

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.

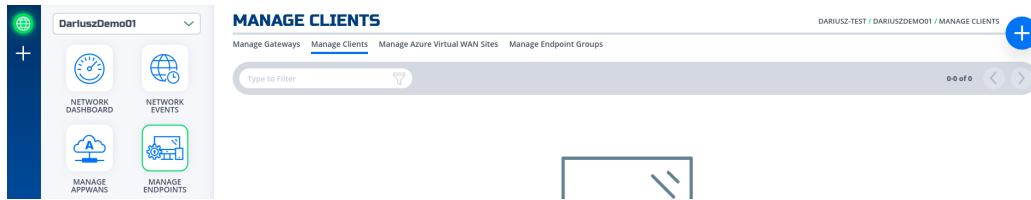
Through NF Web Console UI

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.

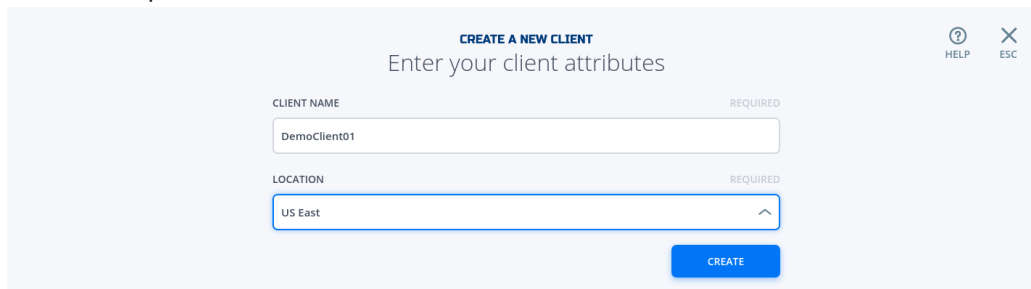


1. Navigate to Manage Clients Page

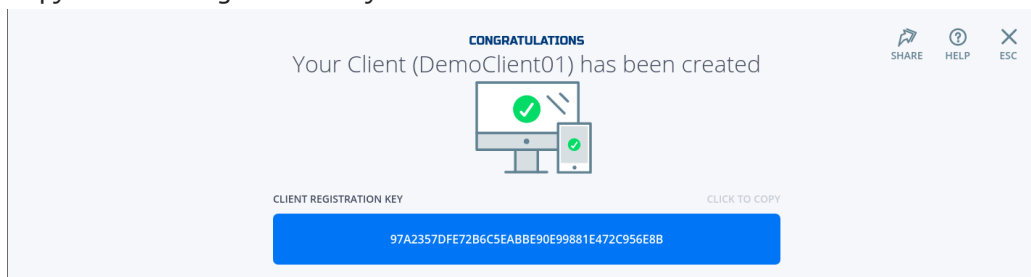


2. Click on + sign in the top right corner.

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



4. Copy the Client Registration Key



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

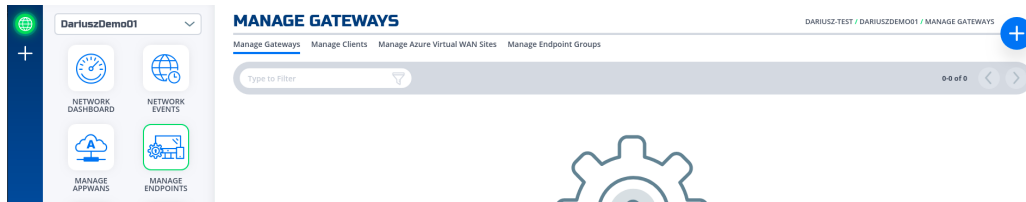
- a. Window
- b. Mac

Create and Deploy NF Azure Gateway

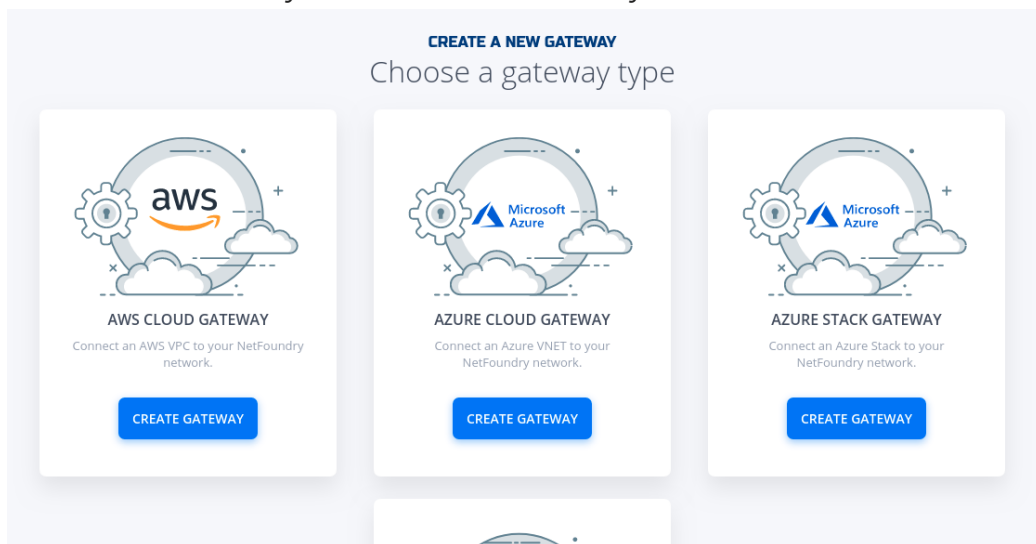
This section will guide a user through the steps on how to create a NF Manage Gateway in the NF Console UI and install it in the Azure vNet.



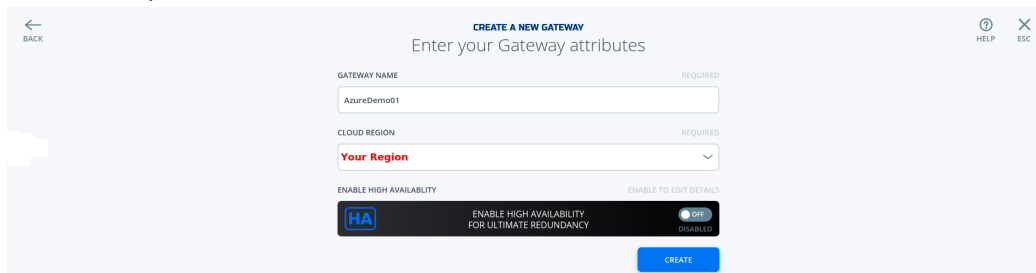
1. Navigate to Manage Gateways Page
2. Click on + sign in the top right corner.



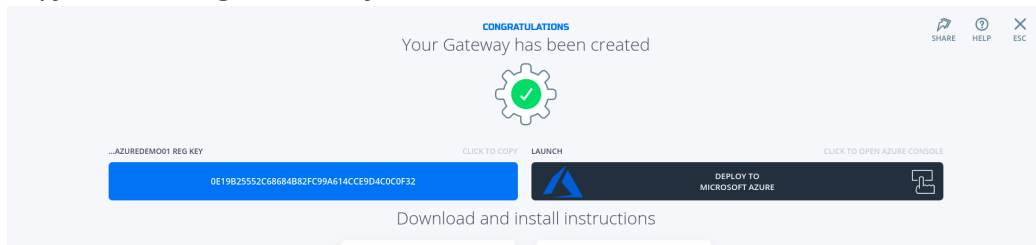
3. Click on "Create Gateway" on the Azure Cloud Gateway Card



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 *	<input type="text" value="Your Subscription Name"/>
Resource group *	<input type="text" value="Your Resource Group Name"/> Create new
Location *	<input type="text" value="(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	<input type="text" value="Your Region"/>
Network Interface Name	<input type="text" value="azuredemo01-if"/>
Security Group Name	<input type="text" value="azuredemo01-sg"/>
Virtual Network Name	<input type="text" value="azuredemo01-vnet"/>
Address Prefix	<input type="text" value="10.0.8.0/24"/>
Subnet Name	<input type="text" value="default"/>
Subnet Prefix	<input type="text" value="10.0.8.0/24"/>
Public Ip Address Name	<input type="text" value="azuredemo01-ip"/>
Public Ip Address Type	<input type="text" value="Dynamic"/>
Public Ip Address Sku	<input type="text" value="Basic"/>
Virtual Machine Name	<input type="text" value="azuredemo01"/>
Virtual Machine RG	<input type="text" value="nf-sandbox"/>
Os Disk Type	<input type="text" value="Premium_LRS"/>
Virtual Machine Size	<input type="text" value="Standard_B1ms"/>
Nfreg Key * ⓘ	<input type="text" value="....."/>
Admin Username ⓘ	<input type="text" value="nfadmin"/>
Ssh Key Data * ⓘ	<input type="text" value="ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQACjga67wcolSXaD1bswknLrejRYtZ..."/>

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 Conole UI.

The screenshot displays the Azure portal interface. The top section shows the 'nf-sandbox' resource group overview, including subscription details and a table of resources:

Name	Type
azuredemo01-if	Network interface
azuredemo01-ip	Public IP address
azuredemo01-ig	Network security group
azuredemo01-vnet	Virtual network

The bottom section shows the 'MANAGE GATEWAYS' view for 'DariuszDemo01'. It includes a filter bar and a table of gateway resources:

Gateway Label	Status	Type	Location	Cloud Provider
AzureDemo01	Online	Azure Private Gateway	Your Region	

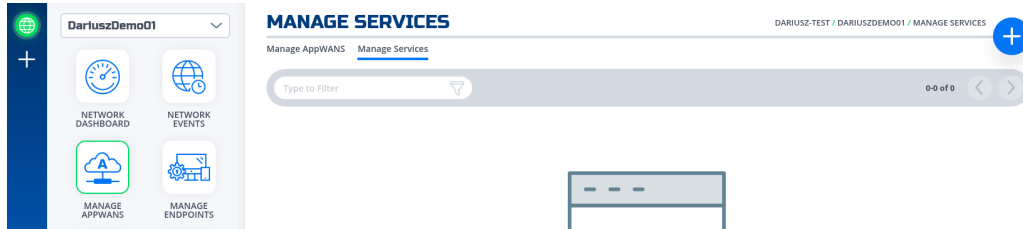
11. Done

Create IP Host Service

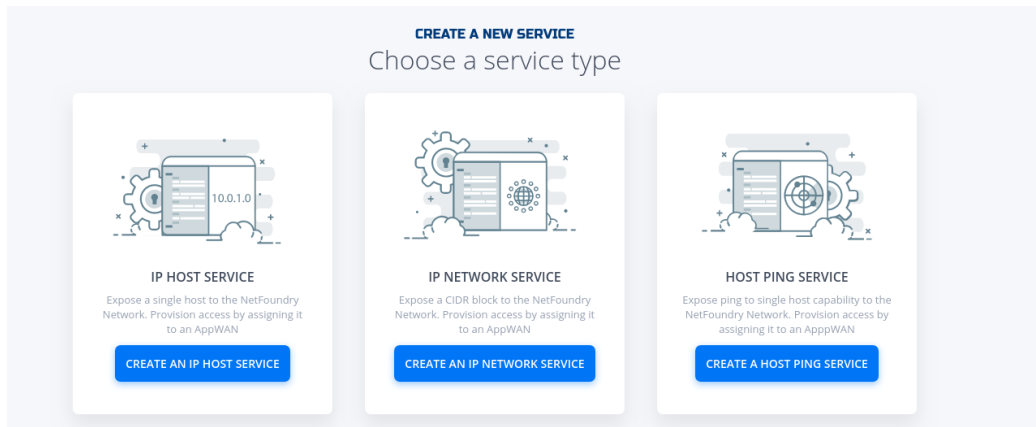
This section will guide a user through the steps on how to create a NF Service.



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"

5. If successfully, the service is green.

The screenshot shows a web interface for managing services. On the left is a navigation menu with icons for Network Dashboard, Network Events, Manage AppWans, and Manage Endpoints. The main area is titled 'MANAGE SERVICES' and includes a search bar and a table of services. The table has columns for Service Name, Type, Protocol, IP Address, Intercept IP, and Port Range. One service, 'DemoServiceSsh', is listed with a green dot next to its name, indicating it is active.

Service Name	Type	Protocol	IP Address	Intercept IP	Port Range
<input checked="" type="radio"/> DemoServiceSsh	IP Host	TCP	10.0.8.5	10.0.8.5	22 - 22

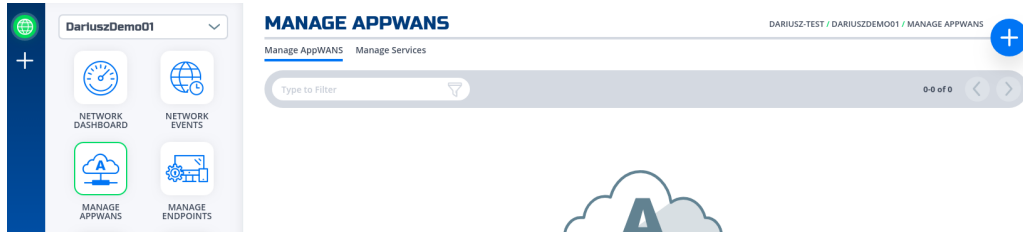
6. Done

Create AppWan

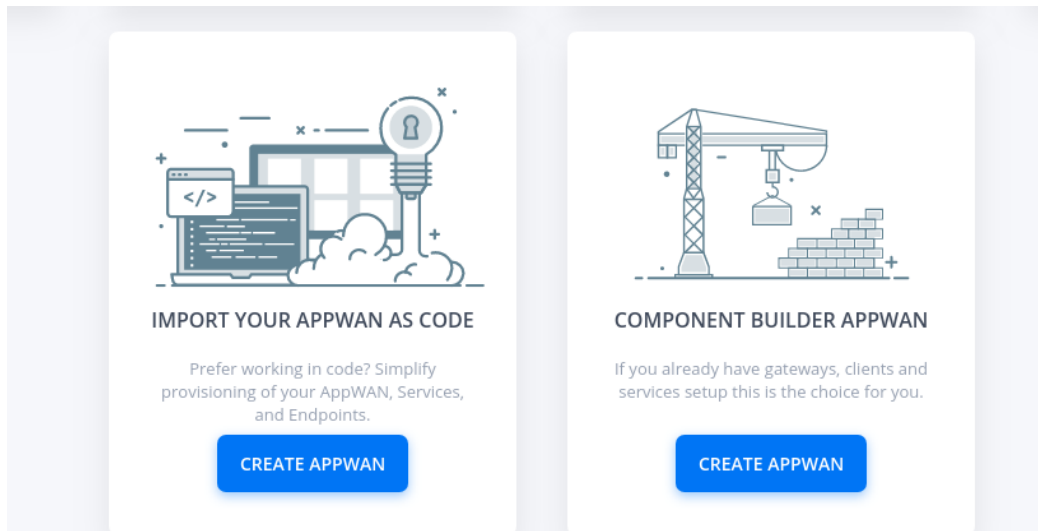
This section will guide a user through the steps on how to enable service connectivity to users by creating an appwan.



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 +

SELECTED ENDPOINTS

● DemoClient01 ✕

AVAILABLE CLIENTS ADD NEW +

←
→

AVAILABLE GATEWAYS ADD NEW +

● AzureDemo01

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.

HINT **NEW CLIENTS**
Share Client Registration Info 


HINT **NEW GATEWAYS**
Tap to Launch and Register 

- 1 APPWAN NAME**
DemoAppWan 
- 2 ENDPOINTS**
CLIENTS [SHARE NEW CLIENTS](#)
 - DemoClient01 
GATEWAYS [REGISTER NEW GATEWAYS](#)
- 3 SERVICES**
SERVICE DEFINITIONS
 - DemoServiceSsh 
- 4 ENDPOINT GROUPS**
GROUPS

 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@privatelp`

```
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 the name provided in Resource yaml, if one does not exist 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/api/python/source/netfoundry/nf_resources.py --file quickstarts/docs/api/python/etc/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 checked="" 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	<input checked="" type="radio"/> 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.

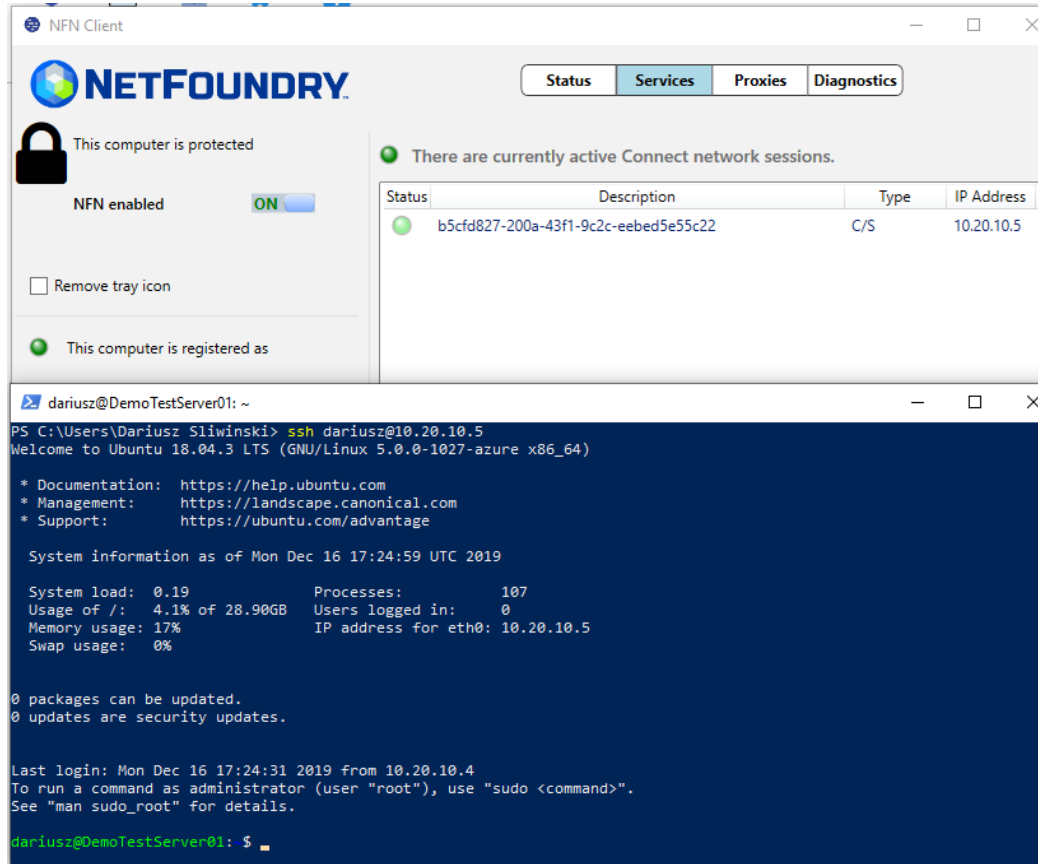
HINT EDIT YOUR APPWAN
Tap the Edit icon to make changes

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 interface with the 'Services' tab selected. It displays a table of active network sessions. Below the interface, a terminal window shows the execution of an SSH command from a Windows PowerShell prompt, resulting in a successful login to an Ubuntu 18.04.3 LTS server. The terminal output includes system information, login details, and a prompt for the user 'dariusz'.

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

```
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.

- a. 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

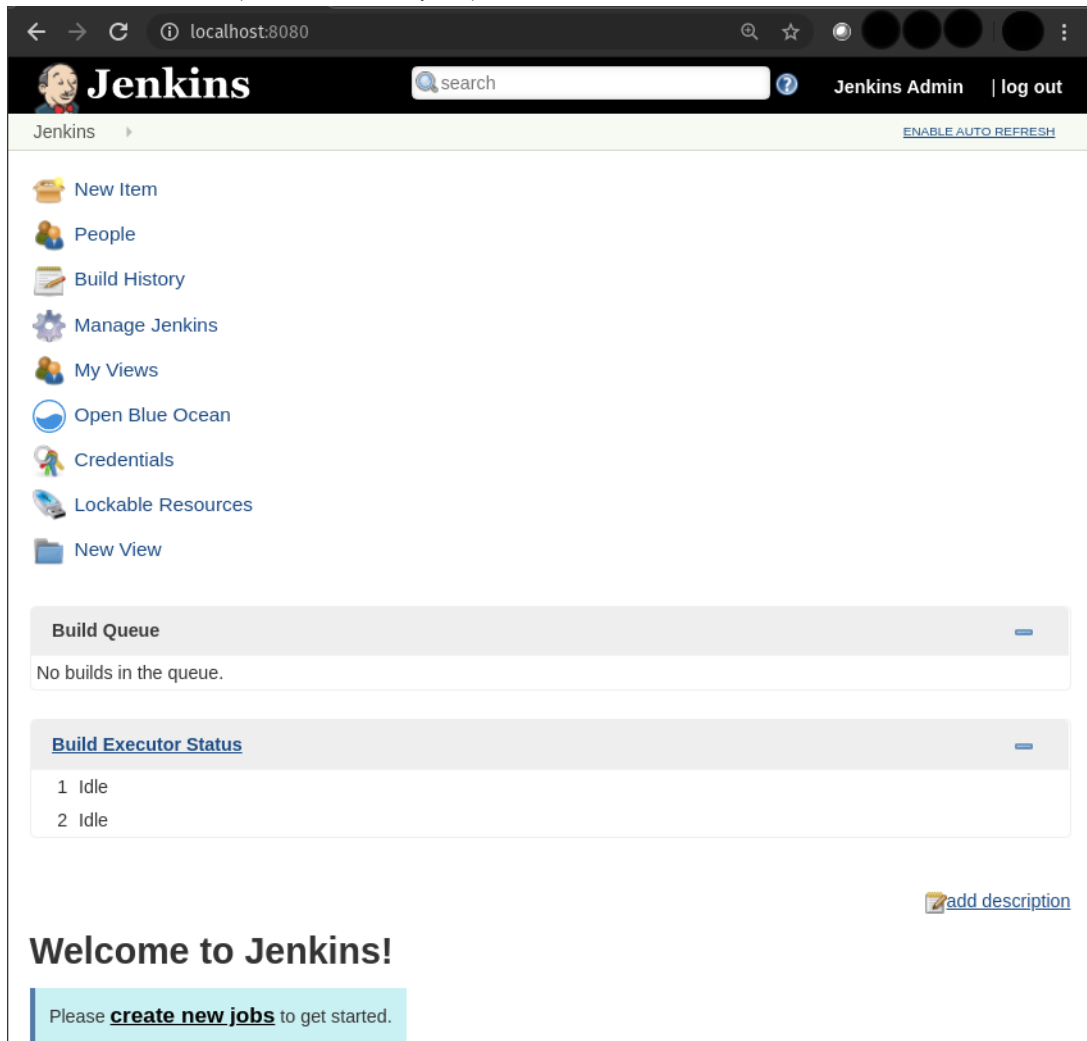
In this section, we will use Resource yaml along with Jenkinsfile to show how to automate the steps further by creating the Jenkins Job



Jenkins Requirements

1. java
2. docker

Then follow jenkins installation using docker to install Jenkins on the localhost and choose "Install suggested plugins". After successful installation, one should be able to reach the Jenkins Dashboard (8080 is default port).



The screenshot shows the Jenkins dashboard in a browser window. The address bar displays 'localhost:8080'. The page header includes the Jenkins logo, a search bar, and the user 'Jenkins Admin' with a 'log out' link. A navigation menu on the left lists various options: New Item, People, Build History, Manage Jenkins, My Views, Open Blue Ocean, Credentials, Lockable Resources, and New View. The main content area features two panels: 'Build Queue' with the message 'No builds in the queue.' and 'Build Executor Status' showing two 'Idle' executors. At the bottom, there is a 'Welcome to Jenkins!' message and a button to 'add description'. A light blue box at the bottom contains the text: 'Please **create new jobs** to get started.'



Note

If one wants to add the gateway deployed in the Private DataCenter and/or NF Client, it must be created prior to running the next steps. Otherwise the options of APPWAN_PRIVATE_GATEWAY and APPWAN_PRIVATE_CLIENT can be left blank and added after the appwan is created using the steps described in the Console UI section above. GATEWAY_NAME and SERVICE_NAME are automatically generated by the scripts in this version. GATEWAY_NAME = "GW TYPE"+x0x+"LOCATION OF AZURE GW", e.g. AZCPEGWx0xWESTUS; SERVICE_NAME = "GW NAME"--"SERVICE IP"--"SERVICE PORT", e.g. AZCPEGWx0xWESTUS--10.20.10.5--22.



Setting Up Jenkins Pipeline

1. Login to Jenkins

2.




Click on " New Item"


3. Name you Project, select pipeline option and click "Ok"

Enter an item name


» Required field

**Freestyle project**

This is the central feature of Jenkins. Jenkins will build your project, combining something other than software build.

**Pipeline**

Orchestrates long-running activities that can span multiple build agents. Suitable for organizing complex activities that do not easily fit in free-style job type.

**Multi-configuration project**

Suitable for projects that need a large number of different configurations, such as...

4. In the pipeline details, fill in the scm details as seen in the image below and click "Save".

Everything default apart from:

- Repository Url: <https://github.com/netfoundry/mop.git>

b. Script Path: pipeline/netfoundrydeploy2cloud.jenkinsfile

Pipeline

Definition: Pipeline script from SCM

SCM: Git

Repositories

Repository URL:

Credentials:

Branches to build

Branch Specifier (blank for 'any'):

Repository browser: (Auto)

Additional Behaviours:

Script Path:

Lightweight checkout:

[Pipeline Syntax](#)

5. Set up users for Azure API and NF MOP API access -- More on Credentials setup

Jenkins

Jenkins > Credentials

- New Item
- People
- Build History
- Manage Jenkins
- My Views
- Open Blue Ocean
- Lockable Resources
- Credentials**
- System
- New View

Build Queue

No builds in the queue.

Credentials

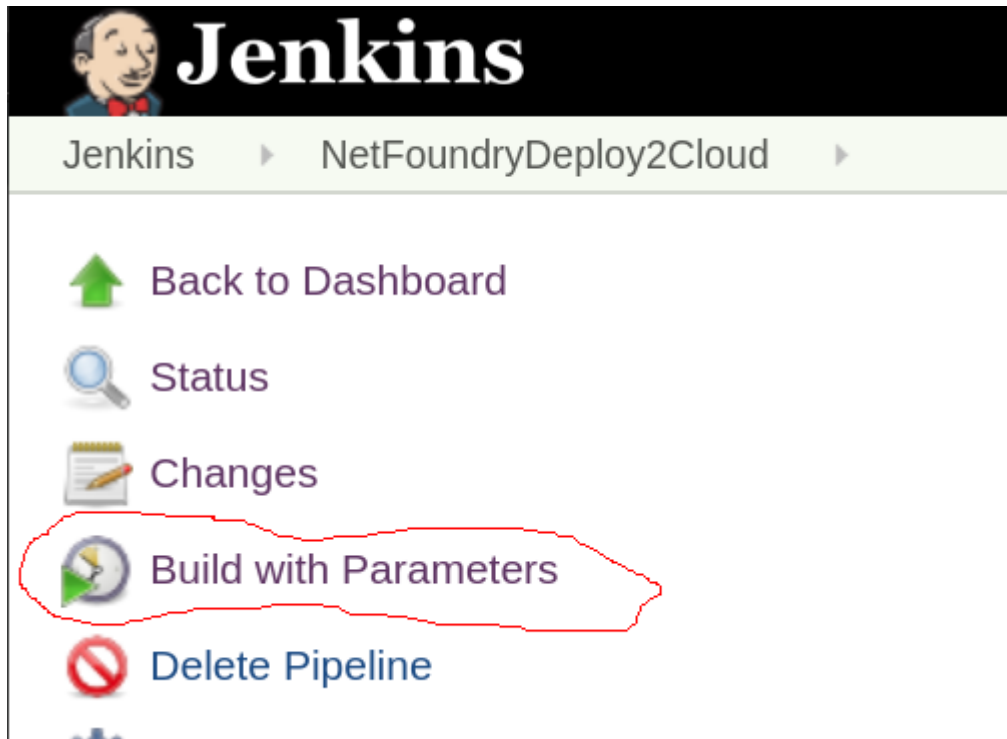
T	P	Store ↓	Domain	ID
		Jenkins	(global)	azure_user_creds 164b
		Jenkins	(global)	sandbox-mop-user Qj9K

Icon: [S](#) [M](#) [L](#)

Stores scoped to Jenkins

P	Store ↓
	Jenkins (global)

6. Run Jenkinsjob by selecting on the pipeline created in the previous step. Click on "Build with Parameters"





To create the resources

1. Fill in the Azure Details (e.g. RG, Tenant Id, etc) and select the following:
 - a. NF Environment, e.g. production
 - b. NETWORK_ACTION - create
 - c. NETWORK_NAME, e.g. DEMONET
 - d. GATEWAY_ACTION - create
 - e. If Azure RG needs to be preserved, then KEEP_RG option must be left checked.
 - f. LOCATION, e.g. westus - location where the Azure GW will be deployed in
 - g. SUBNET_PREFIX, e.g. 10.20.10.0/24 - the subnet used for the vNet in the location of the Azure GW deployment.

localhost:8080/job/NetFoundryDeploy2Cloud/build?delay=0sec

Jenkins > NetFoundryDeploy2Cloud >

- Build with Parameters
- Delete Pipeline
- Configure
- Full Stage View
- Open Blue Ocean
- Rename
- Pipeline Syntax

Build History trend

Atom feed for all Atom feed for failures

AZURE_TENANT_ID	[REDACTED]
AZURE_SUBSCRIPTION_ID	[REDACTED]
RESOURCE_GROUP_NAME	[REDACTED]
RESOURCE_GROUP_LOC	centralus
ENVIRONMENT	sandbox
NETWORK_ACTION	create
NETWORK_NAME	DEMONET
GATEWAY_ACTION	create
GATEWAY_NAME	
SERVICE_ACTION	get
SERVICE_NAME	
SERVICE_IP	
SERVICE_PORT	
APPWAN_ACTION	get
APPWAN_NAME	
APPWAN_PRIVATE_GATEWAY	
APPWAN_PRIVATE_CLIENT	
APPWAN_SERVICE	
LOCATION	westus
SUBNET_PREFIX	10.20.10.0/24

Build

2. Run Jenkins job again by selecting on the pipeline created in the previous step. Click on "Build with Parameters"

3. Fill in service and appwan details by selecting the following:

- a. KEEP_RG - not selected
- b. NF Environment, e.g. production
- c. SERVICE_ACTION - create
- d. APPWAN_ACTION - create
- e. GATEWAY_NAME, e.g. AZCPEGWx0xWESTUS (this is created in the previous step automatically)
- f. SERVICE_NAME, e.g. AZCPEGWx0xWESTUS--10.20.10.5--22 (this is created automatically during this step)
- g. SERVICE_IP, e.g. 10.20.10.5
- h. SERVICE_PORT, e.g. 22
- i. APPWAN_NAME, e.g. appwan-ssh-22
- j. APPWAN_PRIVATE_GATEWAY, e.g. private-gateway-name (this is created outside of the jenkins job, prior to running this step)
- k. APPWAN_PRIVATE_CLIENT, e.g. client-name (this is created outside of the jenkins job, prior to running this step)

I. APPWAN_SERVICE, e.g. AZCPEGWx0xWESTUS--10.20.10.5--22

Jenkins x +

localhost:8080/job/NetFoundryDeploy2Cloud/build?delay=0sec

Jenkins > NetFoundryDeploy2Cloud >

- Build with Parameters
- Delete Pipeline
- Configure
- Full Stage View
- Open Blue Ocean
- Rename
- Pipeline Syntax

Build History trend

Atom feed for all Atom feed for failures

AZURE_TENANT_ID	<input type="text"/>	Tenant ID in Azure
AZURE_SUBSCRIPTION_ID	<input type="text"/>	Subscription ID in Azure
RESOURCE_GROUP_NAME	<input type="text"/>	RG Name in Azure
RESOURCE_GROUP_LOC	<input type="text"/>	RG Location in Azure
ENVIRONMENT	<input type="text" value="sandbox"/>	<input type="checkbox"/> KEEP_RG Not to check this if RG can be deleted Select NF Console Environment to spin the network and gateways in
NETWORK_ACTION	<input type="text" value="get"/>	Selection an action to perform on the network in NF
NETWORK_NAME	<input type="text" value="DEMONET"/>	Name to be used to create a network with
GATEWAY_ACTION	<input type="text" value="get"/>	Selection an action to perform on the gateway in NF Network
GATEWAY_NAME	<input type="text" value="AZCPEGWx0xWESTUS"/>	Name of NF Gateway generated in NF Console
SERVICE_ACTION	<input type="text" value="create"/>	Selection an action to perform on the service in NF Network
SERVICE_NAME	<input type="text" value="AZCPEGWx0xWESTUS--10.20.10.5--22"/>	Name of NF Service generated in NF Console
SERVICE_IP	<input type="text" value="10.20.10.5"/>	IP of NF Service App
SERVICE_PORT	<input type="text" value="22"/>	IP of NF Service App
APPWAN_ACTION	<input type="text" value="create"/>	Selection an action to perform on the appwan in NF Network
APPWAN_NAME	<input type="text" value="appwan-ssh-22"/>	Name of NF APPWAN to be used in NF Console
APPWAN_PRIVATE_GATEWAY	<input type="text" value="private-gateway-name"/>	Endpoint Name in Private Datacenter Gateway to be included in AppWan
APPWAN_PRIVATE_CLIENT	<input type="text" value="client-name"/>	Endpoint Name for Client to be included in AppWan
APPWAN_SERVICE	<input type="text" value="AZCPEGWx0xWESTUS--10.20.10.5--22"/>	Service Name to be included in AppWan
LOCATION	<input type="text" value="westus"/>	Azure Cloud DC Location where to deploy GW
SUBNET_PREFIX	<input type="text" value="10.20.10.0/24"/>	Subnet CIDR in Azure Cloud DC Location where to deploy GW

Build



To delete the resources

1. Run Jenkins job again by selecting on the pipeline created in the previous step. Click on "Build with Parameters"
2. Fill in the Azure Details (e.g. RG, Tenant Id, etc) and select the following:
 - a. NF Environment, e.g. production
 - b. NETWORK_ACTION - delete
 - c. NETWORK_NAME, e.g. DEMONET
 - d. GATEWAY_ACTION - delete

Pipeline View

Jenkins

Jenkins > NetFoundryDeploy2Cloud >

- [Back to Dashboard](#)
- [Status](#)
- [Changes](#)
- [Build with Parameters](#)
- [Delete Pipeline](#)
- [Configure](#)
- [Full Stage View](#)
- [Open Blue Ocean](#)
- [Rename](#)
- [Pipeline Syntax](#)

Build History [trend](#)

[Atom feed for all](#) [Atom feed for failures](#)

Pipeline NetFoundryDeploy2Cloud

This build requires parameters:

AZURE_TENANT_ID	<input type="text" value="REDACTED"/>
	<small>Tenant ID in Azure</small>
AZURE_SUBSCRIPTION_ID	<input type="text" value="REDACTED"/>
	<small>Subscription ID in Azure</small>
RESOURCE_GROUP_NAME	<input type="text" value="REDACTED"/>
	<small>RG Name in Azure</small>
RESOURCE_GROUP_LOC	<input type="text" value="centralus"/>
	<small>RG Location in Azure</small>
	<input type="checkbox"/> KEEP_RG
	<small>Not to check this if RG can be deleted</small>
ENVIRONMENT	<input type="text" value="sandbox"/>
	<small>Select NF Console Environment to spin the network and gateways in</small>
NETWORK_ACTION	<input type="text" value="delete"/>
	<small>Select an action to perform on the network in NF</small>
NETWORK_NAME	<input type="text" value="DEMONET"/>
	<small>Name to be used to create a network with</small>
GATEWAY_ACTION	<input type="text" value="delete"/>
	<small>Select an action to perform on the gateway in NF Network</small>
GATEWAY_NAME	<input type="text"/>
	<small>Name of NF Gateway generated in NF Console</small>
SERVICE_ACTION	<input type="text" value="get"/>
	<small>Select an action to perform on the service in NF Network</small>
SERVICE_NAME	<input type="text"/>
	<small>Name of NF Service generated in NF Console</small>
SERVICE_IP	<input type="text"/>
	<small>IP of NF Service App</small>
SERVICE_PORT	<input type="text"/>
	<small>IP of NF Service App</small>
APPWAN_ACTION	<input type="text" value="get"/>
	<small>Select an action to perform on the appwan in NF Network</small>
APPWAN_NAME	<input type="text"/>
	<small>Name of NF APPWAN to be used in NF Console</small>
APPWAN_PRIVATE_GATEWAY	<input type="text"/>
	<small>Endpoint Name in Private Datacenter Gateway to be included in AppWan</small>
APPWAN_PRIVATE_CLIENT	<input type="text"/>
	<small>Endpoint Name for Client to be included in AppWan</small>
APPWAN_SERVICE	<input type="text"/>
	<small>Service Name to be included in AppWan</small>
LOCATION	<input type="text" value="westus"/>
	<small>Azure Cloud DC Location where to deploy GW</small>
SUBNET_PREFIX	<input type="text" value="10.20.10.0/24"/>
	<small>Subnet CIDR in Azure Cloud DC Location where to deploy GW</small>

3. Done