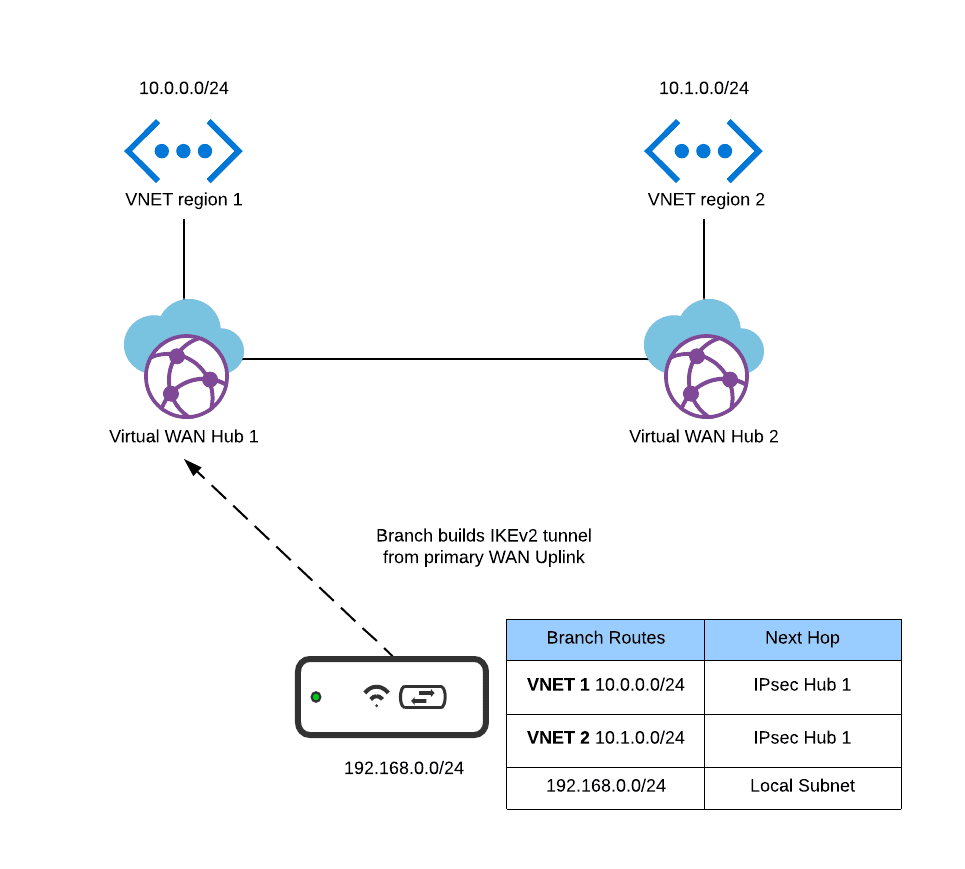
# Cisco Meraki Azure Virtual WAN Deployment Guide

## Executive Summary

This document encompasses a detailed step by step deployment process on automating connectivity from your Cisco Meraki branch to Azure Virtual WAN. Supported Use Cases along with a reference architecture will be provided.

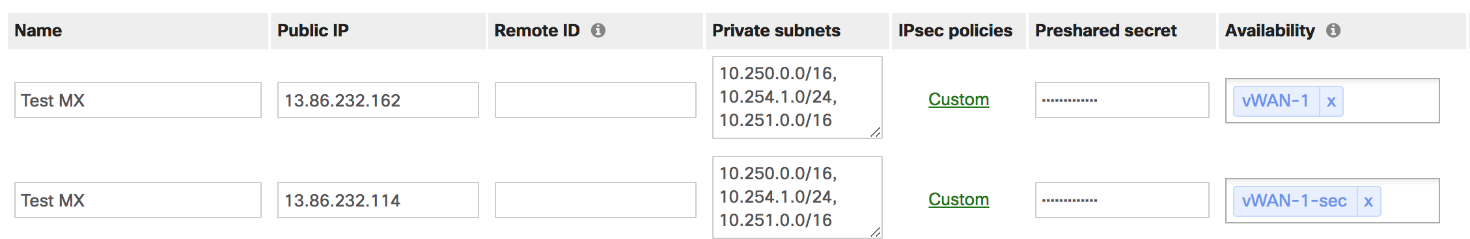
**Sample Topology**



In the above diagram, there is a branch that tunnels directly to the Azure Virtual WAN Hub. This tunnel is automatically built utilizing the toolkit provided. IKEv2 is utilized for securing connectivity to the VPN Gateway.

## Supported Use Cases

The design supports the use cases for in region and inter-region Azure connectivity for branch (spoke) sites. In this scenario, a branch securely builds an IPsec tunnel to the Azure Virtual WAN VPN Gateway. Specifically, the branch will build a **single** tunnel to the Instance 0 IP of the VPN Gateway. The toolkit will configure two tunnels with two different IPs (Each being Instance 0 and 1 for the VPN Gateway). However, only a single tunnel will be active at any given time. This is due to the fact that Cisco Meraki does not support Equal Cost Multipath and therefore cannot have 2 VPN tunnels to the same destination (private subnet). Below is a screenshot of the sample configuration that will be built once the branch has been tagged accordingly:



Failover scenarios will be discussed in the upcoming sections.

### ISP Failover

If a branch site experiences an ISP outage, the IPsec tunnel will be re-built utilizing the backup ISP automatically. Only one active tunnel can be configured per destination (In the example VNET 1 and 2). In other words, only a single tunnel is used on an uplink at any given time. When that uplink is detected as Down state, the appliance then builds the tunnel out its backup WAN interface.

The events for detecting an uplink failover are:

**DNS test**

* Query the DNS servers (primary or secondary) configured on the Internet interface for the following hosts:
* meraki.com
* google.com
* yahoo.com

**Internet test**

* Ping 8.8.8.8 every second.
* Uses a round-robin technique to send an HTTP GET to [http://google.com](http://google.com/), [http://yahoo.com](http://yahoo.com/), or [http://meraki.com](http://meraki.com/). An HTTP response of any kind will result in a success.

**ARP test**

* ARP for the default gateway and its own IP (to detect a conflict).

Details on these tests are outlined in the following documentation:

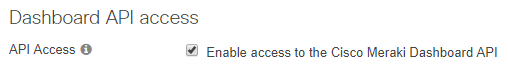
<https://documentation.meraki.com/MX/Firewall_and_Traffic_Shaping/Connection_Monitoring_for_WAN_Failover>

## Deployment Steps

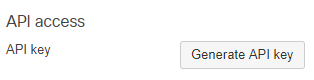
### Obtain Cisco Meraki API Key

The API Key and Org ID will be needed for the script to configure your Meraki device. For access to the **API**, first enable the **API** for your organization.

1. Navigate to **Organization** > **Settings** > **Dashboard** **API** **access**
2. Enable the **API**



1. Navigate to the **my profile page** and generate an **API key**
   1. Note: The **API key** is associated with a Dashboard administrator account.

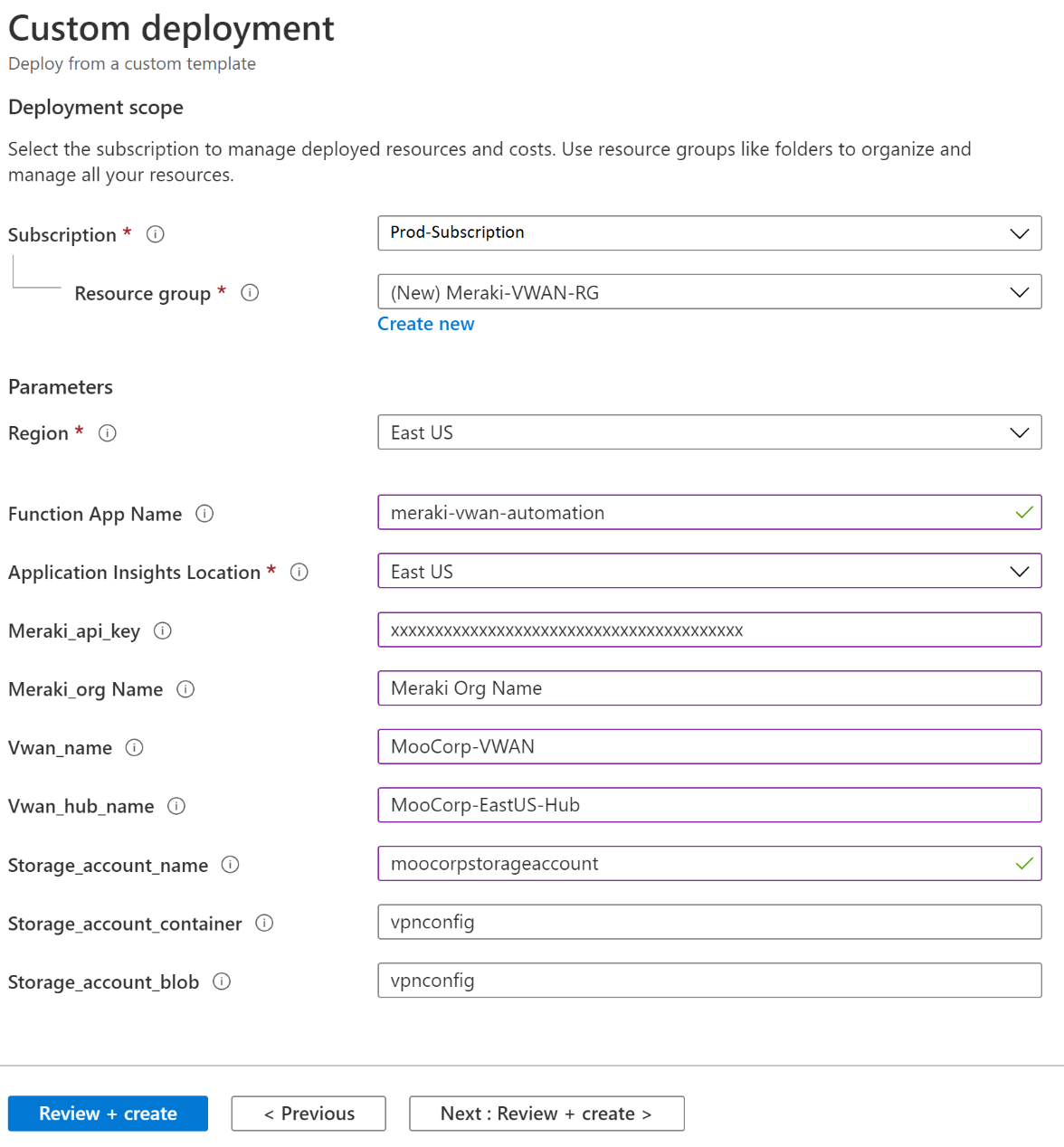
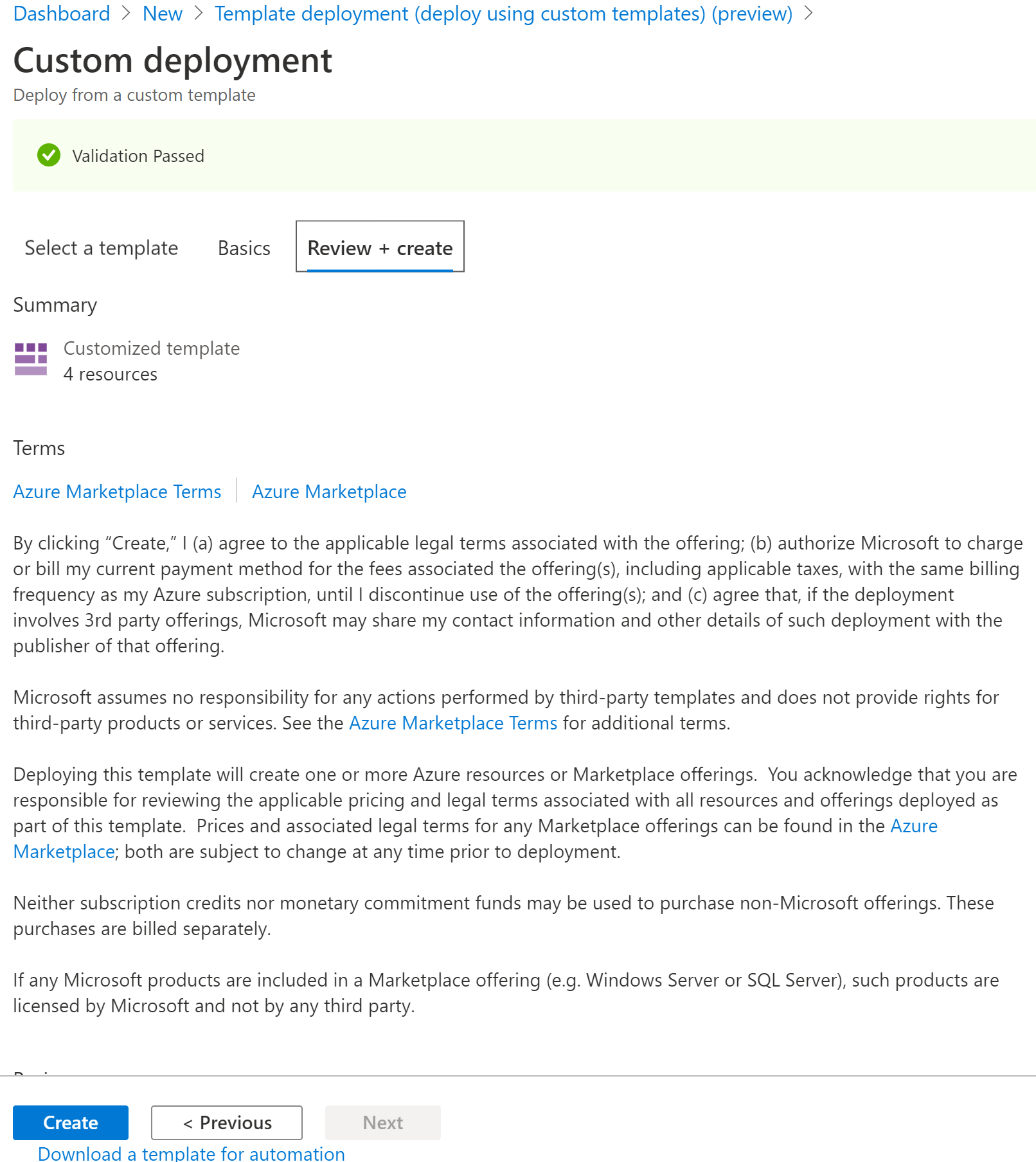


**Note:** In addition to the Dashboard API Key and the Dashboard Organization name, IKEv2 will have to be enabled for all the sites that will connect to Azure Virtual WAN. **Today, there is no way for customers to enable this capability in the dashboard and requires your Cisco Meraki sales engineer to enable the capability.**

**Note:** The toolkit also contains a firmware validation checker. One of the requirements for the solution is that the branch MX must be on **firmware 15 or greater**.

### Run the deployment script to create the Azure Function app:

For automation, Azure Functions are used to run a Python script every 5 minutes to sync the configuration between the Cisco Meraki and Virtual WAN configurations. An Azure Resource Manager (ARM) Template is used to help facilitate the deployment process and creation of the Azure Function, Storage Account, and Application Insights resources.

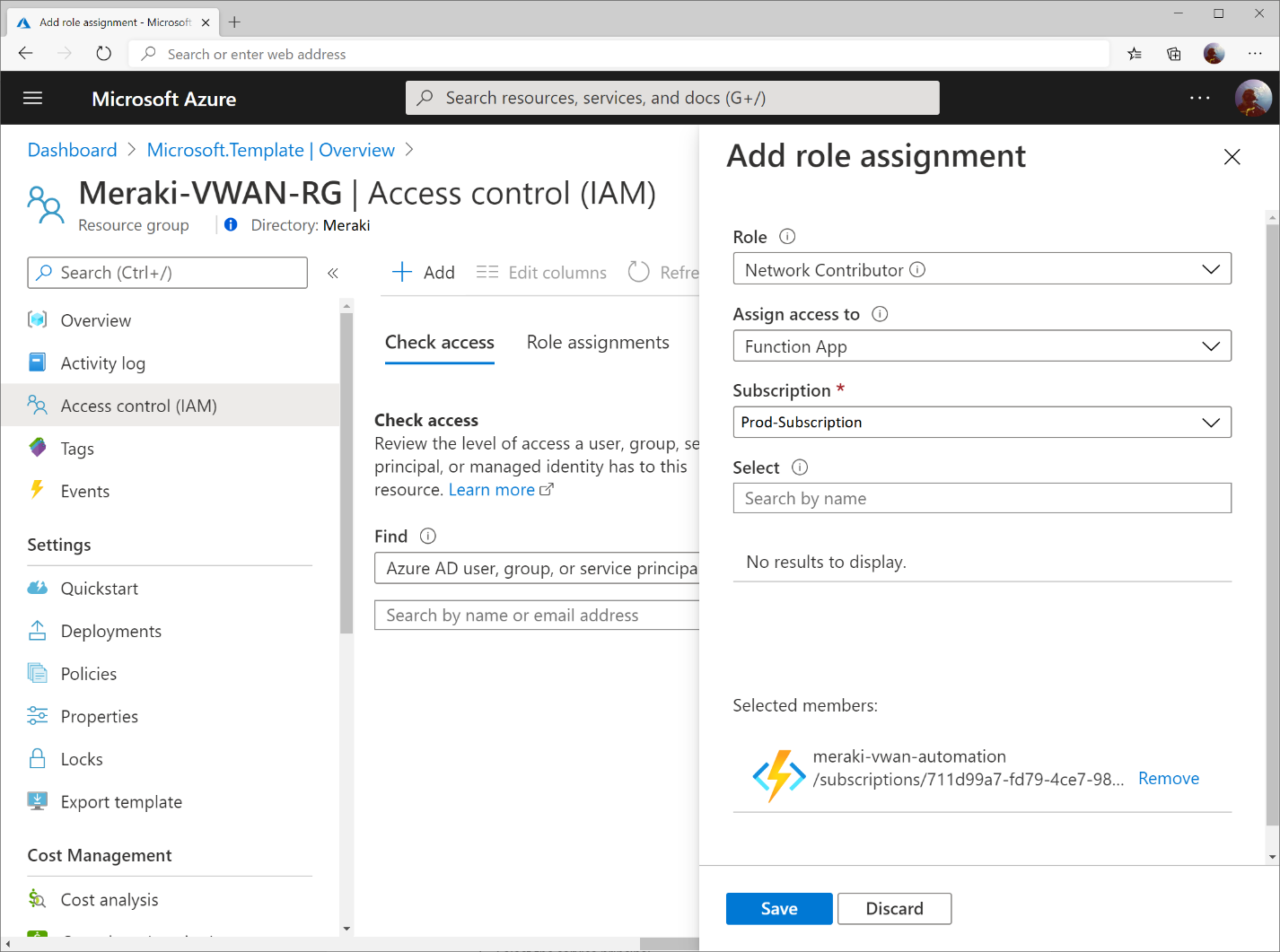
1. To deploy the Azure Function, navigate to Meraki’s GitHub repository for Azure Virtual WAN.
   1. <https://github.com/meraki/azure-cloud-scanning-api>
2. Next, click the **Deploy to Azure** button.
3. When logged in to the Azure Portal, fill out the required parameters and click **Review + Create**.
   1. **Note:** The Function App Name should not contain any special characters other than “-“. The function app also needs to be unique globally within Azure, so please use a name that is unique to your organization.
   2. **Tip:** More information on each of the required parameters can be found by hovering over the  icon in the Azure Portal.
   3. **Example Configuration**
      1. 
4. Click **Create** on the Summary blade.
   1. 

### Delegate access to the function app in the Azure Portal:

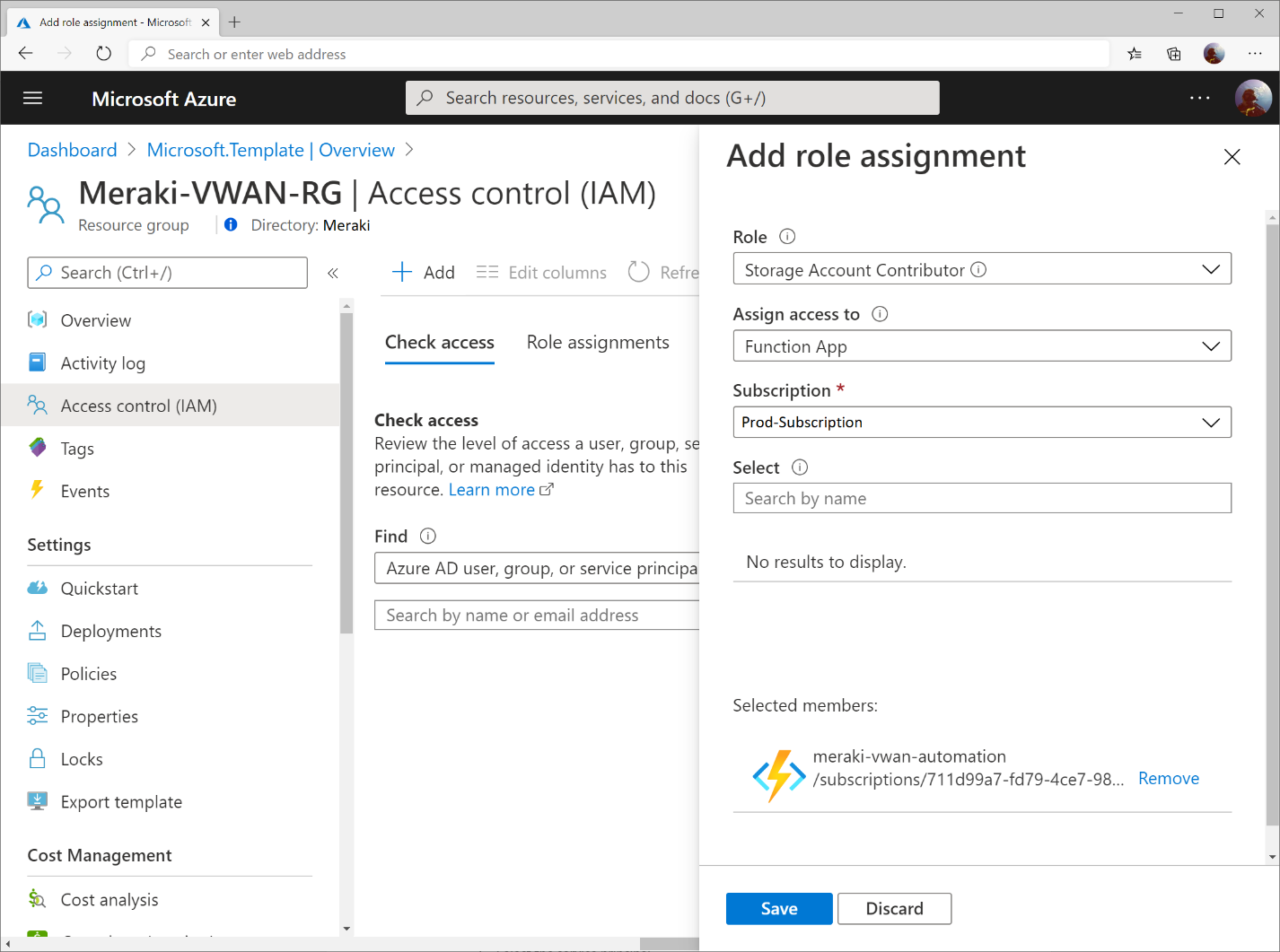
Once the template has finished deploying, the Azure Function will not have any access to your Virtual WAN resources once provisioned. To delegate access to Virtual WAN and a Storage Account used to store the Virtual WAN configuration files, please complete the following steps.

**Note:** When you delegate permissions, you can select the Contributor role for delegation at the resource group level, however if you are looking for stricter permissions, you can use the built-it Network Contributor and Storage Account Contributor roles.

1. Sign in to the [Azure portal](https://portal.azure.com/) with your Azure account.
2. Navigate to your Resource Group that contains your Virtual WAN and Storage Account resources.
3. Select **Access control (IAM)**, **+ Add**, and select **Add Role Assignment**.
4. Click the **Assign access** to dropdown and select **Function App**
5. Specify the following:
   1. Role: **Network Contributor**
   2. Select: **Cisco-Meraki-VWAN**
   3. Select the service principal
   4. Click **Save**

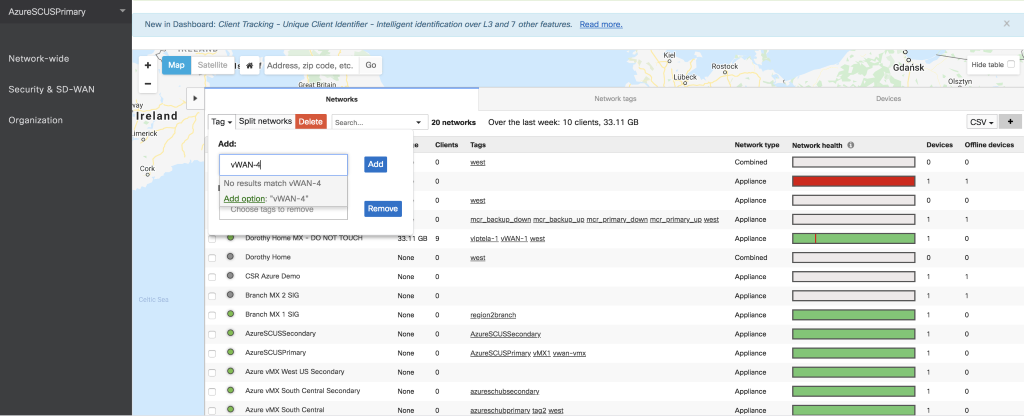


1. On the **Access control (IAM)** blade, click **+ Add**, and select **Add Role Assignment**.
2. Click the **Assign access** to dropdown and select **Function App**
3. Specify the following:
   1. Role: **Storage Account Contributor**
   2. Select: **Cisco-Meraki-VWAN**
   3. Select the service principal
   4. Click **Save**



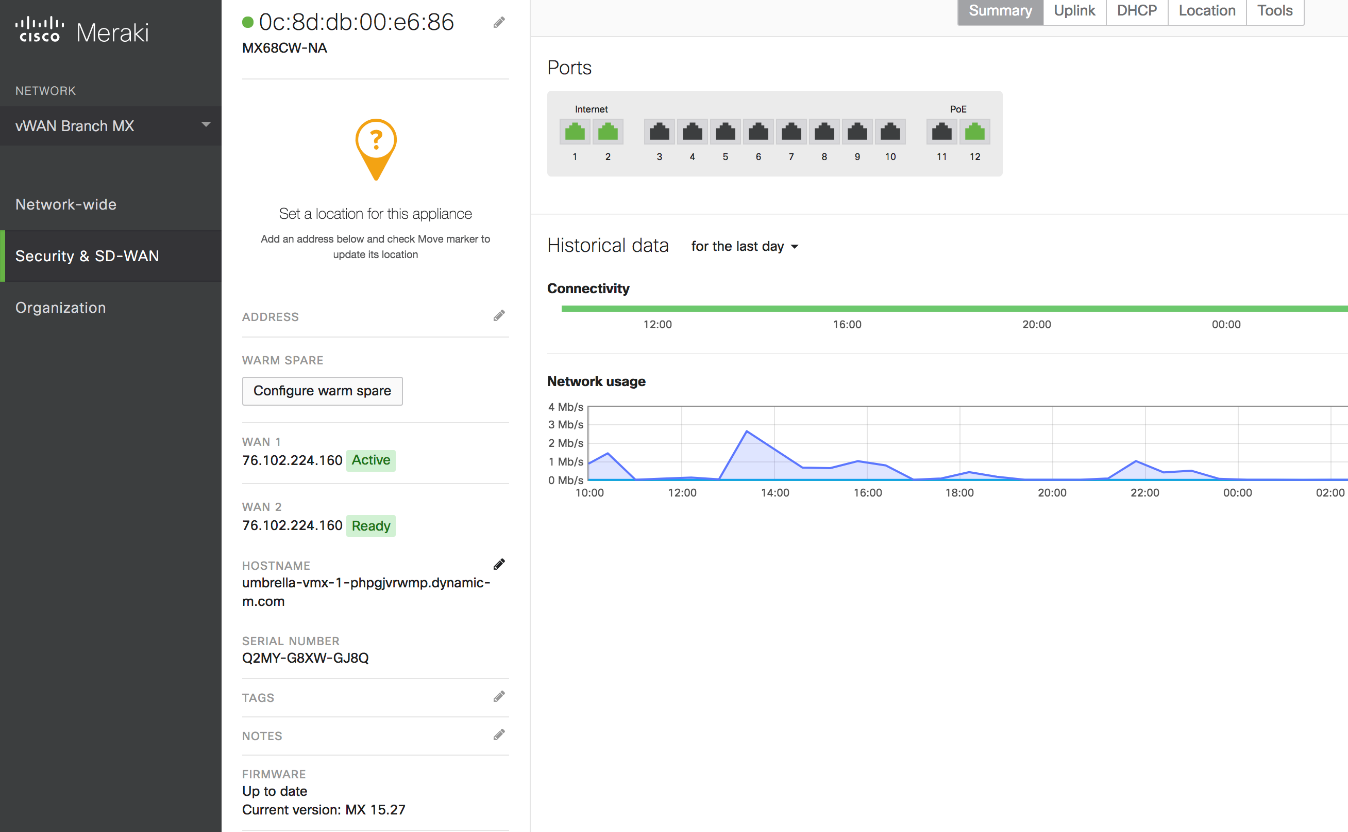
### Meraki Workflow

Initially there will be no tagged Meraki networks so the script will sleep and perform the same GET to obtain any networks with the relevant tag. In order to deploy a new branch the user would navigate to Organization > Overview and select the checkbox next to the network that you wish to connect. Below is a snippet of the Meraki Dashboard Overview page:



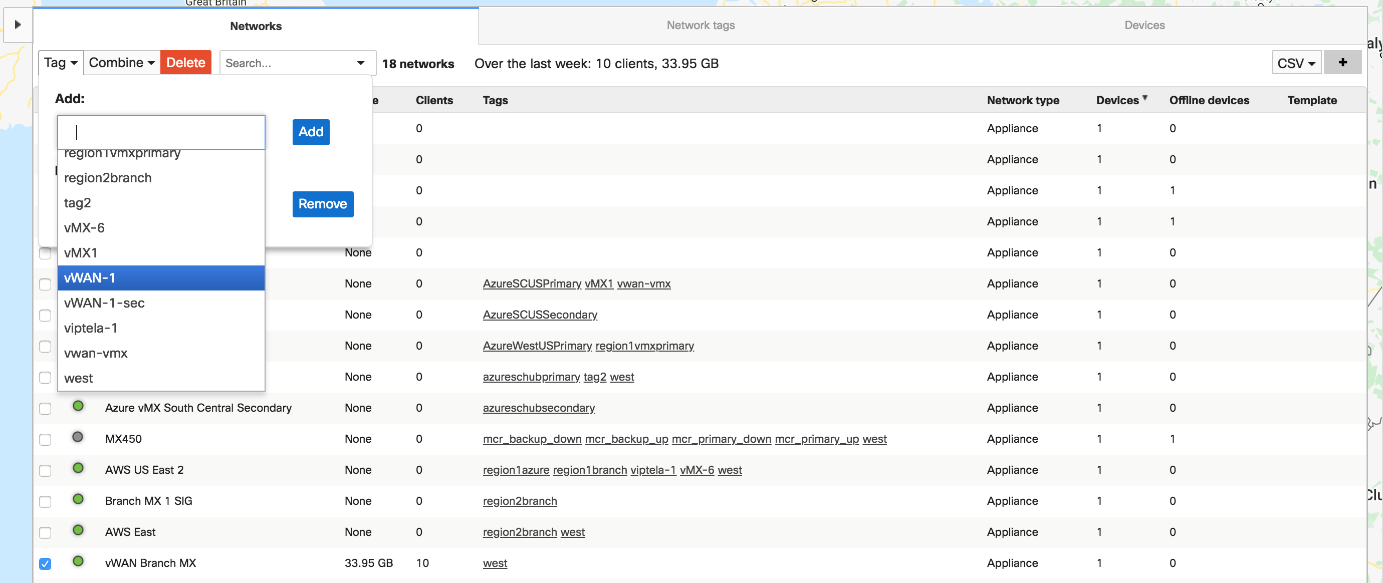
Once the network is tagged appropriately, connectivity is then automatically established. A customer gateway in the Azure Virtual WAN dashboard is created with a matching network name to that of the Meraki branch. Site1 will be named Site1 in both the Azure Virtual WAN and Meraki dashboards. Additionally, a site to site VPN will appear on the site to site VPN page.

In this example we have a site named branch 1 that has been provisioned and already connected to dashboard. Additionally, on the Appliance Status page the firmware version can be verified as seen below:

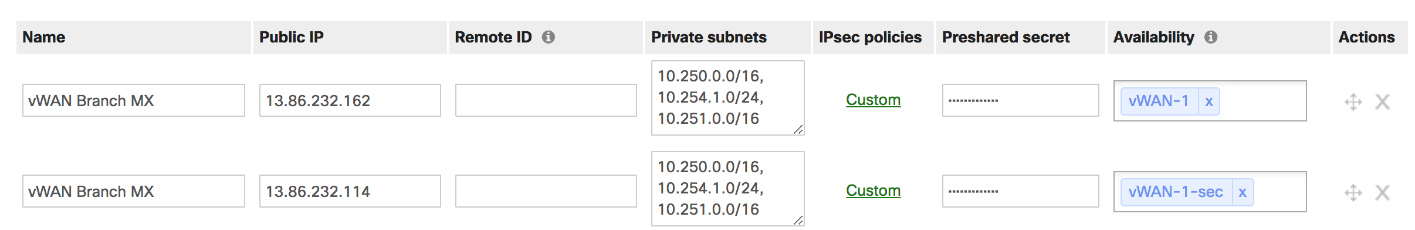


Once the device has been provisioned and connected to dashboard with the correct firmware, we are ready to connect. As described above, the next step is to appropriately tag the dashboard network.

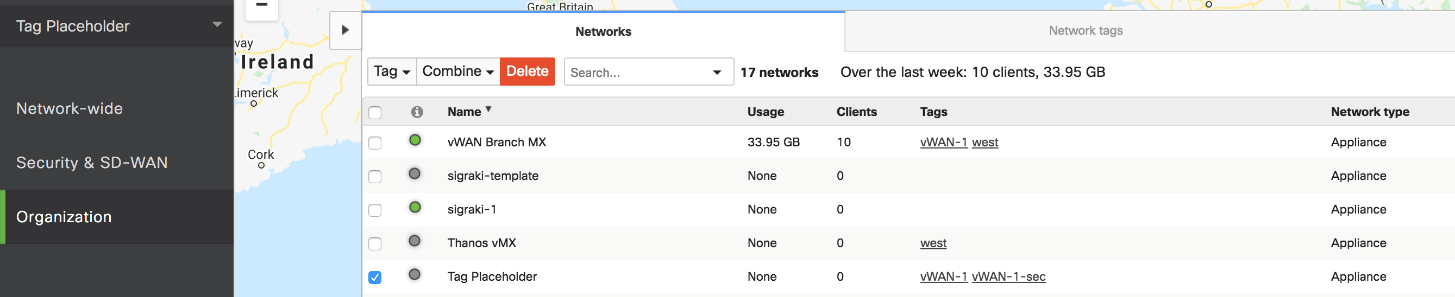
To create a Network tag, simply go to the **Organization > Monitor > Overview** page and expand out the Network list by clicking the left facing arrow at the top of the network list if it is not expanded by default. The picture below shows the **Organization > Monitor > Overview** page with the Network list already expanded. Demonstrated below is the tag being applied to the network (vWAN Branch MX) for vWAN-1:



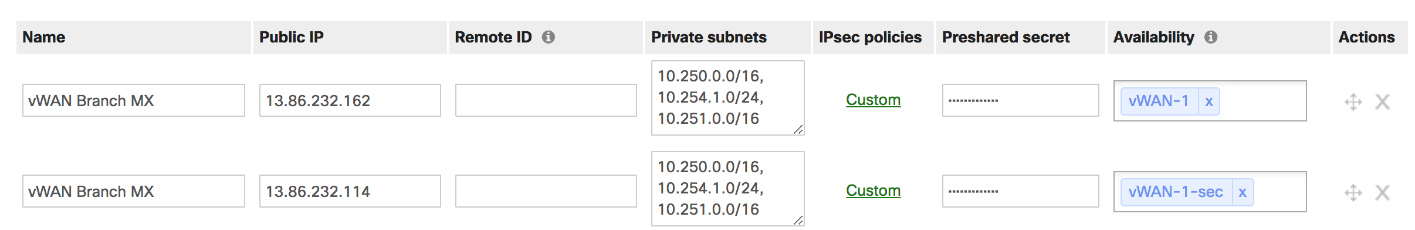
In addition to tagging the network, a placeholder network with no network devices must be created in order for Azure VPN Gateway failover to work. This is due to the fact that Instance 0 and 1 IPsec VPN peer configuration will have the tags of vWAN-1 and vWAN-1-sec as seen below:



Following the steps listed above for tagging a network, you will need to appropriately tag the placeholder network for tags as seen below:



In order to scale the configuration, availability tags are placed on the Meraki VPN config. This ensures that the correct IPsec config is being distributed to the branch sites. Both the tunnel name and availability tag match what was configured by the user. This can be seen by inspecting the same example that was displayed above.



## Summary

This document encompassed a step by step deployment guide along with a design guide.

## FAQ

Q: Why do I see 1.2.3.4 as a WAN uplink?

A: If you have two uplinks with the same public IP from the same WAN, you will receive 1.2.3.4 as a place holder address. NAT-T is currently unsupported by VWAN and the automation will check for this behavior, so 1.2.3.4 has been leveraged as a placeholder in the event additional checks have been disabled within the script.

Q: How do I ensure the automation remains working if I generate a new Meraki API Key?

A: You can navigate to the Azure Function in the Azure Portal and select **Configuration** under Settings. Under **Application settings**, modify the value for the setting called **meraki\_api\_key**.

### References

<https://documentation.meraki.com/zGeneral_Administration/Organizations_and_Networks/Organization_Menu/Manage_Tags>