

Article

[John Murray](#) · Apr 12m read

Docker Containers on Windows sometimes unable to get ports during startup

I have recently started making more use of Docker Desktop on my Windows 11 workstation, particularly in conjunction with VS Code's [Remote - Containers](#) extension and the [iris-python-template](#) repo from [@Guillaume Rongier](#)

Sometimes the container would start smoothly, but other times it would fail and report being unable to use a port.

Eventually I tracked the problem down. The Windows NAT Driver service apparently uses an OS facility to reserve a large range of ports.

Here's what an admin-level command shell showed when the container wouldn't start:

```
PS C:\WINDOWS\system32> netsh int ipv4 show exclude proto=tcp
```

```
Protocol tcp Port Exclusion Ranges
```

Start Port	End Port	
80	80	
443	443	
5357	5357	
49685	49784	
49785	49884	
49885	49984	
50000	50059	*
50160	50259	
50260	50359	
50360	50459	
50460	50559	
50560	50659	
52182	52281	
52579	52678	
52679	52778	
52779	52878	
52879	52978	
52979	53078	
53170	53269	
53270	53369	
53370	53469	
53470	53569	
53582	53681	
53682	53781	

* - Administered port exclusions.

```
PS C:\WINDOWS\system32>
```

Notice how these reservations include ones that the docker-compose.yml of iris-python-template specifies:

```
ports:
  - 52775:52773
```

Solution was to restart the winnat service:

```
PS C:\WINDOWS\system32> net stop winnat
```

The Windows NAT Driver service was stopped successfully.

```
PS C:\WINDOWS\system32> netsh int ipv4 show exclude proto=tc
p
```

Protocol tcp Port Exclusion Ranges

Start Port	End Port	
-----	-----	
80	80	
443	443	
5357	5357	
50000	50059	*

* - Administered port exclusions.

```
PS C:\WINDOWS\system32> net start winnat
```

The Windows NAT Driver service was started successfully.

```
PS C:\WINDOWS\system32> netsh int ipv4 show exclude proto=tc
p
```

Protocol tcp Port Exclusion Ranges

Start Port	End Port
-----	-----
80	80
443	443
5357	5357

50000 50059 *

* - Administered port exclusions.

PS C:\WINDOWS\system32>

Interestingly the service didn't immediately reserve the port ranges again. I haven't dug any deeper, but am posting this article in case someone else hits the same problem.

[#Docker](#) [#Microsoft Windows](#) [#Other](#)

Source URL: <https://community.intersystems.com/post/docker-containers-windows-sometimes-unable-get-ports-during-startup>