
(HTTPS://WWW.ASSISTANZ.COM/)





# Creating custom NAT network in windows c

April 13, 2017 Posted by: Loges Category: Blog, Containers

(https://www.assist
windows-co
ne


# Creating custom NAT network in windows container

In this blog, we will show you creating custom NAT network in windows container using docker commands.

## INTRODUCTION

The windows container networks are similar to virtual networks like HYPER-V, VMWARE. The container will ha
which connected to a virtual switch. We can also create our own virtual networks, customize IP address spac

## NAT NETWORKING OVERVIEW

- Go to PowerShell window and type the below command to get the network information.

**Get-ContainerNetwork**

```
PS C:\> Get-ContainerNetwork

Name Id                                   Subnets          Mode SourceMac DNSServer
---- --                                   -------          ---- --------- ---------
nat  e7d75de1-41bf-4fd9-a020-047d2c421ec5 {172.23.176.0/20} NAT

PS C:\> _
```

(https://www.assistanz.com/wp-content/uploads/2017/04/image-141.png)

- This network (NAT) defined by **default** while building the container host on windows 2016 server. The subnet tha network is **172.23.176.0/20.** As we spinning up the containers, the virtual NIC in this containers has been gone int containers have to use this network to reach the **internet.**

## WINDOWS CONTAINER NETWORK TYPES

- There are **four** types of network types available for **docker networking.** Go to PowerShell window and type belc view the help page of **new-containernetwork** cmdlet.

**help new-containernetwork**

```
PS C:\> help New-ContainerNetwork

NAME
    New-ContainerNetwork

SYNTAX
    New-ContainerNetwork [-Name] <string> [-SubnetPrefix <string[]>] [-GatewayAddress <string[]>
    [-Mode {NAT | Transparent | L2Bridge | L2Tunnel}] [-SourceMac <string>] [-DNSServers
    <string[]>] [-DNSSuffix <string>] [-NetworkAdapterName <string>] [-WhatIf] [-Confirm]
    [<CommonParameters>]


ALIASES
    None


REMARKS
    Get-Help cannot find the Help files for this cmdlet on this computer. It is displaying only
    partial help.
        -- To download and install Help files for the module that includes this cmdlet, use
    Update-Help.
```

(https://www.assistanz.com/wp-content/uploads/2017/04/image-142.png)

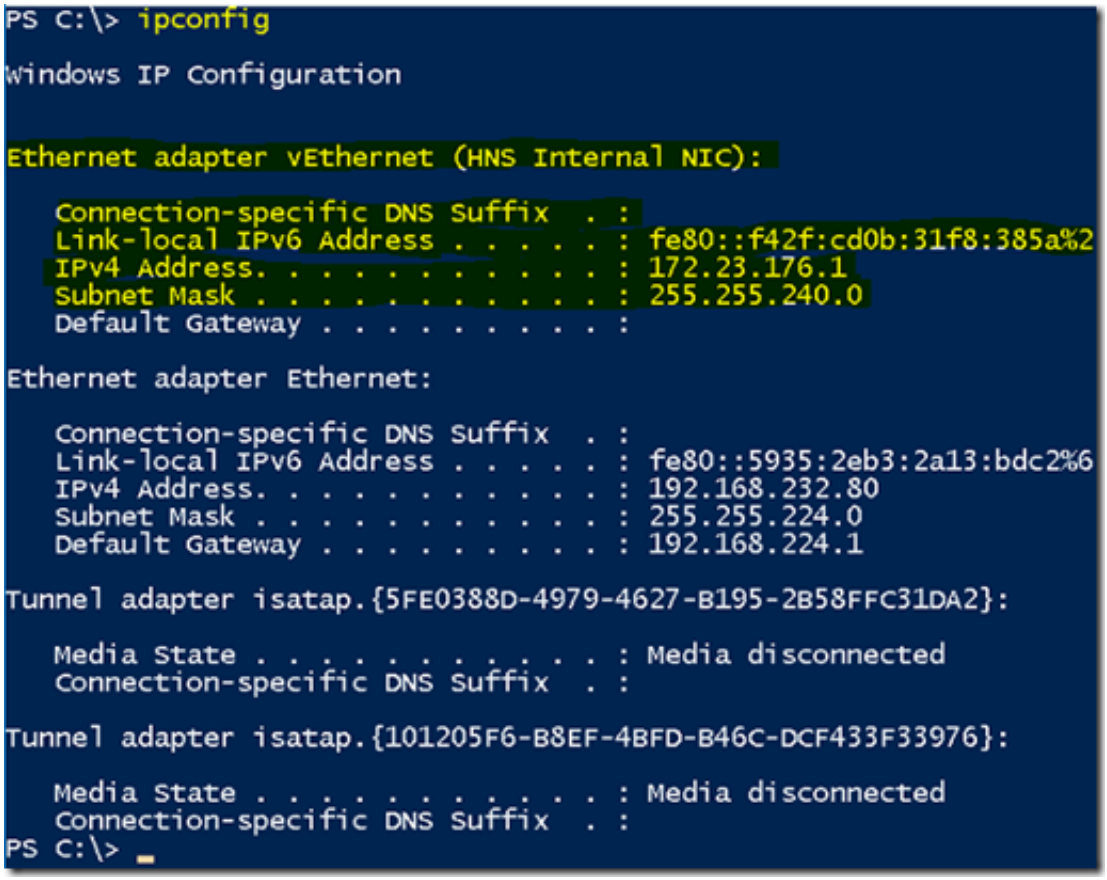
- The four network driver options are **NAT, Transparent, L2Bridge and L2Tunnel.**

**NAT –** We will get the NAT (Network Address Translation) network by default. The container in this network w **isolated network**. To reach the internet they need to use the IP address of windows container host.

**Transparent –** It's little bit different to NAT. Each container in this network will get an IP address from the p windows container host. It's similar to **bridged network**.

**L2Bridge & L2Tunnel –** These two networks drivers are using for public and private cloud deployments. It's g (Software Defined Network) network environments. Typically we will use L2 type networks in multiple contair environments.

- Go to PowerShell window and type **ipconfig** command.



(https://www.assistanz.com/

content/uploads/2017/04/image-143.png)

- You can see the virtual ethernet adapter that has an IP address as **172.23.176.1**and subnet mask as **255.255.240.0**

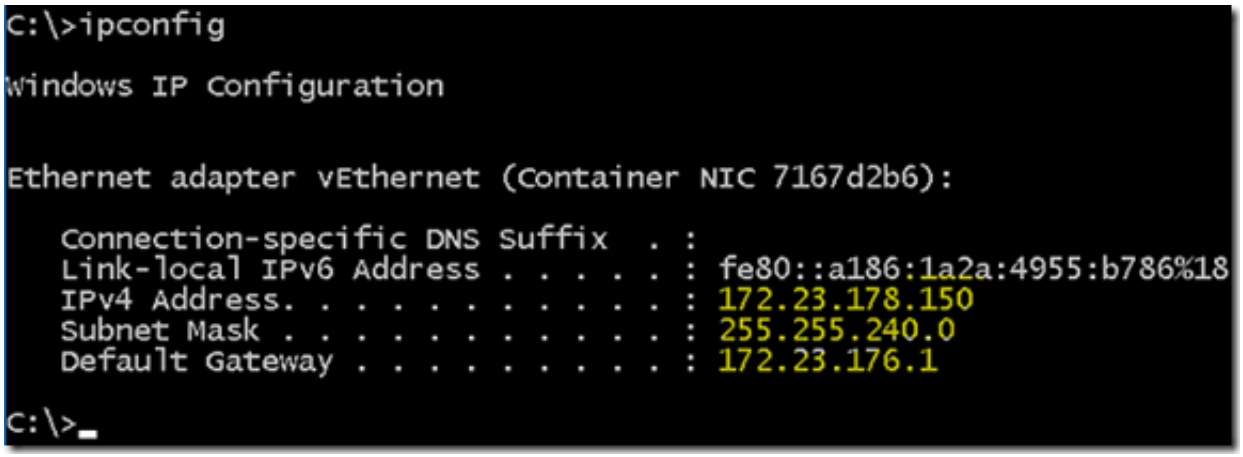
network has the IP address of 192.168.232.80 and subnet mask as 255.255.224.0.

- By default, the containers that we launch in this container host will go to 172 network.
- Launching a container using **docker run** command to check the default NAT networking.

**docker run –it microsoft/nanoserver**

(https://www.assistanz.com/wp-content/uploads/2017/04/image-144.png)

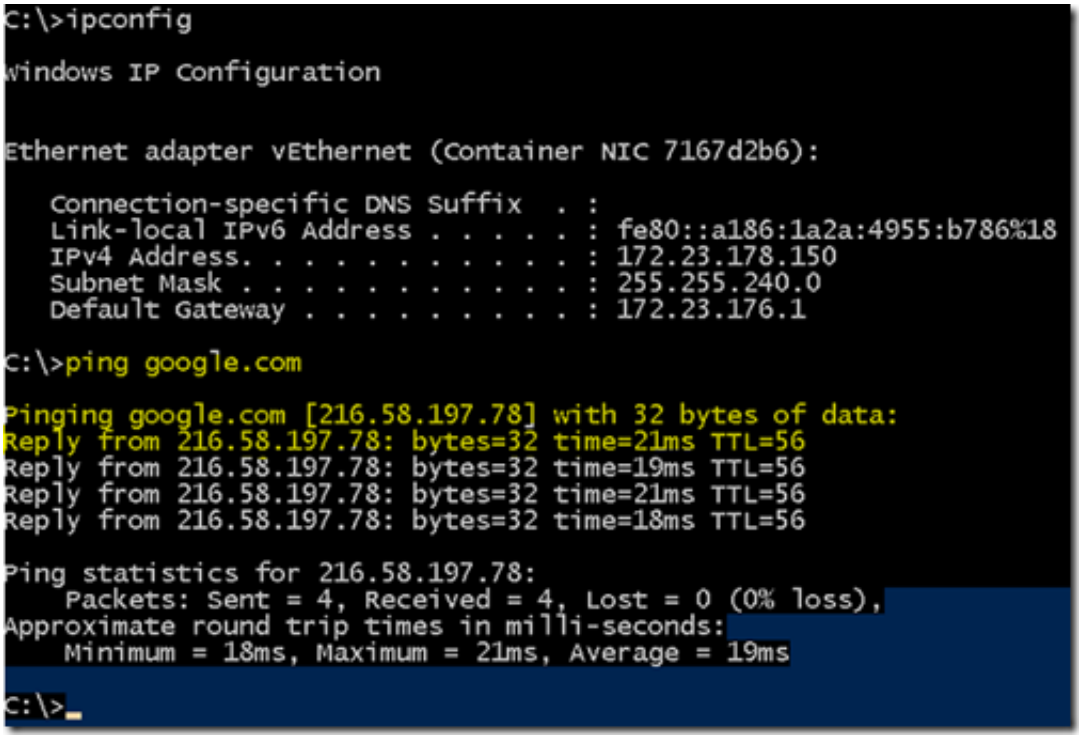
- Once the container is up and running, type **ipconfig** inside the container.



(https://www.assist

content/uploads/2017/04/image-145.png)

- The IP address that has assigned to this container is **172.23.178.150.** Since NAT has been configured, we will able t
domain.



(https://www.assistanz.com/w

content/uploads/2017/04/image-146.png)

## CREATING CUSTOM NAT NETWORK

- As if we are using this address space **172.18** already, we can create our own **NAT** network for our environment. To docker service using below command.

**stop-service docker**

(https://www.assistanz.com/wp-content/uploads/2017/04/ima

- Remove all the container network using the below command.

**Get-ContainerNetwork | Remove-ContainerNetwork**

Press A to confirm the deletion.

```
PS C:\> Get-ContainerNetwork | Remove-ContainerNetwork

Confirm
Remove-ContainerNetwork will remove the container network "".
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is "Y
PS C:\> _
```

(https://www.assistanz.com/wp-content/uploads/2017/04/image-148.png)

- To customize the address space for **NAT** network, we need to add the content in the **daemon.json** file. Use the b create the **daemon.json** file under **C:\ProgramData\docker\config** folder.

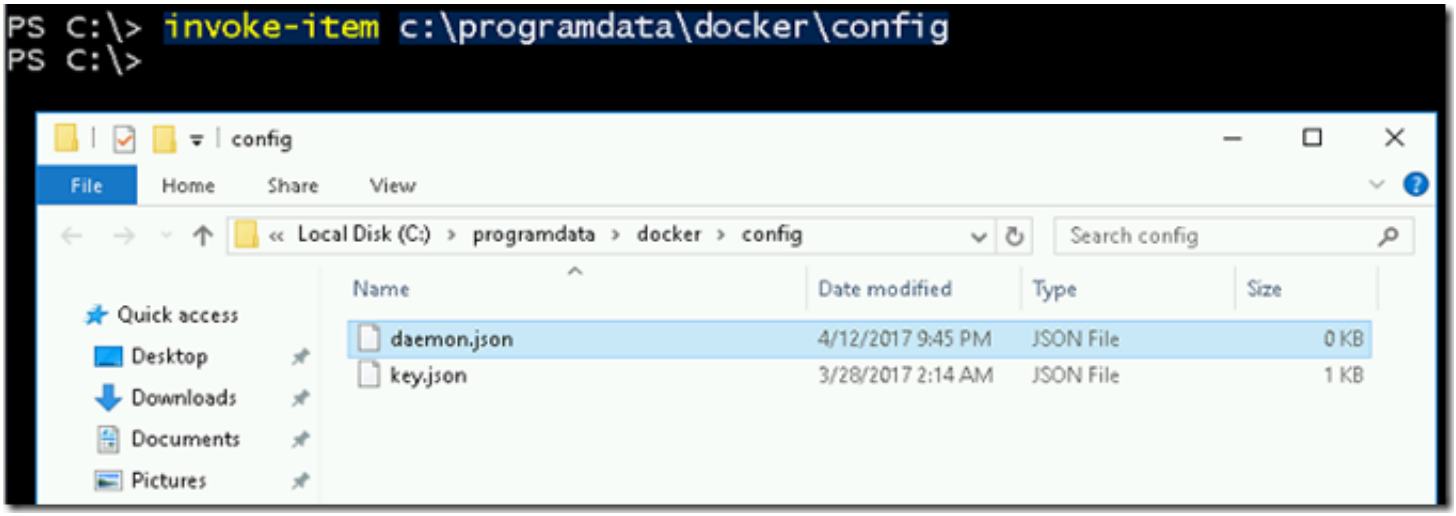
**New-Item –ItemType file –Path C:\ProgramData\docker\config –Name daemon.json**

```
PS C:\> New-Item -ItemType file -Path C:\ProgramData\docker\config\ -Name d

    Directory: C:\ProgramData\docker\config

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        4/12/2017   9:45 PM              0 daemon.json

PS C:\> _
```

(https://www.assistanz.com/wp-content/uploads/2017/04/image-149.png)

- Navigate the **C:\ProgramData\docker\config** folder using **invoke-item** command.

**invoke-item c:\programdata\docker\config**



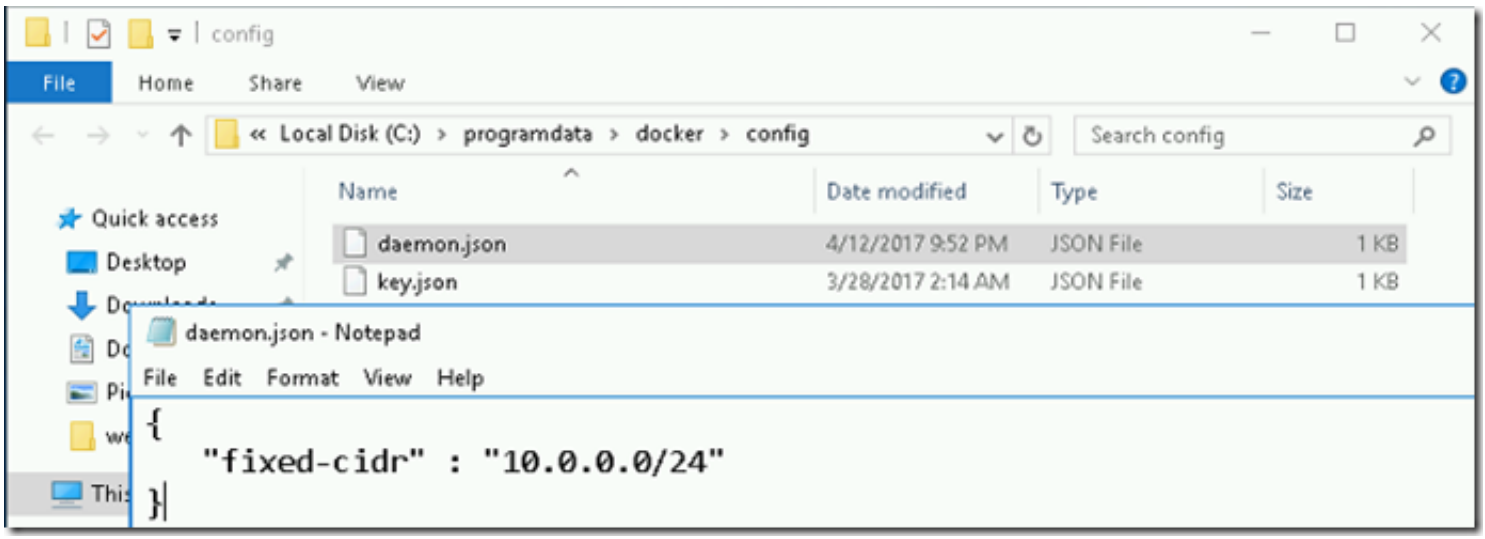

(https://www.assistanz.com/wp-content/uploads/2017/04/image-150.png)

- Add the below content in **daemon.json** file.

**{**

**"fixed-cidr" : "10.0.0.0/24"**

**}**




(https://www.assistanz.com/wp-content/uploads/2017/04/image-151.png)

- Save the file and close it.
- Start the docker service using below command.

**Start-Service docker**

```
PS C:\> Start-Service docker
WARNING: Waiting for service 'docker (Docker)' to start...
PS C:\>
PS C:\>
PS C:\> get-service Docker

Status   Name               DisplayName
------   ----               -----------
Running  Docker             Docker


PS C:\> _
```

(https://www.ass

content/uploads/2017/04/image-152.png)

- Docker service is up and running fine. Check the NAT networking status using below command.

**Get-ContainerNetwork**

```
PS C:\> Get-ContainerNetwork

Name Id                                   Subnets       Mode SourceMac DNSServers DNSSuffix
---- --                                   -------       ---- --------- ---------- ---------
nat  640ecfc2-e770-4053-868e-5b831f9895d4 {10.0.0.0/24} NAT

PS C:\> _
```

(https://www.assistanz.com/wp-content/uploads/2017/04/image-153.png)

- The subnet CIDR has been changed to **10.0.0.0/24** network. We have successfully modified network address spac
- Create a new container and make sure this new network works. Create a new container using the below commar

**docker run -it microsoft/nanoserver**

```
PS C:\> docker run -it microsoft/nanoserver_
```

(https://www.assistanz.com/wp-

content/uploads/2017/04/image-154.png)

- Once the container is up and running type **ipconfig** to check the IP address information.

```
C:\>ipconfig

Windows IP Configuration


Ethernet adapter vEthernet (Container NIC 84836501):

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::dc62:40cb:4659:df2e%18
   IPv4 Address. . . . . . . . . . . : 10.0.0.25
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.0.0.1

C:\>_
```

(https://www.assistanz.com/wp-content/uploads/2017/04/image-155.png)

- Also, we are able to ping the internet domain without any problem.

```
C:\>ipconfig

Windows IP Configuration


Ethernet adapter vEthernet (Container NIC 84836501):

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::dc62:40cb:4659:df2e%18
   IPv4 Address. . . . . . . . . . . : 10.0.0.25
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.0.0.1

C:\>ping google.com

Pinging google.com [216.58.220.46] with 32 bytes of data:
Reply from 216.58.220.46: bytes=32 time=168ms TTL=56
Reply from 216.58.220.46: bytes=32 time=264ms TTL=56
Reply from 216.58.220.46: bytes=32 time=724ms TTL=56
Reply from 216.58.220.46: bytes=32 time=814ms TTL=56

Ping statistics for 216.58.220.46:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 168ms, Maximum = 814ms, Average = 492ms

C:\>_
```

(https://www.assistanz.com/wp-content/uploads/2017/04/image-156.png)

**VIDEO**