Forcepoint NGFW and Azure Virtual WAN

Integration Guide

Forcepoint

Integration Guide

Tom Meaney
06 June 2020
Public

Table of Contents

Version	Date	Author	Notes
0.1	17 April 2020	Tom Meaney	First draft
0.2	27 April 2020	Mattia Maggioli	Review
0.3	05 May 2020	Tom Meaney	Update
0.4	07 May 2020	Neelima Ray	Added Troubleshooting chapter
0.5	07 May 2020	Mattia Maggioli	Review
0.6	06 June 2020	Jonathan Knepher	Review

Summary

This guide provides step by step instructions to setup an integration between **Forcepoint NGFW** and **Azure Virtual WAN**.

The software developed for this integration will automatically create and configure IPsec tunnels between a fleet of NGFW engines controlled by Forcepoint Security Management Center (SMC) and geographical Azure Virtual WAN sites, in order to create an SDWAN layer that can be used to route traffic between sites over the Azure Virtual WAN backbone.

The code and instructions provided enable system administrators to automatically:

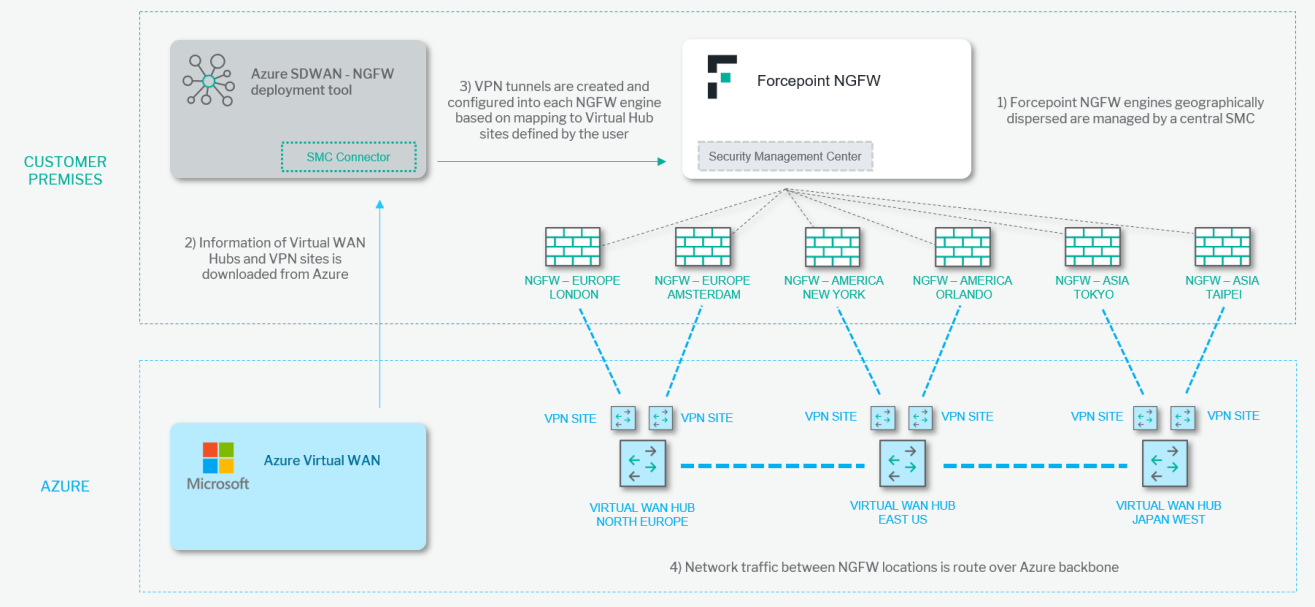
→ Create redundant VPN tunnels in each NGFW engine controlled by SMC using IPsec standard

→ Connect VPN tunnels in each NGFW engine to selected Azure Virtual WAN regions

Azure Virtual WAN resources are assumed to be in place, but instructions to create all Virtual WAN components are also provided so customers implementing Azure Virtual WAN for the first time can create all the necessary resources.

Configuration of routing, firewall policies and NAT settings are not included in the scope of this integration, to avoid unintended impact on network traffic in a wide range of different scenarios: after the SD-WAN layer is created, the user will have to perform the necessary changes to allow and route traffic over the VPN tunnels as deemed necessary.

A description of the workflow between the components involved in this POC is depicted in this diagram:



Caveats

The integration described in this document was developed and tested with the following product versions:

→ Forcepoint NGFW 6.7.2

→ Forcepoint SMC 6.7.3

Implementation options

Two implementation options are provided in this document

1. **Docker** – Uses a docker image where the integration component is already installed with all necessary dependencies: the user only has to run the container on an existing docker setup.
2. **Traditional** – requires manual deployment of the integration component inside a clean host machine (recommended) or an existing one, provided all requirements are satisfied.

The docker image for exporting risk level information has been tested working with

→ Docker v19.03.6

while the traditional version has been tested working with the following requirements

→ Ubuntu 18.04
  - with at least 1 GB RAM and 1vCPU
  - 250 MB free storage for the integration

→ Python 3.7

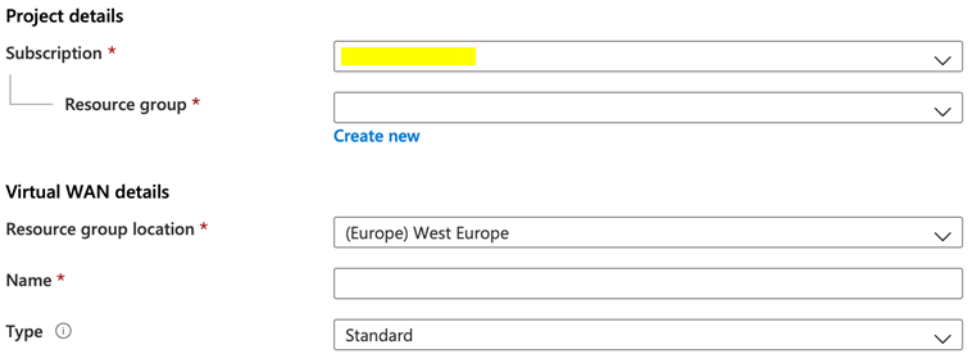
Setup Azure Virtual WAN resources

The integration component does not create Azure Virtual WAN resources but expects the Virtual WAN entities of **Hubs** and **Sites** to exist already: doing so the integration can be utilized by users who already have an Azure Virtual WAN setup and only want to automate the connection to sites secured by Forcepoint NGFW.

→ Skip to the next chapter if this is your case.

For users who don't have any Azure Virtual WAN resource in place, the following instructions provide step by step guidance to deploy everything necessary to achieve a full SD-WAN solution. The instructions are based on a traditional network routing based on static routes, without any dynamic routing protocol (i.e. BGP).

1. Open your Azure dashboard, click **Create Resource** and search for **Virtual WAN.** Once that page opens, click **Create** and fill in the required details:

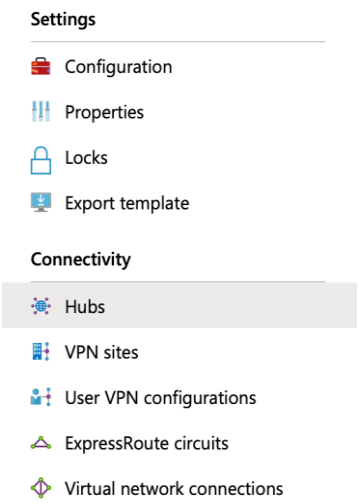


   Choice of **Resource group location** is typically driven by network latency or compliance requirement: for best network performance and lowest latency, the **Virtual WAN** resources should be deployed in the region closest to the NGFW sites that will be connected. For compliance region, the user might need to use more distant Azure locations.

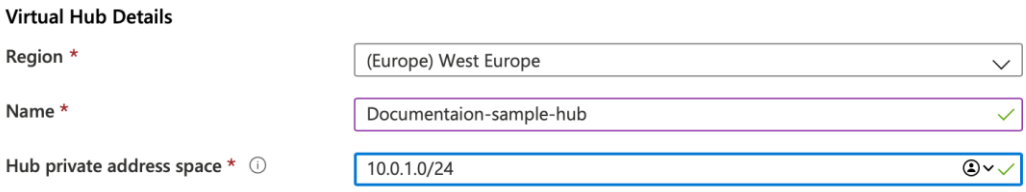
2. Once you have filled in the required information, click **Review + Create** and wait until the deployment is finished, once it has click **Go to resource**.

   Once the VWAN overview page is open, at least one Hub must be added: **Hubs** are regional containers of **VPN sites** and each NGFW location will be connected to one site. The typical association is then one Hub in each geographic region (e.g. EMEA) and one Site each office within that region.

3. To create one, find the **"Hubs"** section in the left menu:

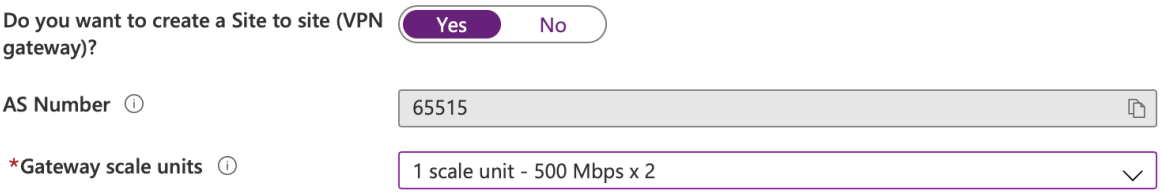


4. Once there, find the button at the top of the page to add a new hub, then fill in the required details:



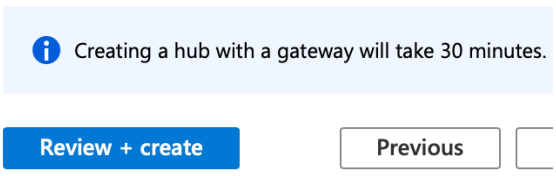
Select the appropriate region, name it and add a private address space: the **Hub private address space** choosen in this step must NOT overlap with any other range utilized in any network location that will be connected to the Virtual WAN.

Once you have added the required fields, click **Next: Site to Site** and you will be presented with the next screen:

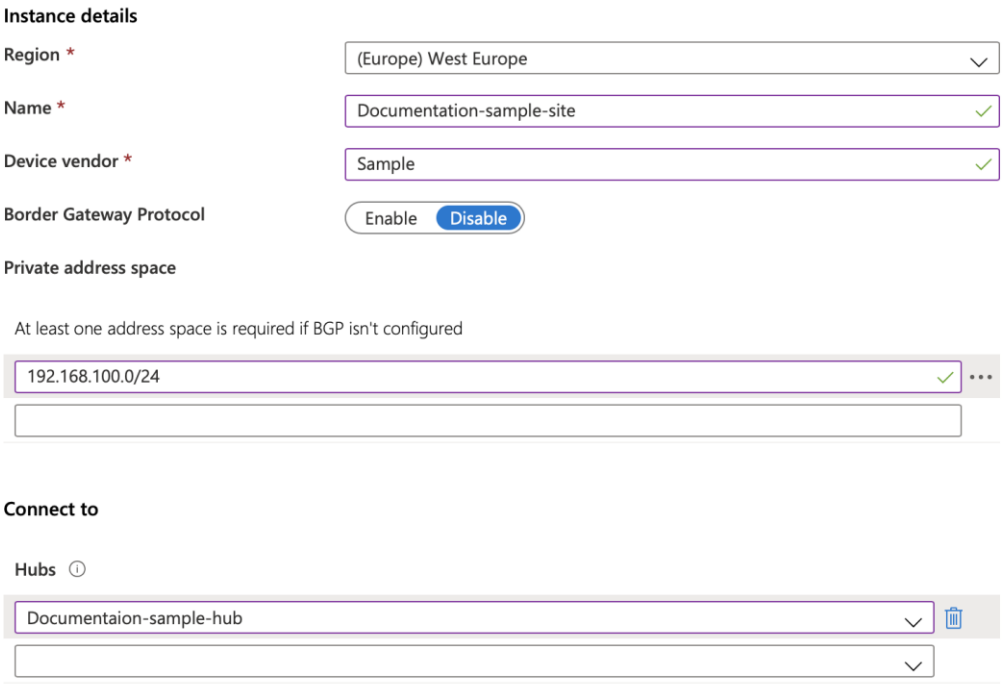


5. Select **Yes** to create a **Site to Site (VPN gateway)** and select the appropriate gateway scale unit that you require: this is based on the estimated throughput that every VPN tunnel must provide.

   Next, click **Review + Create** at the bottom of the screen, then click **Create** at the bottom of the next screen, this will begin the deployment of the hub.



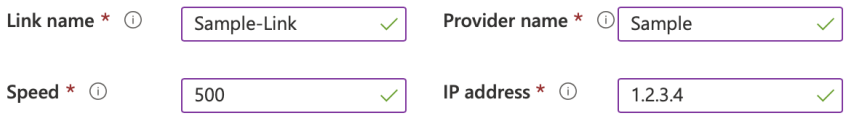
6. Once the deployment is completed, navigate back to the overview page for the VWAN you have created and select **VPN Sites** from the left menu. Here we will add a VPN site for this hub to allow us to connect from our NGFW engines in one office.
7. Click the **Create Site** button at the top of the page and fill in the requested details:



The **Private address space** fields must be filled with the actual network ranges utilized behind the NGFW engines that will connect to this VPN Site inside Azure Virtual WAN: by doing so routing information for all networks used in each office will propagate across the Virtual WAN resources of Azure.

Connect the site to the hub and click **Next: Links** at the bottom of the screen.

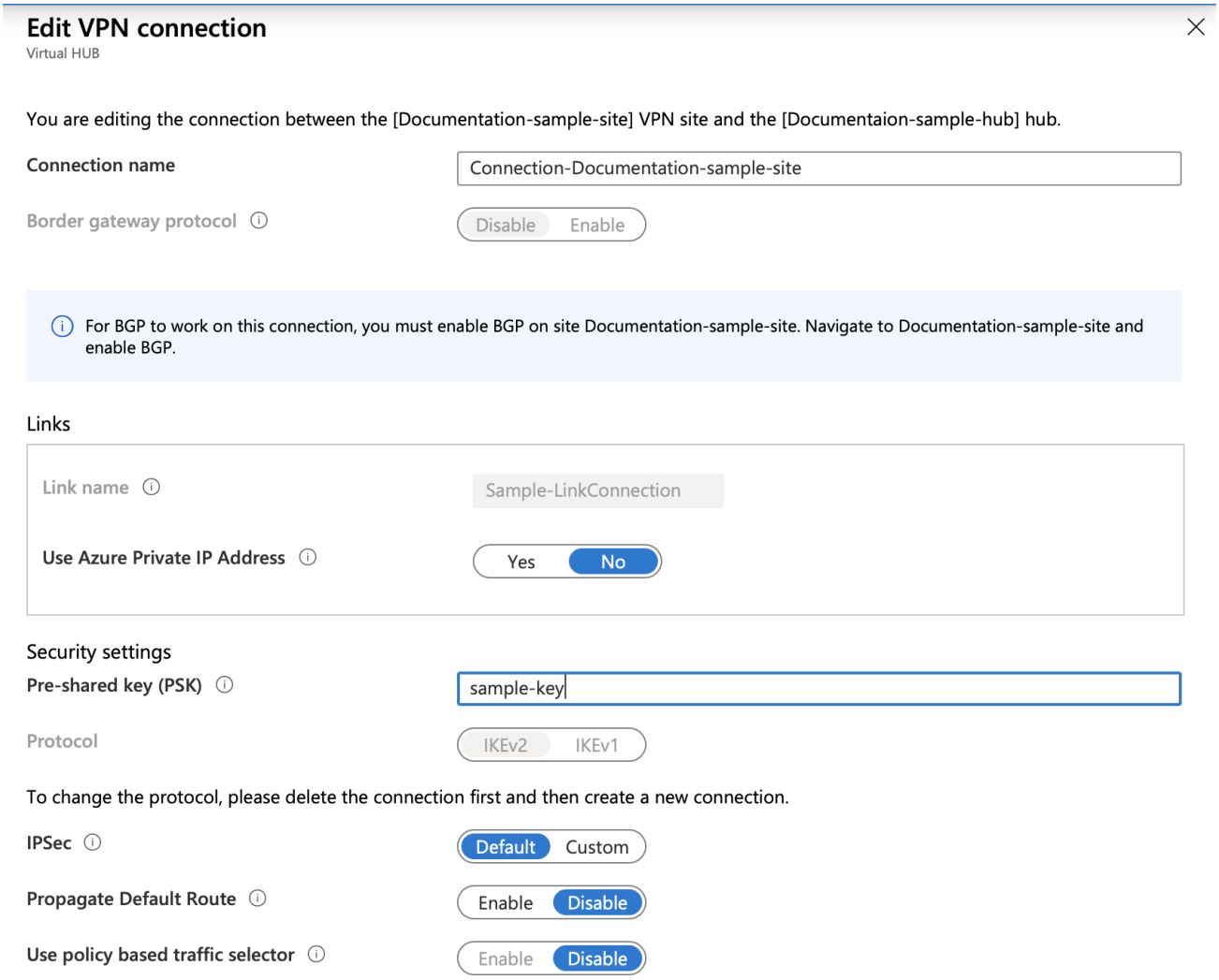
8. Fill in the required information for the **IP address** field (the external IP of the NGFW at the other side of the VPN tunnels). Next click **Review + Create** at the bottom of the screen, then click **Create** at the bottom of the next screen, this will begin the deployment of the site. This may take some time.



Once deployment is completed, a regional Hub and a VPN site are created.

Next we need to set the **Pre-shared key** for the IPsec tunnels.

9. Open your newly created hub, click **VPN (Site-to-Site)** and edit the link using the three dots menu at the far right of the entry. This will open the following dialog which will allow you to set your key:



Once done click **Save**.

You can now proceed to the next chapter or repeat steps 3 to 9 if you have other regions and sites to create.

Configuration of security protocols and IPsec policies

By default VPN tunnels created as a result of this integration will be assigned with the default **Gateway Profile** available inside the SMC: this includes all cyphers and IPsec policies supported by NGFW and the ones used in the IPsec tunnel will be negotiated during the connection phase.

→ User can create its own **Gateway Profile** with only specific protocols and cyphers enabled, instructions are provided at this link: https://help.stonesoft.com/onlinehelp/StoneGate/SMC/6.7.0/GUID-02FC2E17-34B9-4296-B0D3-ED5578D32072.html#GUID-02FC2E17-34B9-4296-B0D3-ED5578D32072

→ Cyphers and protocols supported by Azure Virtual WAN are listed at this link: https://docs.microsoft.com/en-us/azure/virtual-wan/virtual-wan-ipsec

The custom profile can be assigned to the IPsec tunnels created as explained in the next chapter. If you do not specify a **Gateway Profile** name in the config, it will default to using the default profile which allows all capabilities.

Configuration files and automated creation of VPN tunnels

Information needed to automatically create and configure all IPsec tunnels between NGFW engines and Azure Virtual WAN are stored in two configuration files used by the integration component.

The first file contains all Azure Virtual WAN sites information:

1. Go to the Azure portal, open your **Virtual WAN** overview page and click on **VPN Sites** on the left menu. This will open a new page and at the top you will see the button **Download Site-to-Site VPN configuration**:

2. Click **Download Site-to-Site VPN configuration** and wait for Azure to generate the json file for you. Download the file and save it in a secure location, in the rest of this document the file is named **azure-config.json**.
3. Edit **config.yaml** located in the config folder of the extracted package if you have used the traditional method, or if you have used the docker method you will need to create the file and add all the information related to the SMC, the network parameters for each NGFW engine and the details of the Azure Virtual WAN resources from **azure-config.json**:

A description of each value of the **config.yaml** file is provided in the following table:

Field	Examples	Notes
smc-host	192.168.100.20	The host address of the SMC component
smc-port	8082	The port used by the SMC component, usually 8082
smc-api-key	abdefghijklmnop1234567890	The API key generated by the user in the SMC
smc-api-version	6.7	The installed version of the SMC without the patch version, for example, if the installed version is 6.7.3 then the value for this will be 6.7
gateway-profile	MyProfile	Use this field to specify a custom **Gateway Profile** to be used, if left blank it will default to the default profile available in SMC which allows all capabilities
ngfw-azure-locations	- {'NGFW Paris': 'North Europe'} - {'NGFW London': 'North Europe'} - {'NGFW New York': 'East US'} - {'NGFW Cabot Cove ': 'East US'} …	The pairing between the NGFW engines and the Azure region, this allows to associate sites secure by NGFW and Azure Virtual WAN Sites so VPN tunnels are automatically created inbetween each device/location. The format is a **yaml array** made of {'Engine name': 'Azure location'} To find the values for these fields you need to match the name of the NGFW engine to the name of the Azure location you wish to connect it to: • **Engine name**: can be found in the SMC dashboard • **Azure location**: can be found in **azure-config.json** downloaded from Azure as explained in the previous chapter

		A full list of Azure locations is available here: https://azure.microsoft.com/en-us/global-infrastructure/locations/
ngfw-external-interface-ip	{ 'Engine 1': '192.168.122.100', …}	This is to specify the IP addresses of the NGFW interfaces that will serve as VPN endpoint in each NGFW engine. **(The format is a python dictionary)**
ngfw-external-address-space	{ 'Engine 1': '192.168.122.0/24', …}	Address space for the interface specified above **(The format is a python dictionary)**
ngfw-internal-vpn-interface-ip	{ 'Engine 1': '172.16.16.1', …}	IP address for the VPN Tunnel interface **(The format is a python dictionary)**
ngfw-internal-vpn-address-space	{ 'Engine 1': '172.16.16.1/32', …}	Address space for the VPN Tunnel interface specified above **(The format is a python dictionary)**

4. If not already enabled, API access to the SMC must be enabled in order to automate the tasks on each NGFW engine. The instructions to Enable the SMC API and generate an API key can be found in the official documentation at this link:

   https://www.websense.com/content/support/library/ngfw/v67/rfrnce/ngfw_670_rg_smc-api_a_en-us.pdf

Implementation - Docker

fp-ngfw-azure-sdwan is the docker image which contains the integration.

This integration is provided as a docker image which can run on a docker host on premise, or on any cloud-based docker service with network access to the SMC and Azure Virtual WAN.

In order to setup the integration using Docker do as follows:

1. Login to the docker registry with the following command

   docker login docker.frcpnt.com

   User: fp-integrations
   Pass: t1knmAkn19s

2. Pull the image from the registry using the following command.

   docker pull docker.frcpnt.com/fp-ngfw-azure-sdwan

3. Locate the configuration files named **config.yaml** and **azure-config.json** created in the previous chapters of this document.
4. Upload these files to a location where they can be accessed over http/https (e.g. a web server, an AWS S3 bucket, Azure Blob Storage): this will make sure the configurations are not lost in case the docker container is decommissioned. Make sure the files are not available to unintended people since they contain sensitive information.
5. Run the integration tool with the following command, replacing the folder path with the actual path to the directory where you want to download the config files to on your docker host and **<SMC_CONFIG_URL>** and **<AZURE_CONFIG_URL>** with the URL of the config files (SMC config and Azure config)

   docker run -v "/folder_on_docker_host_for_config_files/app/config/" -e SMC_CONFIG <smc_config_url> -e AZURE_CONFIG <azure_config_url> docker.frcpnt.com/fp-ngfw-azure-sdwan

6. While running you will see some output in the terminal along the lines of

   Interface doesn't exist, creating it now...

   IPSec tunnel doesn't exist, creating it now…

   this shows that the integration is running and is creating the required elements for the setup of the IPSec tunnels.
7. Once done you will see this message in the console

   Completed. Please verify success in the SMC console as per the instructions in the guide...

   Once you see this you should go to the **Verify and commit changes** chapter.

Implementation - Traditional

The solution described in this chapter requires the following files

→ **fp-ngfw-azure-sdwan-connector-v1.tgz** available at this link https://frcpnt.com/ngfw-azure-sdwan-latest

fp-ngfw-azure-sdwan-connector-v1.zip contains all files necessary to setup and run the integration and create the connection to the Azure Virtual WAN. This is designed to be deployed on an Ubuntu 18.04 machine (with at least 1 GB RAM and 1vCPU and 250 MB of free storage for the integration) with Python 3.7.

Setup and run the integration

1. Make sure the system has Python 3.7 and pip installed, if not run the following command:

   sudo apt install python3

   sudo apt install python3-pip

2. Download and unpack **fp-ngfw-azure-sdwan-connector-v1.tgz** to a directory of choice.
3. Navigate to the **fp-ngfw-azure-sdwan** directory and open the **config** directory. Drop the configuration files **config.yaml** and **azure-config.json** inside the **config** directory.
4. Return to **fp-ngfw-azure-sdwan** directory and run the following command to run the integration:

   pip3 install -r requirements.txt

   python3 app_runner.py

5. To confirm it is running you should see output to the terminal like

   Interface doesn't exist, creating it now...

   IPSec tunnel doesn't exist, creating it now…

   This will be repeated the same number of times as the number of tunnels you wish to create.
6. Once done you will see this message in the console

   Completed. Please verify success in the SMC console as per the instructions in the guide...

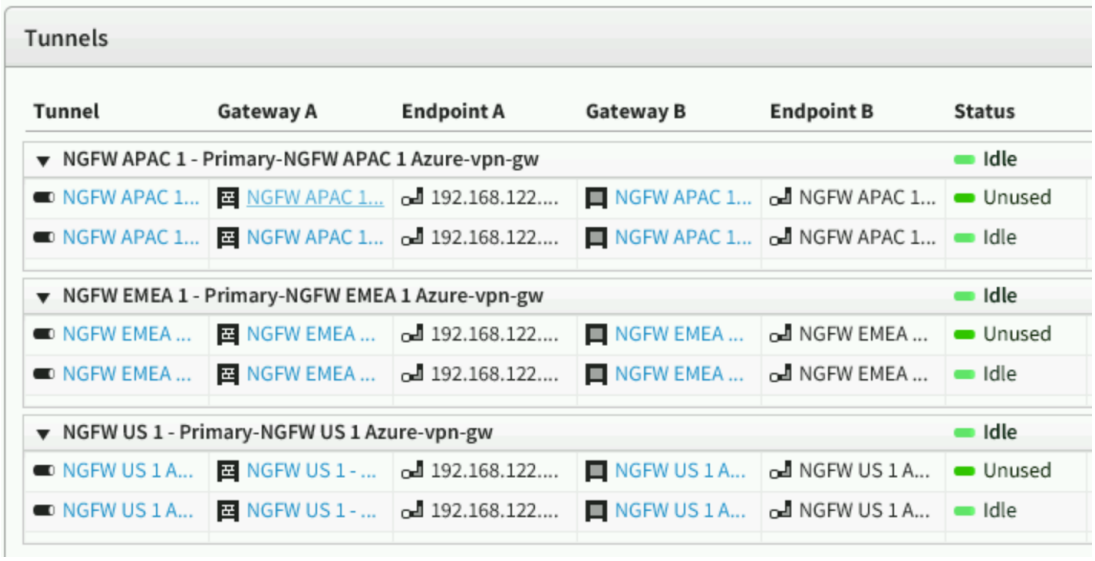
   Once you see this you should go to the **Verify and commit changes** chapter.

Verify and commit changes

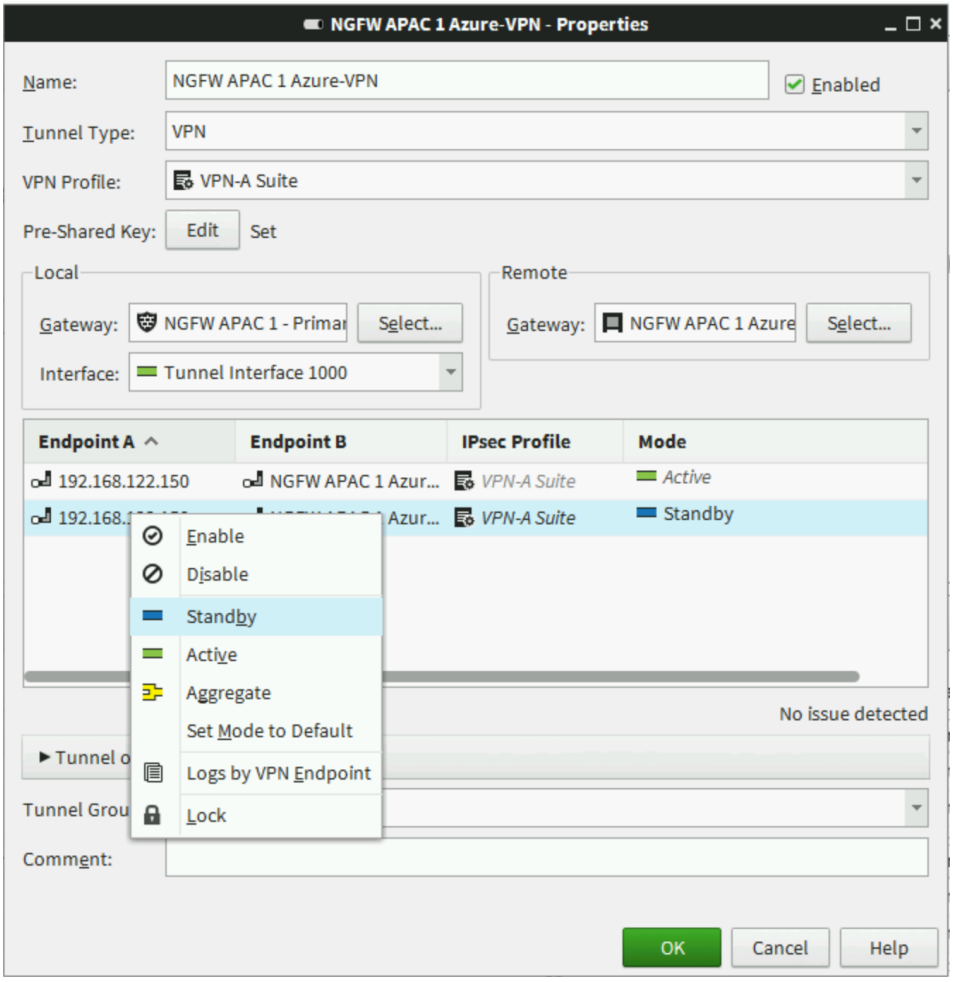
VPN tunnels created via automated tools are configured as **Active** by default but only one will be used: we suggest changing the configuration of the two tunnels in each NGFW engine so that one tunnel is **Active** and the other one is **Standby**:

1. Open the SMC UI, go to the **Home** tab and verify that you can see the new tunnels in the VPN pane:

2. In the list of **Tunnels** click on a single tunnel

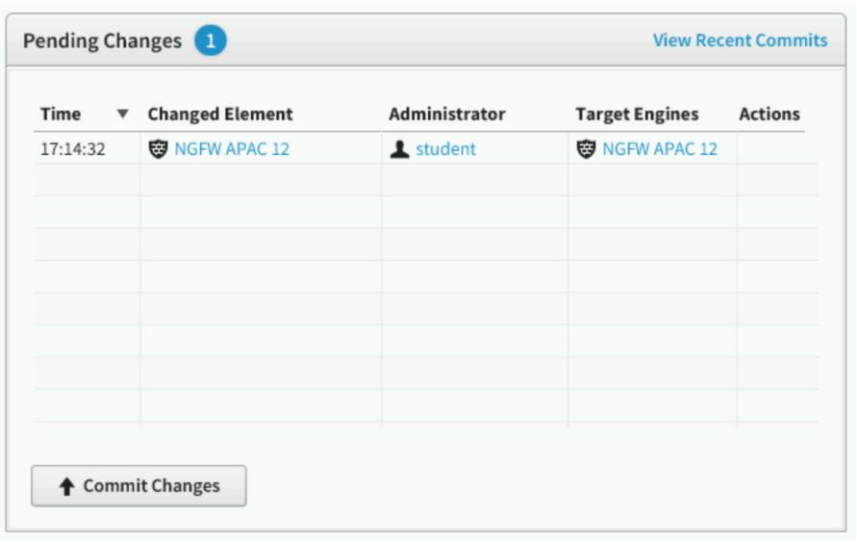


3. Right-click on the second tunnel available in each NGFW engine and set it to **Standby**



Now perform the necessary changes to routing, firewall policies and NAT to allow the network traffic to reach its destinations over the SD-WAN layer, as deemed necessary. This step is intentionally left to the user, since networking configuration is different and specific to each scenario.

4. Once all settings are configured, click on **Commit Changes** button in the **Pending Changes** window on the **Home** dashboard, this will write our changes to the NGFW engines:



Once changes are committed, traffic will be routed over the SD-WAN layer according to the routing policies set by the user.

Troubleshooting

Follow these steps to identify issues impacting the normal operation of the integration described in this document. However, please bear in mind this guide is not an exhaustive list of solutions for every scenario/problem.

Docker Implementation

Validate the prerequisites

Make sure the prerequisites described in the **Summary** chapter are all satisfied:

→ Check the versions of Forcepoint NGFW SMC in use are listed as compatible

Forcepoint NGFW 6.7.2

Forcepoint SMC 6.7.3

→ API access to the SMC must be enabled in order to automate the tasks on each NGFW engine. The instructions to Enable the SMC API can be found in the official documentation at this link:

https://www.websense.com/content/support/library/ngfw/v67/rfrnce/ngfw_670_rg_smc-api_a_en-us.pdf

→ Docker images for this integration have been tested with

Docker 19.03.6

Check network connectivity

Make sure firewalls or other security appliances are not impacting the network connectivity necessary for the operation of all components involved into this integration:

→ Check the docker host machine has connectivity to SMC: execute the following command on docker host machine:

ping -c 2 SMC_PRIVATE_IP_ADDRESS

Replacing the SMC_PRIVATE_IP_ADDRESS with your Forcepoint SMC private IP address or the hostname. Once done check the result is similar to below:

```
PING SMC_PRIVATE_IP_ADDRESS.url (10.10.120.12) 56(84) bytes of data.
64 bytes from 10.10.120.12 (10.10.120.12): icmp_seq=1 ttl=128 time=179 ms
64 bytes from 10.10.120.12 (10.10.120.12): icmp_seq=1 ttl=128 time=181 ms
```

Check dependencies are installed

Make sure the software dependencies needed by the components involved into this integration are installed:

→ Check the host machine has docker installed: Execute the following command on the host machine:

docker info

Check the server version in the output is similar to below:

Server Version: 19.03.8

Check all components are configured and running properly

Make sure the products and services involved into this integration are configured as expected and they are running:

→ Check the last line of the output is similar to below after the docker run command finishes

Completed. Please verify success in the SMC console as per the instructions in the guide...

→ In the SMC UI, go to the **Home** tab and verify that you can see the new tunnels in the VPN pane:

Traditional Implementation

Validate the prerequisites

Make sure the prerequisites described in the **Summary** chapter are all satisfied:

→ Check the versions of Forcepoint NGFW SMC in use are listed as compatible

Forcepoint NGFW 6.7.2

Forcepoint SMC 6.7.3

→ This integration requires to be run on an Ubuntu 18.04 machine with at least 1 GB RAM and 1vCPU and 250 MB of free storage for the integration

→ User needs sudo permissions for this integration

→ API access to the SMC must be enabled in order to automate the tasks on each NGFW engine. The instructions to Enable the SMC API can be found in the official documentation at this link:

https://www.websense.com/content/support/library/ngfw/v67/rfrnce/ngfw_670_rg_smc-api_a_en-us.pdf

→ Check the user can download the file with the below command:

wget --content-disposition https://frcpnt.com/ngfw-azure-sdwan-latest

Check network connectivity

Make sure firewalls or other security appliances are not impacting the network connectivity necessary for the operation of all components involved into this integration:

→ Check the Ubuntu host machine can reach the Forcepoint SMC host machine over the network: execute the following command on host machine:

ping -c 2 SMC_PRIVATE_IP_ADDRESS

Replacing the SMC_PRIVATE_IP_ADDRESS with your Forcepoint SMC private Ip address or the hostname. Once done check the result is similar to below:

```
PING SMC_PRIVATE_IP_ADDRESS.url (10.10.120.12) 56(84) bytes of data.
64 bytes from 10.10.120.12 (10.10.120.12): icmp_seq=1 ttl=128 time=179 ms
64 bytes from 10.10.120.12 (10.10.120.12): icmp_seq=1 ttl=128 time=181 ms
```

Check dependencies are installed

Make sure the software dependencies needed by the components involved into this integration are installed:

→ Check python3.6 is installed: Execute following command on host machine:

python3 --version

Check the output is similar to below:

Python 3.7.x

→ Check python3.6 is installed: Execute following command on host machine:

pip3 --version

Check the output is similar to below:

pip 9.0.1 from /usr/lib/python3/dist-packages

Check all components are configured and running properly

Make sure the products and services involved into this integration are configured as expected and they are running:

→ Check the last line of the output is similar to below after the docker run command finishes

Completed. Please verify success in the SMC console as per the instructions in the guide...

→ In the SMC UI, go to the **Home** tab and verify that you can see the new tunnels in the VPN pane:

Forcepoint

forcepoint.com/contact

About Forcepoint

Forcepoint is the global human-centric cybersecurity company transforming the digital enterprise by continuously adapting security response to the dynamic risk posed by individual users and machines. The Forcepoint human point system delivers risk-adaptive protection to continuously ensure trusted use of data and systems. Based in Austin, Texas, Forcepoint protects the human point for thousands of enterprise and government customers in more than 150 countries.