev”, “test”, “prod”, “pci”, “web”, “app1” etc.)

For more information, refer to:

- Download the Terraform Provider for Multicloud Defense.
- Examples in GitHub.
- Multicloud Defense Blog on Terraform.

Terraform Repository

Use case	Description	Github Repository
AWS onboarding	This is for onboarding AWS account using Terraform.	AWS Github Repo

Use case	Description	Github Repository
AWS discovery CFT	This CFT deployment will include all necessary privileges needed to use Multicloud Defense's discovery feature. For full feature set, please use the native product CFT.	AWS Discovery Github Repo
AWS discovery	This is for onboarding AWS account for discovery only mode using Terraform.	AWS Github Repo
Azure onboarding	This is for onboarding Azure Subscription using Terraform.	Azure Github Repo
GCP Project onboarding	This is for onboarding GCP project using Terraform.	GCP Github Repo
GCP Folder onboarding	This is for onboarding GCP folder using Terraform.	GCP Github Repo

Exporting Configuration as Terraform Block

Customers can export security profiles into terraform resource blocks from Multicloud Defense Controller. To export configuration into Terraform block, navigate and select the intended security profile and click on **Export** button. This will download a file that has the terraform block for the selected object/security profile.

All objects and profiles support terraform export with the exception of:

- Gateways
- Service VPCs/VNets
- Diagnostics