

# Branch Application To Cloud Application Connectivity

## Overview

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

### Important

Assumption is that the NF Fabric is already up.

## Through NF Web Console UI

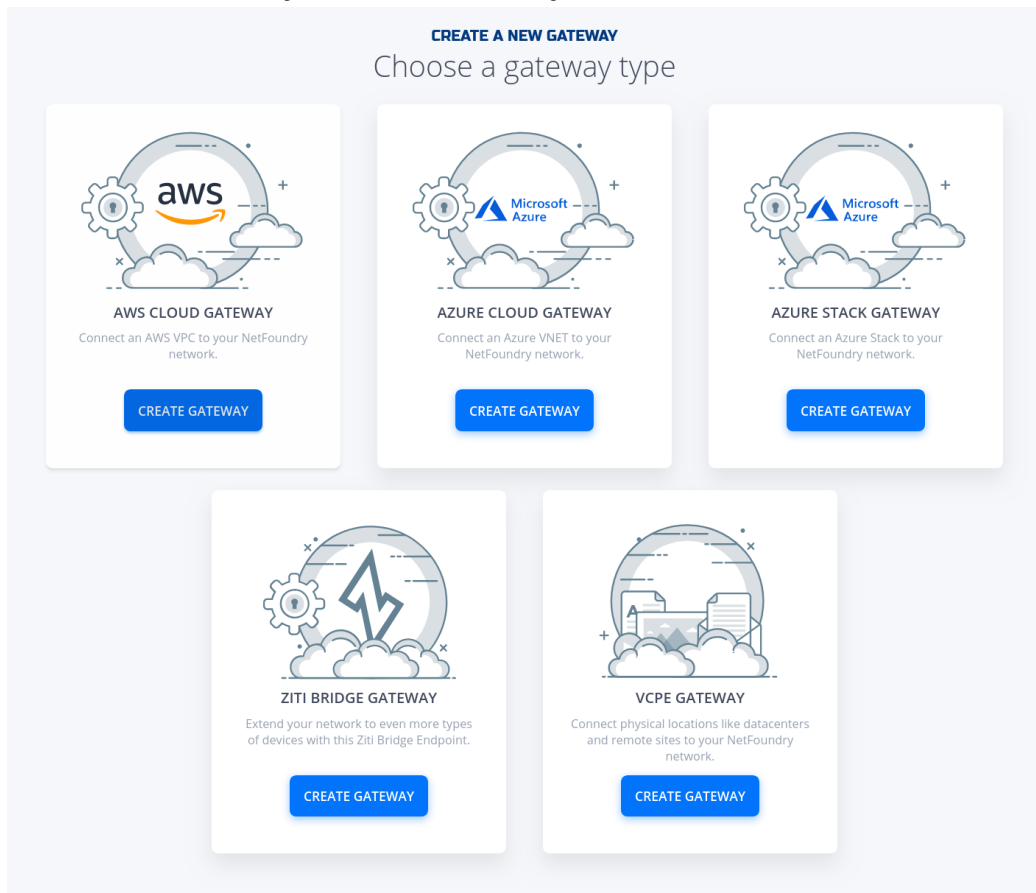
### **Create and Deploy NF Gateway in Branch Datacenter**

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 Branch Datacenter.

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



3. Click on "Create Gateway" on the VCPE Gateway Card



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

### CREATE A NEW GATEWAY

## Enter your Gateway attributes

GATEWAY NAME

REQUIRED

YourBranchGatewayName

CLOUD REGION

REQUIRED

US East

ENABLE HIGH AVAILABILITY

ENABLE TO EDIT DETAILS

HA

ENABLE HIGH AVAILABILITY  
FOR ULTIMATE REDUNDANCY

OFF


DISABLED

CREATE

5. Copy the Client Registration Key

### CONGRATULATIONS

## Your Gateway has been created

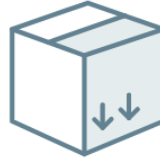


YOURBRANCHGATEWAYNAME REGISTRATION KEY

CLICK TO COPY

FB18C6EBEAD65B723E9C86B6882A3298ED77DBBF


## Download and install instructions



### DOWNLOAD INSTALL PACKAGE

vCPE Gateway to connect to private networks, branches, & data centers using this customer-managed virtual gateway.

DOWNLOAD



### VIEW INSTALL INSTRUCTIONS

Connect public or private IP space inside your vCPE using a customer-managed cloud gateway.

VIEW

## 6. Click on "Download" button on the Installation Package Card

**NETFOUNDRY** SOLUTIONS - PARTNERS RESOURCES - ABOUT - [START NOW](#)

**AWS Cloud Gateway**  
Deploy secure, high performance, global networks as easily and effectively as you spin up your AWS instances. Download it now from the AWS Marketplace. Not sure how to install it? [Click Here](#) to read our simple install instructions.

[VISIT MARKETPLACE](#)

**Azure Cloud Gateway**  
Improve the agility, security and performance of your cloud applications in Azure. Download it now from the Azure Marketplace. Not sure how to install it? [Click Here](#) to read our simple install instructions.

[VISIT MARKETPLACE](#)

**OVA for VMWare ESXi 5.0 or greater**  
This is a V-CPE Gateway to connect to private networks, branches, & datacenters using this customer-managed virtual gateway. Not sure how to install it? [Click Here](#) to read our simple install instructions.

[DOWNLOAD](#)

**Microsoft Hyper-V VHD Image**  
This is a V-CPE Gateway to connect to private networks, branches, & datacenters using this customer-managed virtual gateway. Not sure how to install it? [Click Here](#) to read our simple install instructions.

**KVM QCOW2 Image**  
This is a V-CPE Gateway to connect to private networks, branches, & datacenters using this customer-managed virtual gateway. Not sure how to install it? [Click Here](#) to read our simple install instructions.

**OVA for Virtual Box**  
This is a V-CPE Gateway to connect to private networks, branches, & datacenters using this customer-managed virtual gateway. Not sure how to install it? [Click Here](#) to read our simple install instructions.

## 7. Download the correct image for the desired Hypervisor.

## 8. Follow the installation procedure linked in the description of each image type (i.e. "Click Here").

## 9. Once installed, login into it locally with ssh and register it using the key copied in the previous step. Run the following command `sudo nfnreg "reg key"`

```
infadmin@yourBranchGatewayName ~]$ sudo nfnreg 36BFD32C7F09883BA4C0C3D991D3BB013D89EA87
Created symlink from /etc/systemd/system/multi-user.target.wants/dvn-driver.service to /usr/lib/systemd/system/dvn-driver.service.
Created symlink from /etc/systemd/system/multi-user.target.wants/dvn.service to /usr/lib/systemd/system/dvn.service.
Registration complete
infadmin@yourBranchGatewayName ~]$
```

## 10. Once registered, one should see the gateway status turn to green in NF Console UI

### MANAGE GATEWAYS

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

| Type to Filter                              |   |  |          |
|---|---|--|----------|
| Gateway Label                               | Status                                      | Type   | Location |
| <input type="radio"/> YourBranchGatewayName | <span style="color: green;">●</span> Online | <span style="border: 1px solid blue; padding: 2px;">S</span> V-CPE Gateway | US East  |

## 11. Done

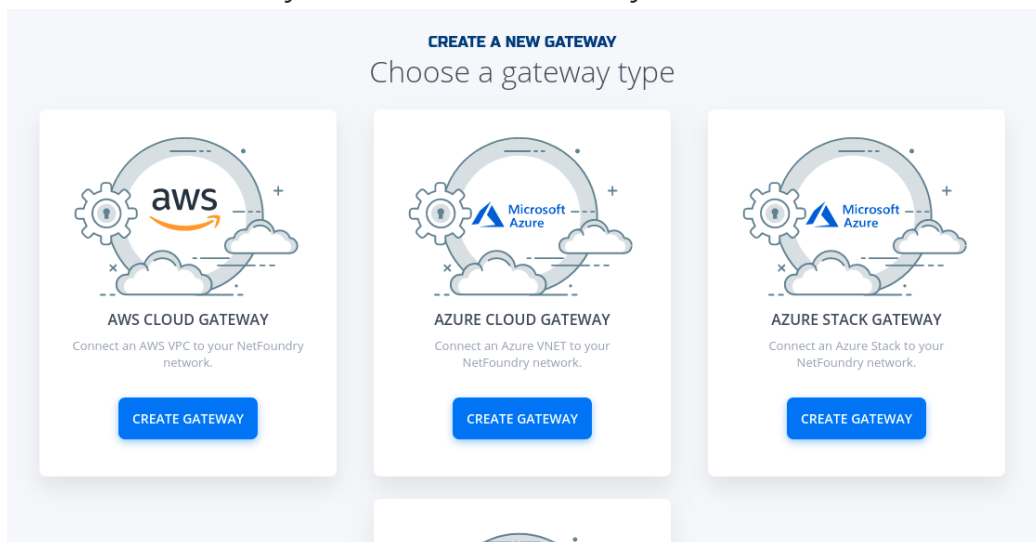
## **Create and Deploy NF Azure Gateway**

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.

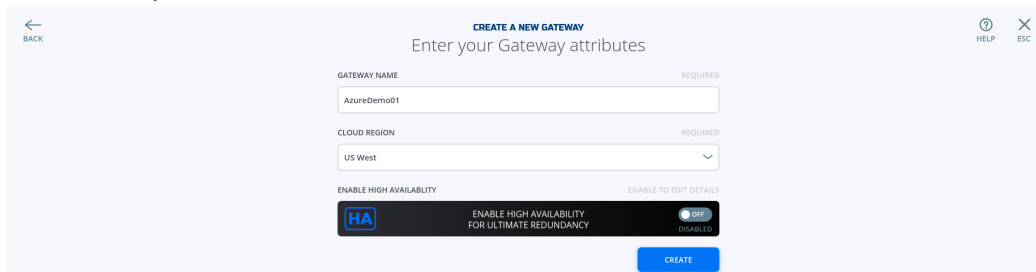
1. Navigate to Manage Gateways Page
2. Click on + sign in the top right corner.



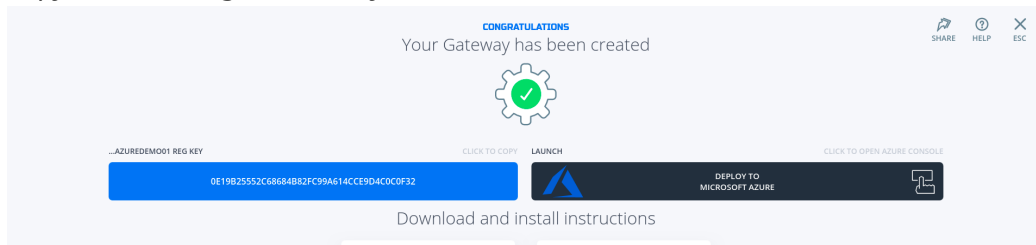
3. Click on "Create Gateway" on the Azure Cloud Gateway Card



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                 | ▼ |
|                  | <a href="#">Create new</a> |   |
| 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. The top section shows the 'nf-sandbox' resource group overview, including subscription details and a list of resources: azuredemo01-if (Network interface), azuredemo01-ip (Public IP address), azuredemo01-sg (Network security group), and azuredemo01-vnet (Virtual network). The bottom section shows the 'MANAGE GATEWAYS' console for 'DariuszDemo01', displaying a table with one gateway: 'AzureDemo01' (Online, Azure Private Gateway, Oregon, Cloud Provider). A notification on the right indicates a successful deployment of 'Microsoft.Template' to the resource group.

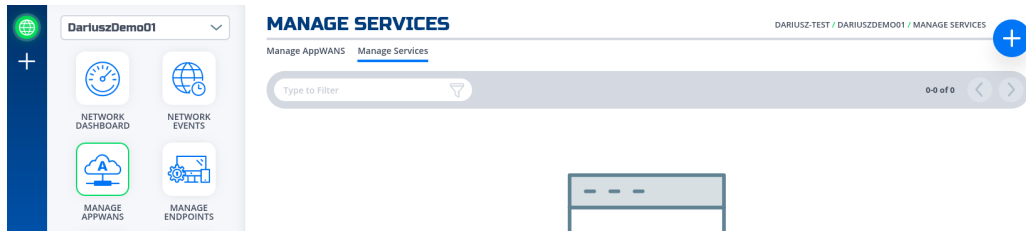
11. Done

## Create IP Host Service

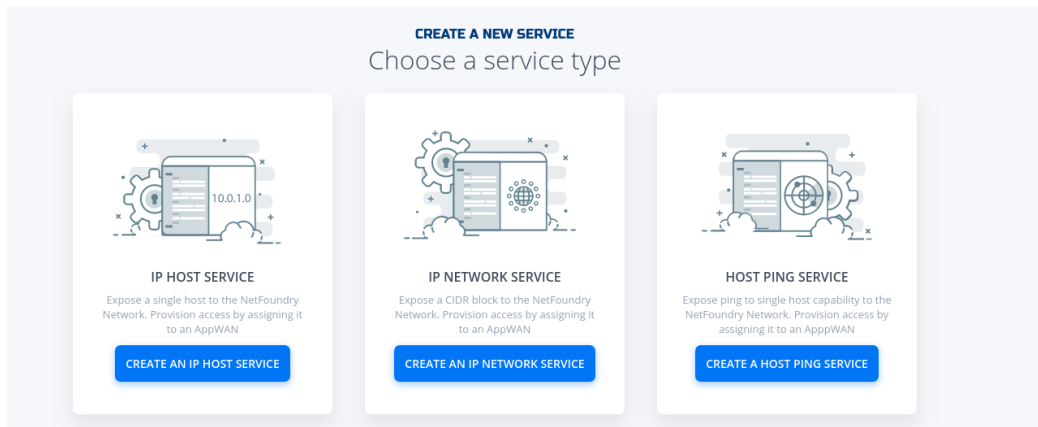
This section will guide a user through the steps on how to create a NF Service.



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 the 'MANAGE SERVICES' interface. On the left is a sidebar with a '+', a globe icon, and a dropdown menu set to 'Demo01'. Below the dropdown are four icons: 'NETWORK DASHBOARD', 'NETWORK EVENTS', 'MANAGE APPWANS', and 'MANAGE ENDPOINTS'. The main panel has a header with 'MANAGE SERVICES' and a breadcrumb trail 'Manage AppWANS / Manage Services'. Below the header is a search bar 'Type to Filter' and a table. The table has columns: 'Service Name', 'Type', 'Protocol', 'IP Address', 'Intercept IP', and 'Port Range'. There is one row with a green dot in the first column, 'DemoServiceSsh' in the second, 'IP Host' in the third, 'TCP' in the fourth, '10.0.8.5' in the fifth, '10.0.8.5' in the sixth, and '22 - 22' in the seventh. A '1-1 of 1' indicator is on the right of the table.

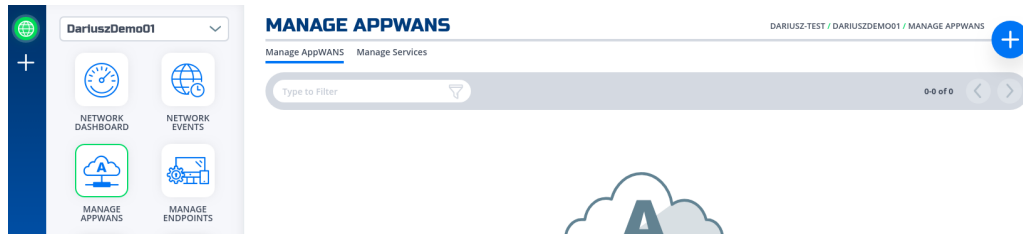
| Service Name     | Type    | Protocol | IP Address | Intercept IP | Port Range |
|------------------|---------|----------|------------|--------------|------------|
| ● DemoServiceSsh | IP Host | TCP      | 10.0.8.5   | 10.0.8.5     | 22 - 22    |

6. Done

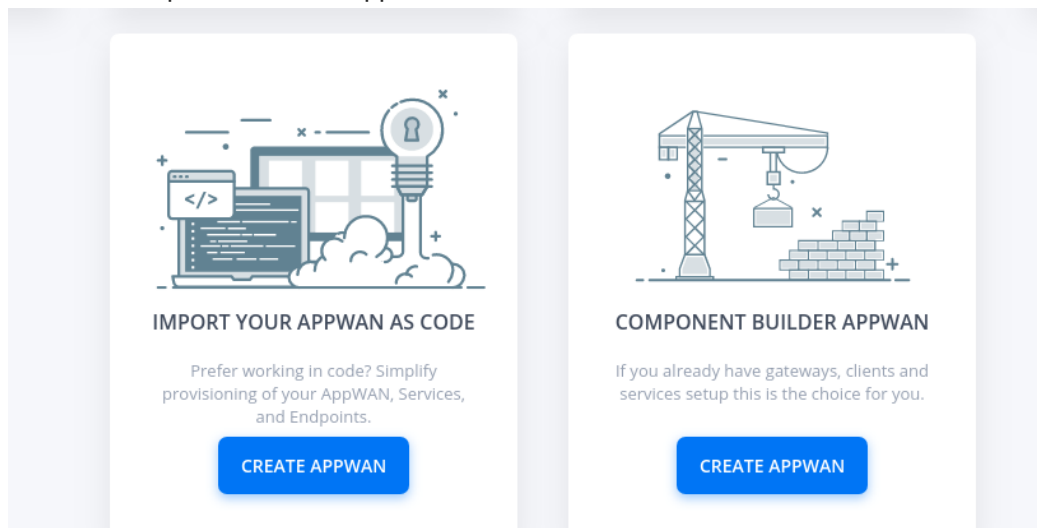
## Create AppWan

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 gateway (e.g. DemoGateway01) from "Available" Gateways 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

YourBranchGatewayName

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  
GATEWAYS REGISTER NEW GATEWAYS  
● YOURBRANCHGATEWAYNAME

3 SERVICES  
SERVICE DEFINITIONS  
● DemoServiceSsh

4 ENDPOINT GROUPS  
GROUPS

6. Done

## Test Connectivity to Application Server

### Route to vNet

The private IP of NF Gateway (e.g. YourBranchGatewayName) needs to be the next hop to reach the vNet in Azure (e.g. 10.0.8.0/24). Thus, a static route will need to be configured in one of the branch routers. NF Gateway can support dynamic routing if needed (e.g.bgp, ospf)

## Steps

1. Log in to a Client App Host in Branch DataCenter
2. Run `ssh username@privateIpOfServerAppHostInAzure`

```
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 < >

| <input type="radio"/> Service Name                               | <input type="checkbox"/> 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 gateway in the branch as the steps in the UI section indicated. We will provide code snippets for private hypervisors deployment through python in later releases (e.g. vSphere)

## MANAGE GATEWAYS

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

Type to Filter 1-1 of 1 < >

| <input type="radio"/> Gateway Label         | <input type="checkbox"/> Status         | Type                         | Location |
|---|---|------------------------------|----------|
| <input type="radio"/> YourBranchGatewayName | <input checked="" type="radio"/> Online | <span>S</span> V-CPE Gateway | US East  |

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

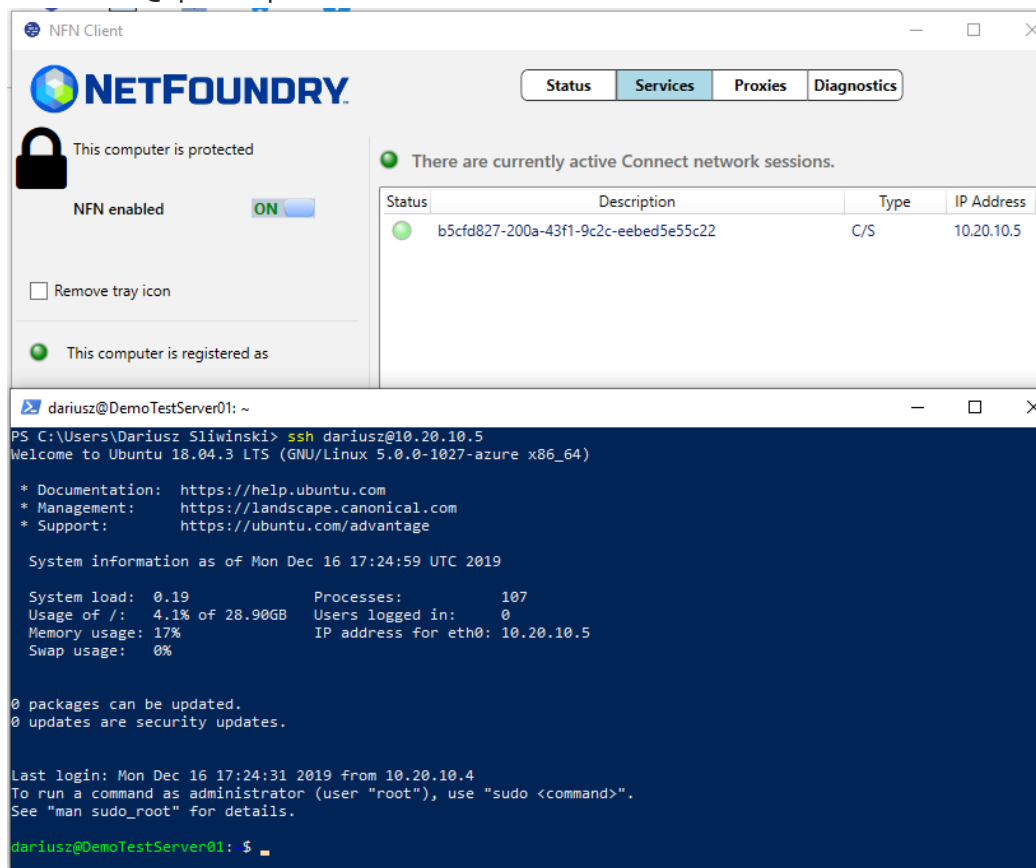


Want to add another environment  
with the same services or endpoints?

TAP TO CLONE



11. To test connectivity, log in to the Remote Client or Branch App Server and run ssh "username"@ "privatelp"



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

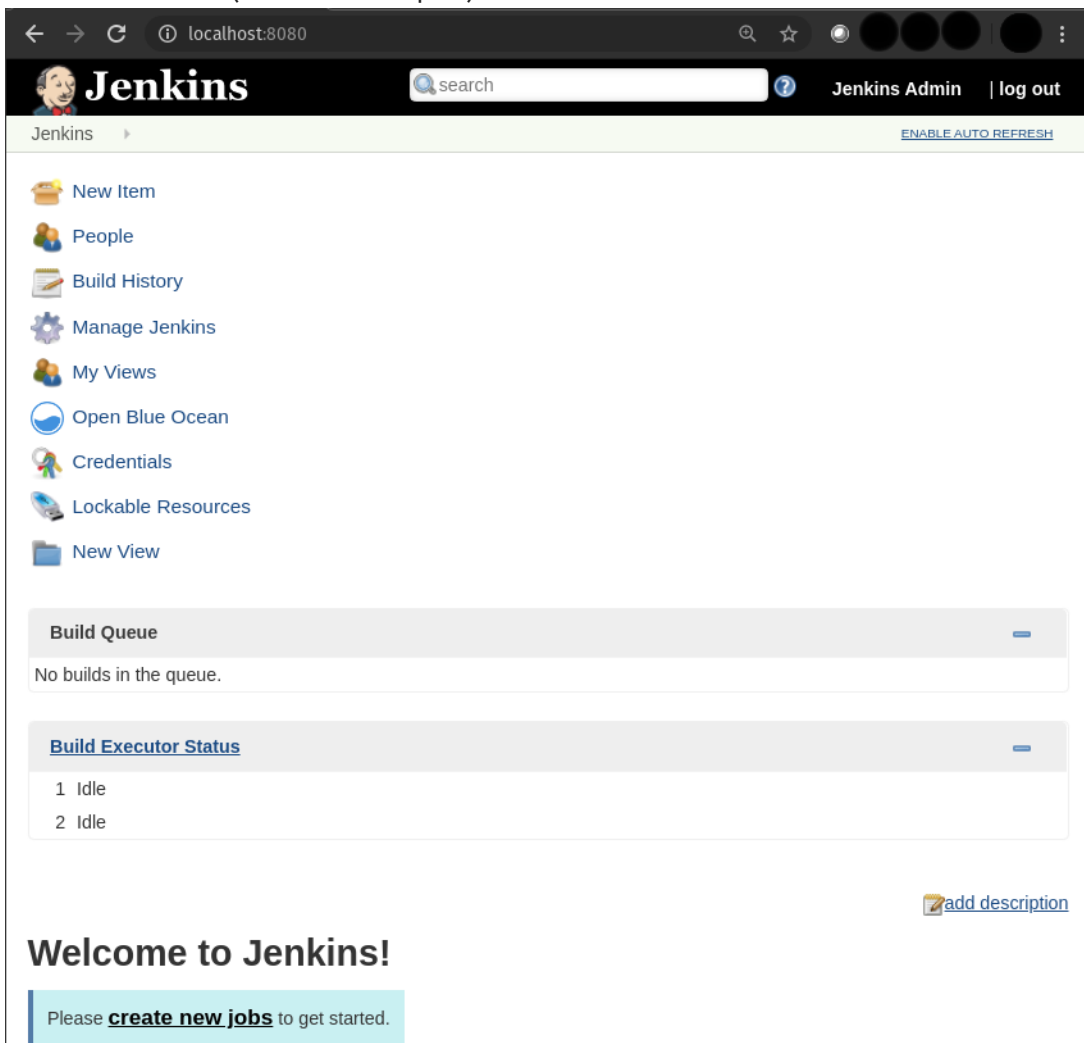
### via Jenkins

In this section, we will use `Resource yml` 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 web browser. The browser's address bar displays 'localhost:8080'. The Jenkins header includes the logo, a search bar, and links for 'Jenkins Admin' and 'log out'. A sidebar on the left contains links to 'New Item', 'People', 'Build History', 'Manage Jenkins', 'My Views', 'Open Blue Ocean', 'Credentials', 'Lockable Resources', and 'New View'. The main content area features two expandable sections: 'Build Queue' (showing 'No builds in the queue.') and 'Build Executor Status' (showing two 'Idle' executors). At the bottom right, there is a link to 'add description'. A large 'Welcome to Jenkins!' message is at the bottom, with a light blue box containing the text 'Please **create new jobs** to get started.'

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

[add description](#)

## Welcome to Jenkins!

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

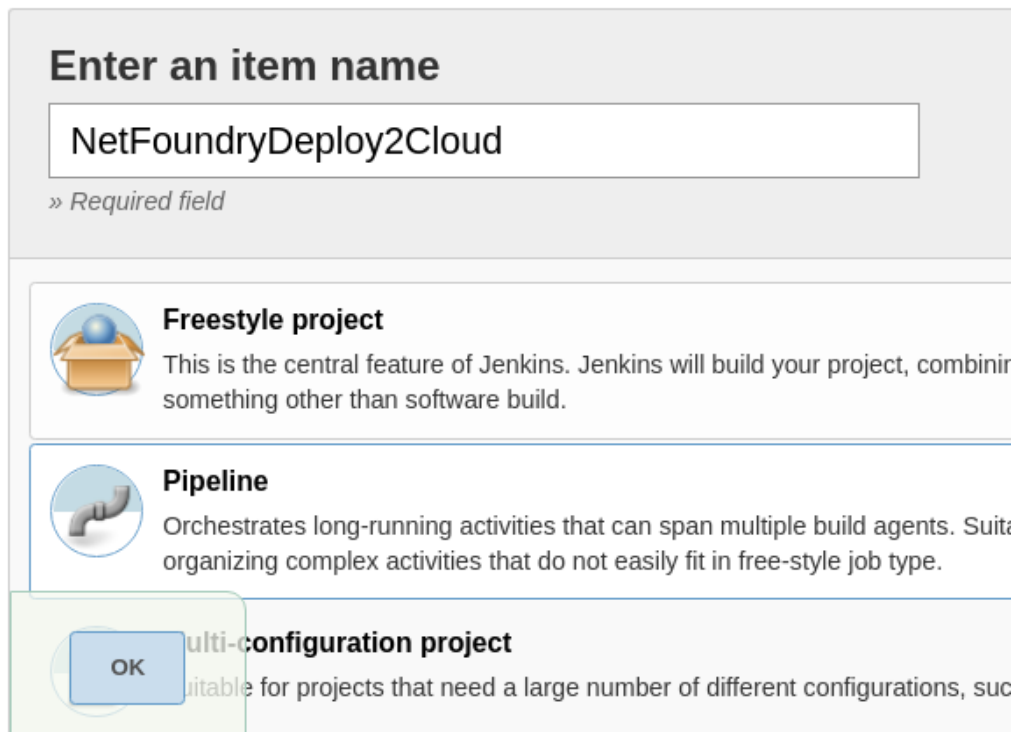
1. Login to Jenkins

2.



Click on "New Item"

3. Name your Project, select pipeline option and click "OK"

A screenshot of the Jenkins "Enter an item name" form. The form has a title "Enter an item name" in bold. Below the title is a text input field containing the text "NetFoundryDeploy2Cloud". Below the input field is a small text label "» Required field". Below the input field are three options, each with an icon and a description. The first option is "Freestyle project" with a box icon and the description "This is the central feature of Jenkins. Jenkins will build your project, combining something other than software build." The second option is "Pipeline" with a pipe icon and the description "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." The third option is "Multi-configuration project" with a box icon and the description "Suitable for projects that need a large number of different configurations, such as...". The "Multi-configuration project" option is highlighted with a green border. An "OK" button is visible in the bottom left corner of the form.

4. In the pipeline details, fill in the scm details as seen in the image below and click "Save". Everything default apart from:

a. 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:  [Add](#)

[Advanced...](#)

[Add Repository](#)

Branches to build

Branch Specifier (blank for 'any'):

[Add Branch](#)

Repository browser: (Auto)

Additional Behaviours: [Add](#)


Script Path:

Lightweight checkout: ☒

[Pipeline Syntax](#)

[Save](#) [Apply](#)

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                                    |
|---|---|-------------------------|----------|---------------------------------------|
|   |   | <a href="#">Jenkins</a> | (global) | azure_user_creds <a href="#">164b</a> |
|   |   | <a href="#">Jenkins</a> | (global) | sandbox-mop-user <a href="#">QJ9K</a> |

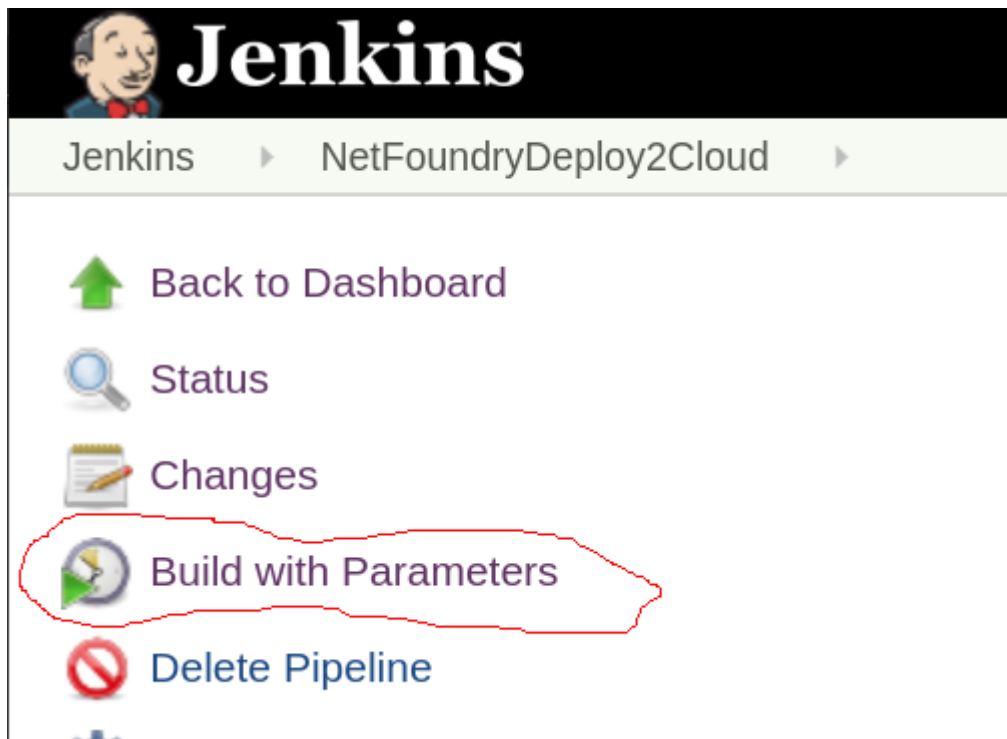
Icon: [S](#) [M](#) [L](#)

### Stores scoped to [Jenkins](#)

| P | Store                            |
|---|----------------------------------|
|   | <a href="#">Jenkins</a> (global) |

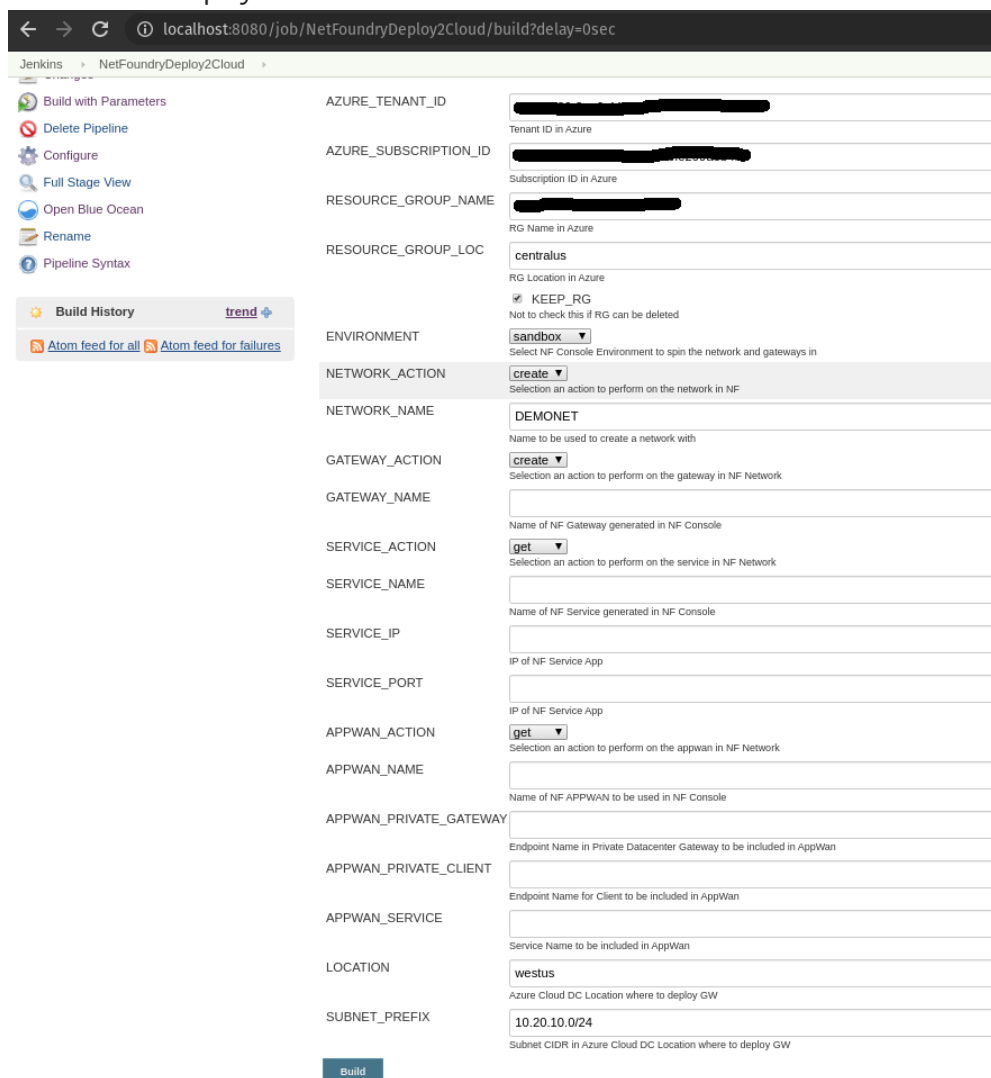
[Add credentials](#)

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

Tenant ID in Azure

AZURE\_SUBSCRIPTION\_ID

Subscription ID in Azure

RESOURCE\_GROUP\_NAME

RG Name in Azure

RESOURCE\_GROUP\_LOC

RG Location in Azure

☒ KEEP\_RG  
Not to check this if RG can be deleted

ENVIRONMENT

Select NF Console Environment to spin the network and gateways in

NETWORK\_ACTION

Select an action to perform on the network in NF

NETWORK\_NAME

Name to be used to create a network with

GATEWAY\_ACTION

Select an action to perform on the gateway in NF Network

GATEWAY\_NAME

Name of NF Gateway generated in NF Console

SERVICE\_ACTION

Select an action to perform on the service in NF Network

SERVICE\_NAME

Name of NF Service generated in NF Console

SERVICE\_IP

IP of NF Service App

SERVICE\_PORT

IP of NF Service App

APPWAN\_ACTION

Select an action to perform on the appwan in NF Network

APPWAN\_NAME

Name of NF APPWAN to be used in NF Console

APPWAN\_PRIVATE\_GATEWAY

Endpoint Name in Private Datacenter Gateway to be included in AppWan

APPWAN\_PRIVATE\_CLIENT

Endpoint Name for Client to be included in AppWan

APPWAN\_SERVICE

Service Name to be included in AppWan

LOCATION

Azure Cloud DC Location where to deploy GW

SUBNET\_PREFIX

Subnet CIDR in Azure Cloud DC Location where to deploy GW

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

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

AZURE\_SUBSCRIPTION\_ID

RESOURCE\_GROUP\_NAME

RESOURCE\_GROUP\_LOC

ENVIRONMENT

NETWORK\_ACTION

NETWORK\_NAME

GATEWAY\_ACTION

GATEWAY\_NAME

SERVICE\_ACTION

SERVICE\_NAME

SERVICE\_IP

SERVICE\_PORT

APPWAN\_ACTION

APPWAN\_NAME

APPWAN\_PRIVATE\_GATEWAY

APPWAN\_PRIVATE\_CLIENT

APPWAN\_SERVICE

LOCATION

SUBNET\_PREFIX

Build

Tenant ID in Azure

Subscription ID in Azure

RG Name in Azure

RG Location in Azure

☐ KEEP\_RG

Not to check this if RG can be deleted

sandbox

Select NF Console Environment to spin the network and gateways in

get

Select an action to perform on the network in NF

DEMONET

Name to be used to create a network with

get

Select an action to perform on the gateway in NF Network

AZCPEGWx0xWESTUS

Name of NF Gateway generated in NF Console

create

Select an action to perform on the service in NF Network

AZCPEGWx0xWESTUS--10.20.10.5--22

Name of NF Service generated in NF Console

10.20.10.5

IP of NF Service App

22

IP of NF Service App

create

Select an action to perform on the appwan in NF Network

appwan-ssh-22

Name of NF APPWAN to be used in NF Console

private-gateway-name

Endpoint Name in Private Datacenter Gateway to be included in AppWan

client-name

Endpoint Name for Client to be included in AppWan

AZCPEGWx0xWESTUS--10.20.10.5--22

Service Name to be included in AppWan

westus

Azure Cloud DC Location where to deploy GW


10.20.10.0/24

Subnet CIDR in Azure Cloud DC Location where to deploy GW

## 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"/>                       | 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" value="centralus"/>     | RG Location in Azure   |
|                        | <input type="checkbox"/> KEEP_RG           | Not to check this if RG can be deleted                               |
| ENVIRONMENT            | <input type="text" value="sandbox"/>       | Select NF Console Environment to spin the network and gateways in    |
| NETWORK_ACTION         | <input type="text" value="delete"/>        | 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="delete"/>        | Selection an action to perform on the gateway in NF Network          |
| GATEWAY_NAME           | <input type="text"/>                       | Name of NF Gateway generated in NF Console                           |
| SERVICE_ACTION         | <input type="text" value="get"/>           | Selection an action to perform on the service in NF Network          |
| SERVICE_NAME           | <input type="text"/>                       | Name of NF Service generated in NF Console                           |
| SERVICE_IP             | <input type="text"/>                       | IP of NF Service App   |
| SERVICE_PORT           | <input type="text"/>                       | IP of NF Service App   |
| APPWAN_ACTION          | <input type="text" value="get"/>           | Selection an action to perform on the appwan in NF Network           |
| APPWAN_NAME            | <input type="text"/>                       | Name of NF APPWAN to be used in NF Console                           |
| APPWAN_PRIVATE_GATEWAY | <input type="text"/>                       | Endpoint Name in Private Datacenter Gateway to be included in AppWan |
| APPWAN_PRIVATE_CLIENT  | <input type="text"/>                       | Endpoint Name for Client to be included in AppWan                    |
| APPWAN_SERVICE         | <input type="text"/>                       | 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

3. Done