

Mobile User To Cloud Application Connectivity

Overview

This quickstart guide will provide all the steps to create a secure service between a remote user and an application hosted in Azure Cloud using NetFoundry Overlay Fabric (NFOF).

Important

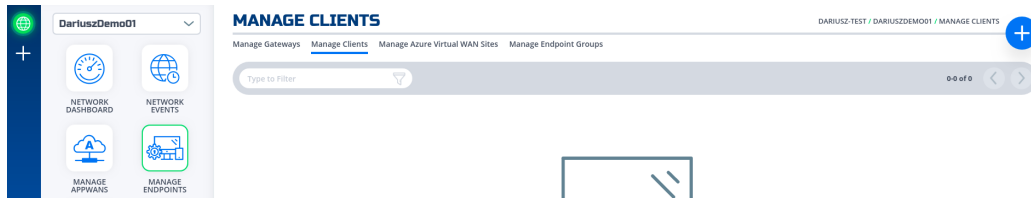
Assumption is that the NF Fabric is already up and the NF Client is installed.

Through NF Web Console UI

Create and install NF Client

This section will guide a user through the steps on how to create a client in the NF Console UI. Then, it will provide links to Guides on how to install the NetFoundry Client Software for Windows and MAC Clients, including the registration with the NF Network Fabric.

1. Navigate to Manage Clients Page



2. Click on + sign in the top right corner.

3. Fill in the required information and click on "Create"

CREATE A NEW CLIENT HELP ESC

Enter your client attributes

CLIENT NAME REQUIRED

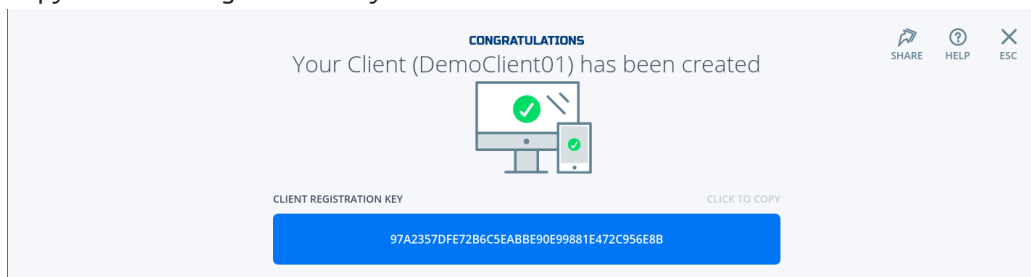
DemoClient01

LOCATION REQUIRED

US East

CREATE

4. Copy the Client Registration Key



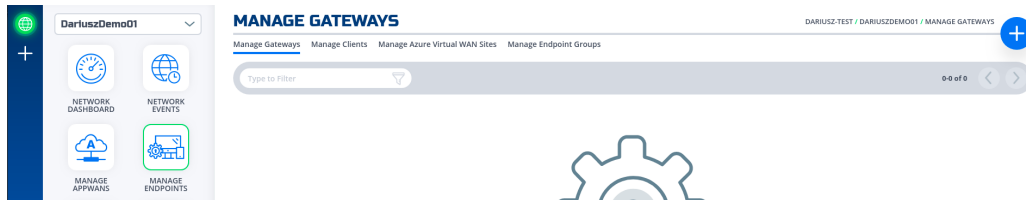
5. Install the NF Client Software by following the directions at the appropriate OS link

- a. Window
- b. Mac

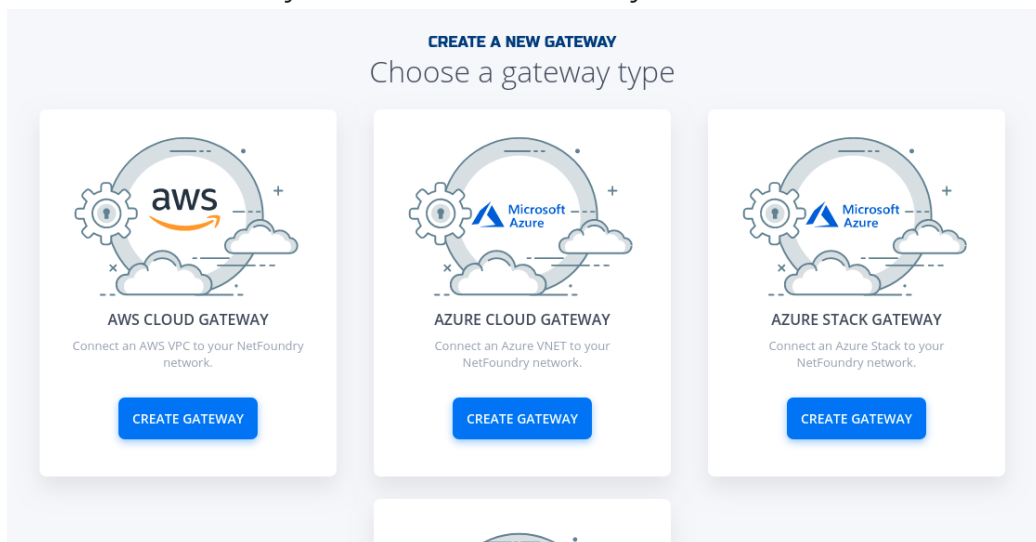
Create and Deploy NF Azure Gateway

This section will guide a user through the steps on how to create a NF Manage Gateway in the NF Console UI and install it in the Azure vNet.

1. Navigate to Manage Gateways Page
2. Click on + sign in the top right corner.

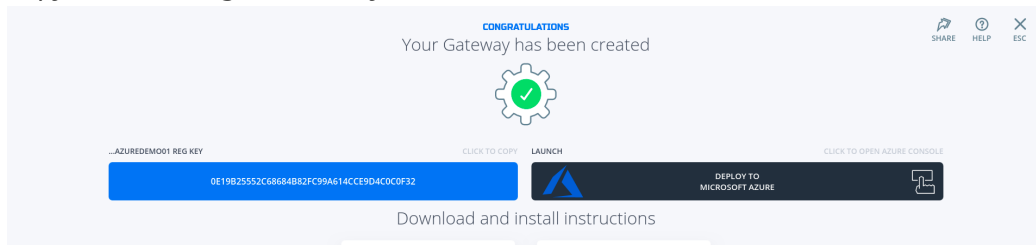


3. Click on "Create Gateway" on the Azure Cloud Gateway Card



4. Fill in the required information and click on "Create"

5. Copy the Client Registration Key



6. Click on "Deploy to Microsoft Azure". It will take you to the Azure Portal and ask you for your login credentials.

7. You will be presented with the template that needs to be filled. The first section is the Basics regarding your Subscription and Resource Group this gateway will be deployed in.

BASICS

Subscription *	NetFoundry Non-Prod	▼
Resource group *	nf-sandbox	▼
	Create new	
Location *	(US) East US	▼

8. The second section related to resources associated with this gateway. e.g. vm name, ip address space, security groups, etc. you will paste the registration key copied in step 5. You will also need the public ssh key to use for access to this gateway remotely.

SETTINGS

Location	westus
Network Interface Name	azuredemo01-if
Security Group Name	azuredemo01-sg
Virtual Network Name	azuredemo01-vnet
Address Prefix	10.0.8.0/24
Subnet Name	default
Subnet Prefix	10.0.8.0/24
Public Ip Address Name	azuredemo01-ip
Public Ip Address Type	Dynamic
Public Ip Address Sku	Basic
Virtual Machine Name	azuredemo01 ✓
Virtual Machine RG	nf-sandbox
Os Disk Type	Premium_LRS
Virtual Machine Size	Standard_B1ms
Nfreg Key * ⓘ ✓
Admin Username ⓘ	nfadmin
Ssh Key Data * ⓘ	ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQACjga67wcoISXaD1bswknLrejRYtZ... ✓

9. You will need to agree to Azure Marketplace Terms and Conditions and click to "Purchase" to continue.

TERMS AND CONDITIONS

[Azure Marketplace Terms](#) | [Azure Marketplace](#)

By clicking "Purchase," I (a) agree to the applicable legal terms associated with the offering; (b) authorize Microsoft to charge or bill my current payment method for the fees associated the offering(s), including applicable taxes, with the same billing frequency as my Azure subscription, until I discontinue use of the offering(s); and (c) agree that, if the deployment involves 3rd party offerings, Microsoft may share my contact information and other details of such deployment with the publisher of that offering.

☒ I agree to the terms and conditions stated above

Purchase

10. If the NF Gateway was deployed successfully. Here is the view of the Resource Group and NF Console UI.

The screenshot displays the Azure portal interface. On the left, the 'nf-sandbox' resource group is selected, showing a list of resources including 'azuredemo01-if', 'azuredemo01-ip', 'azuredemo01-sg', and 'azuredemo01-vnet'. The main pane shows the 'MANAGE GATEWAYS' console for 'DariuszDemo01'. The console includes a search bar, a table with columns for Gateway Label, Status, Type, Location, and Cloud Provider, and a list of gateways. A notification on the right indicates a successful deployment of 'Microsoft.Template' to the resource group 'nf-sandbox'.

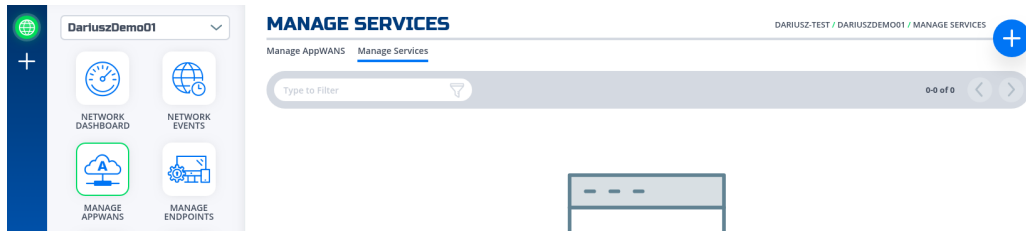
Gateway Label	Status	Type	Location	Cloud Provider
AzureDemo01	Online	Azure Private Gateway	Oregon	Azure

11. Done

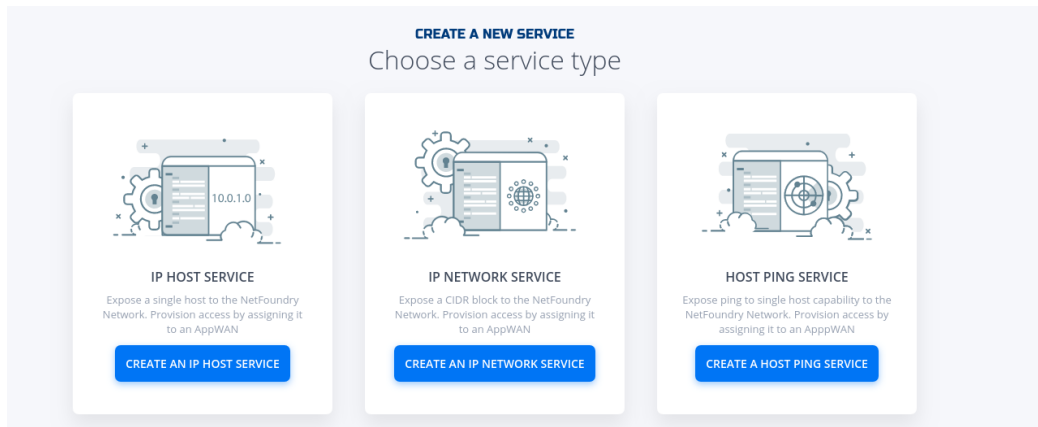
Create IP Host Service

This section will guide a user through the steps on how to create a NF Service.

1. Navigate to Manage Services Page under Manage Appwans
2. Click on + sign in the top right corner.



3. Click on "Create an IP Host Service"



4. Fill in the required information for SSH and click on "Create"

CREATE A NEW IP HOST SERVICE

Enter your service attributes

SERVICE NAME REQUIRED

DemoServiceSsh

GATEWAY REQUIRED

AzureDemo01

IP ADDRESS REQUIRED PORT/RANGE REQUIRED

10.0.8.5 22

INTERCEPT IP ADDRESS INTERCEPT PORT/RANGE

10.0.8.5 22

PROTOCOL TYPE REQUIRED

TCP

ADVANCED OPTIONS OPEN TO EDIT DETAILS

ADVANCED OPTIONS

CREATE

5. If successfully, the service is green.

The screenshot shows a web interface for managing services. On the left is a sidebar with a green status indicator and a plus icon, and four menu items: NETWORK DASHBOARD, NETWORK EVENTS, MANAGE APPWANS, and MANAGE ENDPOINTS. The main area is titled 'MANAGE SERVICES' and has a dropdown menu set to 'Demo01'. Below the title are tabs for 'Manage AppWANS' and 'Manage Services'. A search bar labeled 'Type to Filter' is present. Below the search bar is a table with columns: Service Name, Type, Protocol, IP Address, Intercept IP, and Port Range. The table contains one row for 'DemoServiceSsh' with a green status indicator, 'IP Host', 'TCP', '10.0.8.5', '10.0.8.5', and '22 - 22'. A blue plus icon is in the top right corner.

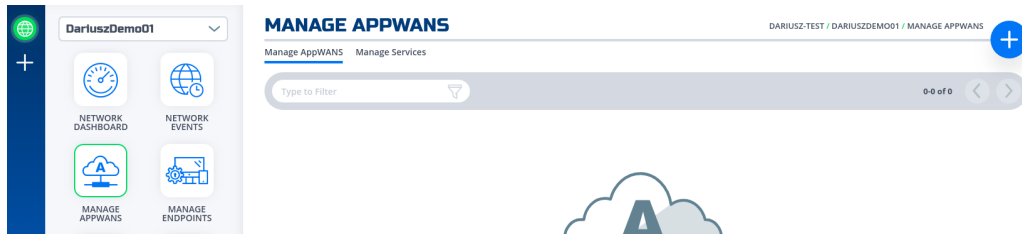
Service Name	Type	Protocol	IP Address	Intercept IP	Port Range
DemoServiceSsh	IP Host	TCP	10.0.8.5	10.0.8.5	22 - 22

6. Done

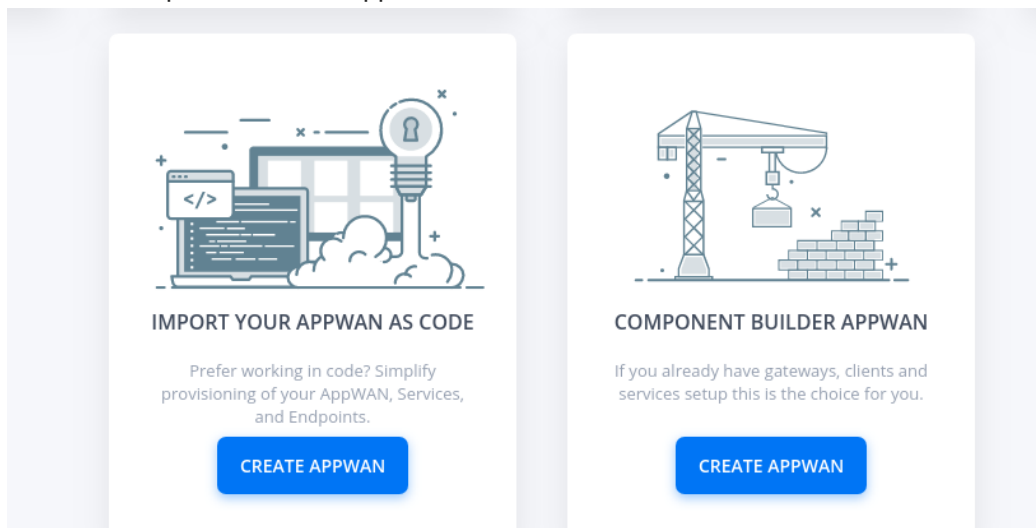
Create AppWan

This section will guide a user through the steps on how to enable service connectivity to users by creating an appwan.

1. Navigate to Manage AppWANS Page under Manage Appwans
2. Click on + sign in the top right corner.



3. Click on "Component Builder Appwan"



4. Move the desired client (e.g. DemoClient01) from "Available" Clients to "Selected" Endpoints. Move the desired service (e.g. DemoServiceSsh) from "Available" to "Selected"

Services.

CREATE A NEW APPWAN

Choose from existing components, or add new ones

1

APPWAN NAME

REQUIRED

DemoAppWan

2

ADD CLIENTS, GATEWAYS, OR ENDPOINT GROUPS

Search for Endpoints

AVAILABLE GROUPS

ADD NEW +

AVAILABLE CLIENTS

ADD NEW +

AVAILABLE GATEWAYS

ADD NEW +

AzureDemo01

SELECTED ENDPOINTS

DemoClient01

3

ADD SERVICES

Search for a Service

AVAILABLE SERVICES

ADD NEW +

SELECTED SERVICES

DemoServiceSsh

CREATE

5. Click on "Create".

YOUR APPWAN SUMMARY

Your AppWAN has been created! A network summary is below.

What's next? Finish connecting your network by registering new clients and gateways.

NEW CLIENTS
Share Client Registration Info

NEW GATEWAYS
Tap to Launch and Register

1 APPWAN NAME
DemoAppWan

2 ENDPOINTS
CLIENTS
DemoClient01

3 SERVICES
SERVICE DEFINITIONS
DemoServiceSsh

4 ENDPOINT GROUPS
GROUPS

SHARE NEW CLIENTS

REGISTER NEW GATEWAYS

Want to add another environment with the same services or endpoints?

TAP TO CLONE

6. Done

Test Connectivity to Application Server

To test connectivity, log in to the DemoClient01 and run `ssh username@privateIp`

```
nfadmin@azuredemoapp:~  
PS C:\>  
PS C:\>  
PS C:\> ssh nfadmin@10.0.8.5  
[nfadmin@azuredemoapp ~]$  
[nfadmin@azuredemoapp ~]$ ls  
[nfadmin@azuredemoapp ~]$ pwd  
/home/nfadmin  
[nfadmin@azuredemoapp ~]$  
[nfadmin@azuredemoapp ~]$
```

Programmatically

via Python and Terraform

Python Modules

For the code clarity, we have broken down the code into multiple Python modules

1. NF REST CRUD (Create,Read, Update and Delete) operations
2. Get MOP Session Token
3. Create NF Network
4. Create NF Gateway(s)
5. Create NF Service(s)
6. Create NF AppWan(s)
7. Wrapper Script to Create NF Resources based on Resource yaml file

Environment Setup Requirements

1. `~/env` to store NF Credentials in (e.g. `clientId`, `clientSecret`) to obtain a session token for NF API
2. Export Azure Credentials (e.g, `export ARM_TENANT_ID, ARM_CLIENT_ID, ARM_CLIENT_SECRET, ARM_SUBSCRIPTION_ID`) to enable resource gateway creation in Azure Resource Group via Terraform.
3. Terraform and Python3 installed in path.

Additional Information:

1. The new Resource Group in Azure is created based on then name provided in Resource yaml, if one does not exists already in the same region (e.g. centralus). The action delete gateway will delete the RG as well even if it was an existing RG. If one does not want to delete the RG, the command `terraform state rm "{tf resource name for RG}"` needs to be run before running the gateway delete step. This will ensure that the RG is not deleted.
2. A new vNet will be created and NF Gateway will be placed in it.
3. Environment means the NF Console Environment used (e.g. production), not Azure.

Steps

1. Clone this repo (git clone <https://github.com/netfoundry/mop.git>)
2. Update Resource yaml file with the desired options to feed into the wrapper script as described in the following code snippet. All Resource.yml Options
3. Run this from the root folder to create GW in NF Console UI and Azure.

```
python3 quickstarts/docs/python/nf_resources.py --file quickstarts/docs/python/nf_resources.yml
```

Required Configuration Parameters for Gateway Creation

```
environment: production
network_action: get
network_name: DemoNet01
gateway_list:
- action: create
  cloud: azure
  count: 1
  names: []
  region: westus
  regionalCidr: [10.20.10.0/24]
  regkeys: []
  resourceGroup:
    name: demoPythonTerraform01
    region: centralus
  tag: TerraformDemo
terraform:
  bin: terraform
  output: 'no'
  source: ./quickstarts/docs/terraform
  work_dir: .
```

4. After the script is run successfully, one can see that the gateway name and registration key were saved in Resource.yml file. The name is created automatically based on region and gateway type joined with x and gateway count (AZCPEGW means an azure type gateway in NF console). One can create more than one gateway in the same region by increasing the count to more than 1.

```
environment: production
gateway_list:
- action: create
  cloud: azure
  count: 1
  names:
  - AZCPEGWx0xWESTUS
  region: westus
  regionalCidr:
  - 10.20.10.0/24
  regkeys:
  - 21DB86724EC3F31C11C1C9D68CE5ECD6A06F057E
  resourceGroup:
```

```

name: demoPythonTerraform01
region: centralus
tag: TerraformDemo
network_action: get
network_name: DemoNet01
terraform:
  bin: terraform
  output: 'no'
  source: ./quickstarts/docs/terraform
  work_dir: .

```

5. Create a test server vm on the same vNet if not already present.

6. Update the Resoure.yaml file to include the Service option to create the NF service on the gateway create in the previous step. Don't forget to change the action on the gateway to "get".

```

environment: production
gateway_list:
- action: get
  cloud: azure
  count: 1
  names:
  - AZCPEGWx0xWESTUS
  region: westus
  regionalCidr:
  - 10.20.10.0/24
  regkeys:
  - 21DB86724EC3F31C11C1C9D68CE5ECD6A06F057E
  resourceGroup:

```

```

name: demoPythonTerraform01
region: centralus
tag: TerraformDemo
network_action: get
network_name: DemoNet01
terraform:
  bin: terraform
  output: 'no'
  source: ./quickstarts/docs/terraform
  work_dir: .
services:
- action: create
  gateway: AZCPEGWx0xWESTUS
  ip: 10.20.10.5
  port: 22
  name:
  type: host

```

7. After the script run again successfully, the service section should have been populated with the service name as so.

```

services:
- action: create
  gateway: AZCPEGWx0xWESTUS
  ip: 10.20.10.5
  name: AZCPEGWx0xWESTUS--10.20.10.5--22
  port: 22
  type: host

```

MANAGE SERVICES / MANAGE SERVICES +

Manage AppWANS Manage Services

Type to Filter 1-1 of 1 < >

Service Name	Type	Protocol	IP Address	Intercept IP	Port Range	
<input type="radio"/> AZCPEGWx0xWESTUS--10.20.10.5-22	IP Host	TCP	10.20.10.5	10.20.10.5	22 - 22	...

8. Create a client endpoint if not already done so.

MANAGE CLIENTS / MANAGE CLIENTS +

Manage Gateways Manage Clients Manage Azure Virtual WAN Sites Manage Endpoint Groups

Type to Filter 1-1 of 1 < >

Client Name	Status	Location	Client Type	Version	
<input type="radio"/> DemoClient01	Online	US East	WIN64	3.6.6.11077	...

9. Update the Resoure.yaml file to include the AppWan option to create the NF AppWan tying the gateway, client and service created in the previous steps. Don't forget to change the action on the service option to "get".

```

environment: production
gateway_list:
- action: get
  cloud: azure
  count: 1
  names:
  - AZCPEGWx0xWESTUS
  region: westus
  regionalCidr:
  - 10.20.10.0/24

```

```
regkeys:
- 21DB86724EC3F31C11C1C9D68CE5ECD6A06F057E
resourceGroup:
  name: demoPythonTerraform01
  region: centralus
  tag: TerraformDemo
network_action: get
network_name: DemoNet01
services:
- action: get
  gateway: AZCPEGWx0xWESTUS
  ip: 10.20.10.5
  name: AZCPEGWx0xWESTUS--10.20.10.5--22
  port: 22
  type: host
terraform:
  bin: terraform
  output: 'no'
  source: ./quickstarts/docs/terraform
  work_dir: .
appwans:
- action: create
  endpoints:
  - BranchGatewayName
  - ClientName
  name: appwan-ssh-22
  services:
  - AZCPEGWx0xWESTUS--10.20.10.5--22
```

10. After the script ran again successfully, the connectivity should have been up.


YOUR APPWAN SUMMARY

Your AppWAN's details are below. Looking to make some changes?
Follow the hints below to edit your AppWAN, Endpoints, and or Services.

HINT

EDIT YOUR APPWAN


Tap the Edit icon to make changes



HINT


ENDPOINTS & SERVICES

Tap to Edit Advanced Options



1

APPWAN NAME


appwan-ssh-22 


2

ENDPOINTS

CLIENTS


●

ClientName 



GATEWAYS


●

BRANCHGATEWAYNAME 

3

SERVICES


SERVICE DEFINITIONS

AZCPEGWx0xWESTUS--10.20.10.5-22 

4


ENDPOINT GROUPS

GROUPS

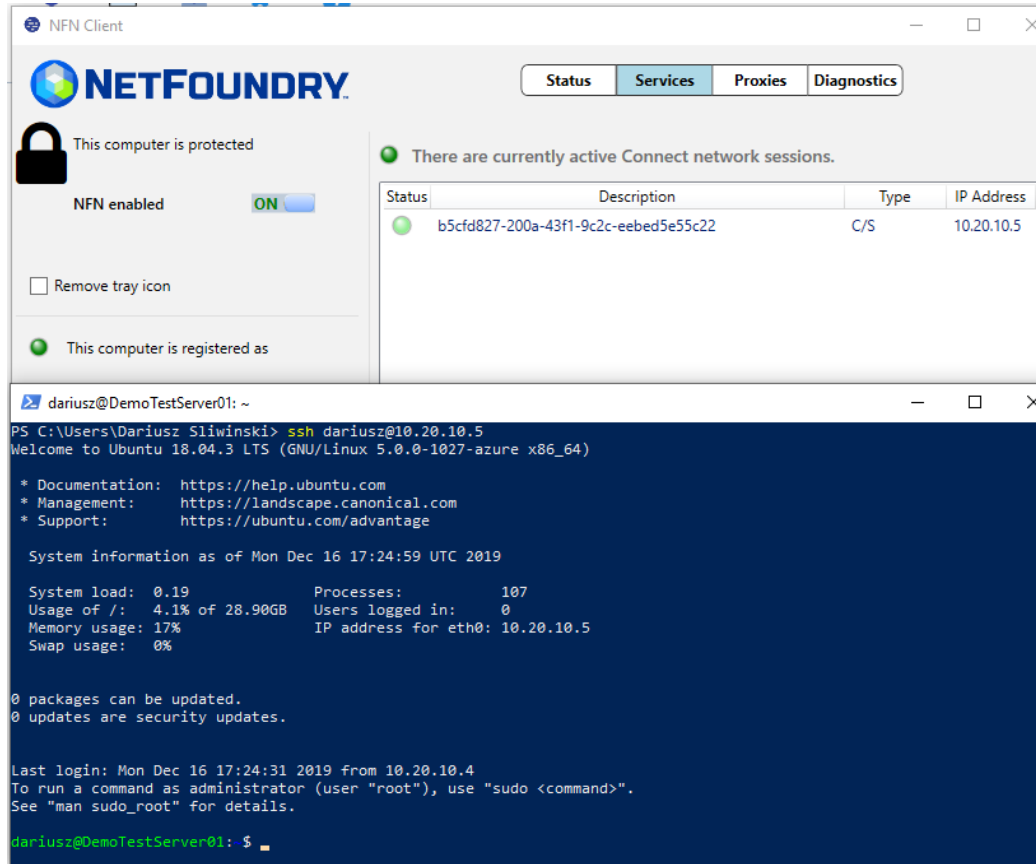


Want to add another environment
with the same services or endpoints?

TAP TO CLONE



11. To test connectivity, log in to the DemoClnet01 and run ssh "username"@privateip"



The screenshot shows the NFN Client application window and a terminal window. The NFN Client window has tabs for Status, Services, Proxies, and Diagnostics. It displays a lock icon and the text "This computer is protected" with a toggle switch for "NFN enabled" set to "ON". Below this is a checkbox for "Remove tray icon" and a green dot indicating "This computer is registered as". The "Status" tab shows a table of active Connect network sessions.

Status	Description	Type	IP Address
●	b5cfd827-200a-43f1-9c2c-eebed5e55c22	C/S	10.20.10.5

The terminal window shows the command prompt for Dariusz Sliwinski on a Windows machine, running an SSH command to connect to a server. The terminal output shows the Ubuntu 18.04.3 LTS login banner, system information, and the prompt for the user.

```
dariusz@DemoTestServer01: ~  
PS C:\Users\Dariusz Sliwinski> ssh dariusz@10.20.10.5  
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 5.0.0-1027-azure x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:       https://ubuntu.com/advantage  
  
System information as of Mon Dec 16 17:24:59 UTC 2019  
  
System load:  0.19           Processes:      107  
Usage of /:   4.1% of 28.90GB Users logged in:  0  
Memory usage: 17%          IP address for eth0: 10.20.10.5  
Swap usage:   0%  
  
0 packages can be updated.  
0 updates are security updates.  
  
Last login: Mon Dec 16 17:24:31 2019 from 10.20.10.4  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
dariusz@DemoTestServer01: $
```

12. To delete resources created, just follow the reverse order. Change the action to delete for AppWans first, then other resources as indicated in the code snippets.

```
appwans:  
- action: delete  
endpoints:  
- BranchGatewayName  
- ClientName  
name: null  
services:  
- AZCPEGWx0xWESTUS--10.20.10.5--22
```

13. Services

```
services:  
- action: delete  
gateway: AZCPEGWx0xWESTUS  
ip: 10.20.10.5  
name: null  
port: 22  
type: host
```

14. Endpoints - will delete all resources in Azure as well.

- terraform state rm "{tf resource name for RG}" // run this before the python script if Resource Group needs to be preserved 1.

```
gateway_list:
- action: delete
  cloud: azure
  count: 1
  names: []
  region: westus
  regionalCidr:
  - 10.20.10.0/24
  regkeys: []
  resourceGroup:
    name: demoPythonTerraform01
    region: centralus
  tag: TerraformDemo
```


15. Network

```
environment: production
network_action: delete
network_name: DemoNet01
```

16. Done

via Jenkins

In this section, we will use Resource yaml along with Jenkinsfile to show how to automate the steps further by creating the Jenkins Job

 **Coming Soon**