Installing CodeReady Containers (OpenShift4) on Hyper-V (Win 10)

Stefaan De Geyter Follow
Feb 5 · 5 min read

Difficulty: 3/5
5 min read

Red Hat OpenShift 4 on your laptop: Introducing Red Hat CodeReady Containers

As a DevOps engineer at FlowFactor, it is always convenient to be able to run your own container platform locally to test actual use cases.

Succeeding Minishift, which allows developers to run an Openshift 3.x container platform locally, you can now use **Red Hat's CodeReady Containers** (CRC) to run an Openshift 4.x platform.

While the installation instructions look very simplistic, you may run into some issues invisible at first which take time to solve. Let's find out what those are and how to mitigate them!

A couple of **prerequisites** you need to check out first:

```
- 4 vCPUs, 8GB memory and 35GB of storage space (get some more if you
have big plans)
- Windows 10 build 1709 or later (I'm running 1909)
```

```
    - Enable the hyper-v feature
    - Add your user to the hyper-v Administrators group
 - CRC binaries and your pull secret https://try.openshift.com/
    - Extract the archive and add the files to your PATH variable
 - Disable your VPNs connections
```

According to the official documentation all you need to do now is open a command prompt and run **crc setup** followed by **crc start** and follow the on-screen instructions.

While this can work perfectly, stopping, starting and re-creating your cluster can break things. You will most likely run into networking issues. Fortunately, I already ran into most of them and found a workaround.

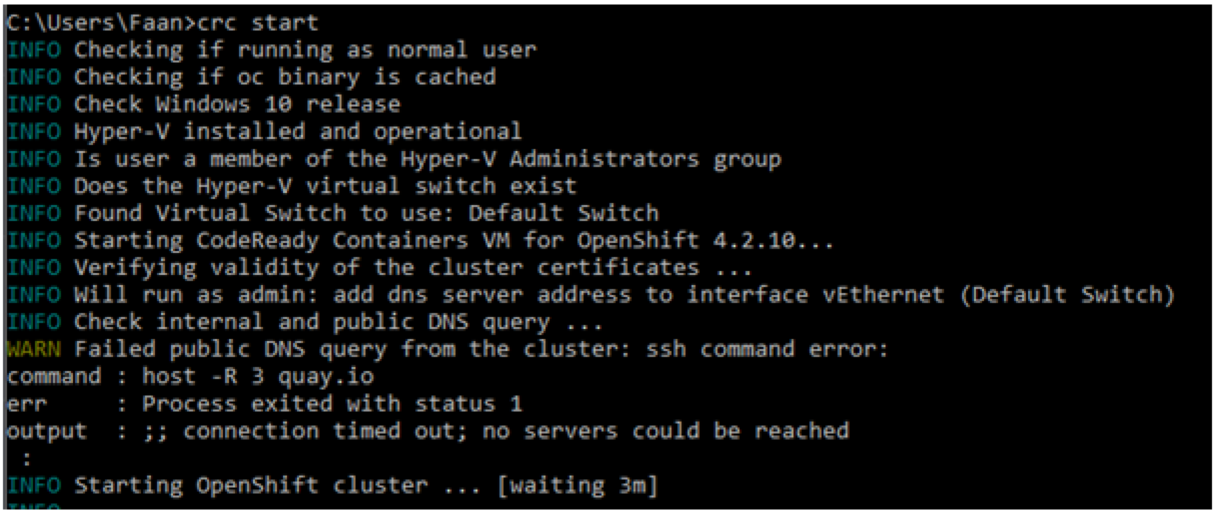
Tip: to show additional output use:

```
crc start --log-level debug
```

Unable to contact quay.io

This is the most frequently encountered issue. Stopping and starting may cause you to see it.

You will still be able to access your cluster, but you won't be able to fetch the list of operators from OperatorHub for example.



```
C:\Users\Faan>crc start
INFO Checking if running as normal user
INFO Checking if oc binary is cached
INFO Check Windows 10 release
INFO Hyper-V installed and operational
INFO Is user a member of the Hyper-V Administrators group
INFO Does the Hyper-V virtual switch exist
INFO Found Virtual Switch to use: Default Switch
INFO Starting CodeReady Containers VM for OpenShift 4.2.10...
INFO Verifying validity of the cluster certificates ...
INFO Will run as admin: add dns server address to interface vEthernet (Default Switch)
INFO Check internal and public DNS query ...
WARN Failed public DNS query from the cluster: ssh command error:
command : host -R 3 quay.io
err     : Process exited with status 1
output  : ;; connection timed out; no servers could be reached
 :
INFO Starting OpenShift cluster ... [waiting 3m]
```

You can fix this by running **crc config set nameserver 8.8.8.8** followed by **crc start**.

This will allow your cluster to perform a DNS lookup via Google's public DNS server instead of the internal address provided by the hyper-v default switch.

> INFO Adding 8.8.8.8 as nameserver to the instance …

Some organizations might not allow you to use Google's public DNS directly. Read how to do this below.

Cluster TLS certificates have expired, renewing them… fails to contact apps-crc.testing

Your cluster is unable to resolve **apps-crc.testing** and **crc.testing** which should actually point to itself. **crc start** normally adds a loopback DNS reference (via hosts.ics) but apparently there can be a case where this fails. Delete your cluster with **crc delete** and try again.

Worst case, adding your cluster's IP (find it in %USERPROFILE%\.crc\machines\crc) in %SystemRoot%\System32\drivers\etc\hosts can resolve this as well. Note that if your cluster's IP changes because the default hyper-v switch decides to hand you a new internal IP address, it will slip up again.

Adding user's pull secret… takes a very long time

There seem to be some cases where it takes a very long time before telling you the cluster is unable to start because the pull secret could not be added.

Like above, first verify that you can resolve **apps-crc.testing** and **crc.testing** with the correct IP.

Secondly, this step tries to contact your single-node cluster via SSH commands to update your pull secret. In my case these were not going through because I was using a custom SSH client on my PATH. Removing it and keeping Windows' own OpenSSH located in system32 solved this issue.

Been messing with your CRC installation too much?

crc delete -> crc setup -> **crc start** will get you a new cluster. If not, run this command first:

```
crc delete --clear-cache
```

This will reset your crc user settings.

Hyper-v switch settings are all over the place

No worries, you can simply disable the Default switch and Windows will automatically recreate it on next reboot. Its new settings may differ from the old ones though.

Delays in DNS resolution

Quite possibly you are having the same issue I had: very slow browsing (resolving host…).

Creating hyper-v switches creates additional network adapters. In turn they can have a higher priority (= lower metric) set than your ethernet or WiFi adapter. Hence, they will take precedence on DNS lookups. Those requests will fail, causing your browser to be a lot slower.

You can tune the metric parameter for ipv4 and ipv6 in the advanced TCP/IP settings or you can simply disable the adapters. Note that if it comes to the Default switch, it will always be re-enabled on the next reboot.

Hyper-v Default switch is missing or you cannot change specific network configurations

So, the biggest issue here is that the default switch is kind of managed by Windows itself and you may be restricted to use specific settings or get them persisted through reboots.

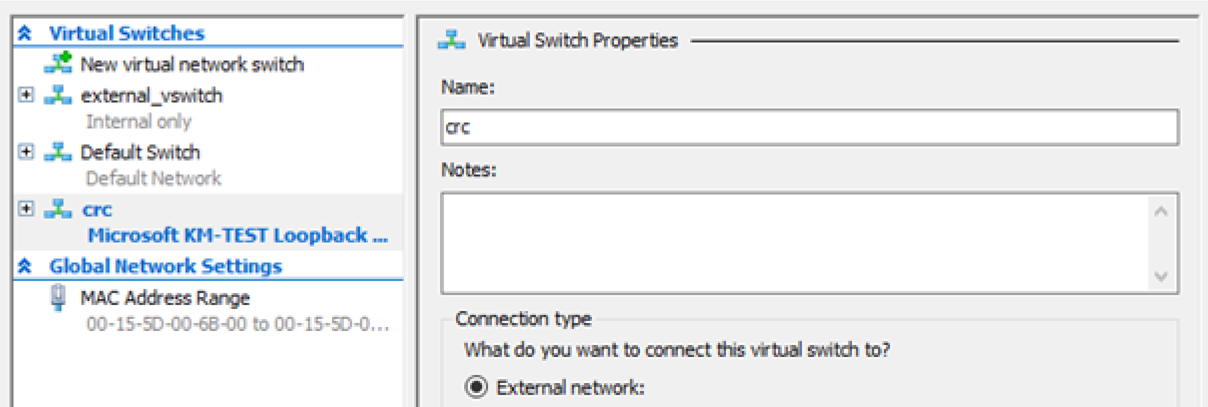
Enterprise networks may also prevent you from using the network settings issued by the Default switch. In some version of Windows, the Default Switch might not even present.

This is a bit trickier, but you can create an advanced networking configuration.

Step 1: creating a loopback adapter

Go to Device Manager, select Network Adapters, click Action and select Add Legacy hardware.
Choose the Advanced option and select Network Adapters from the list.
You are looking for Microsoft’s KM-TEST Loopback Adapter. Add it and you’re ready for the next step!

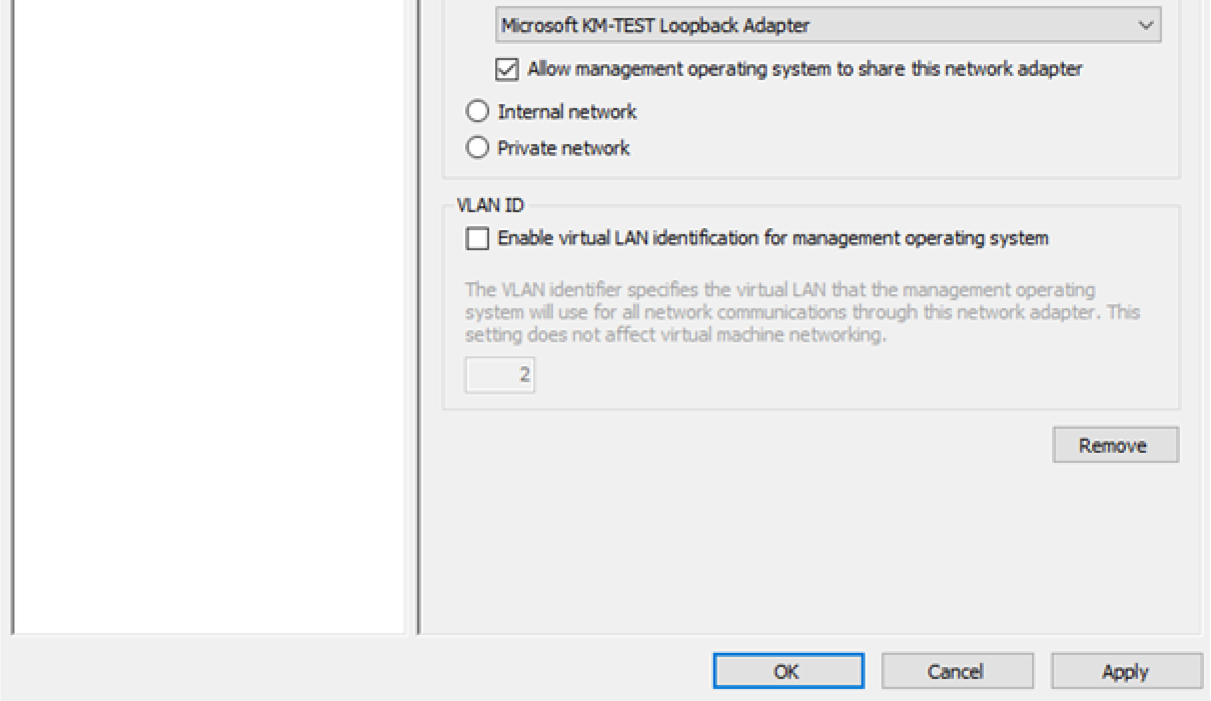
Step 2: creating a new (external type) Hyper-v switch named ‘crc’ and linking them together



Name: crc

What do you want to connect this virtual switch to?

● External network:



Microsoft KM-TEST Loopback Adapter

☑ Allow management operating system to share this network adapter

○ Internal network

○ Private network

VLAN ID

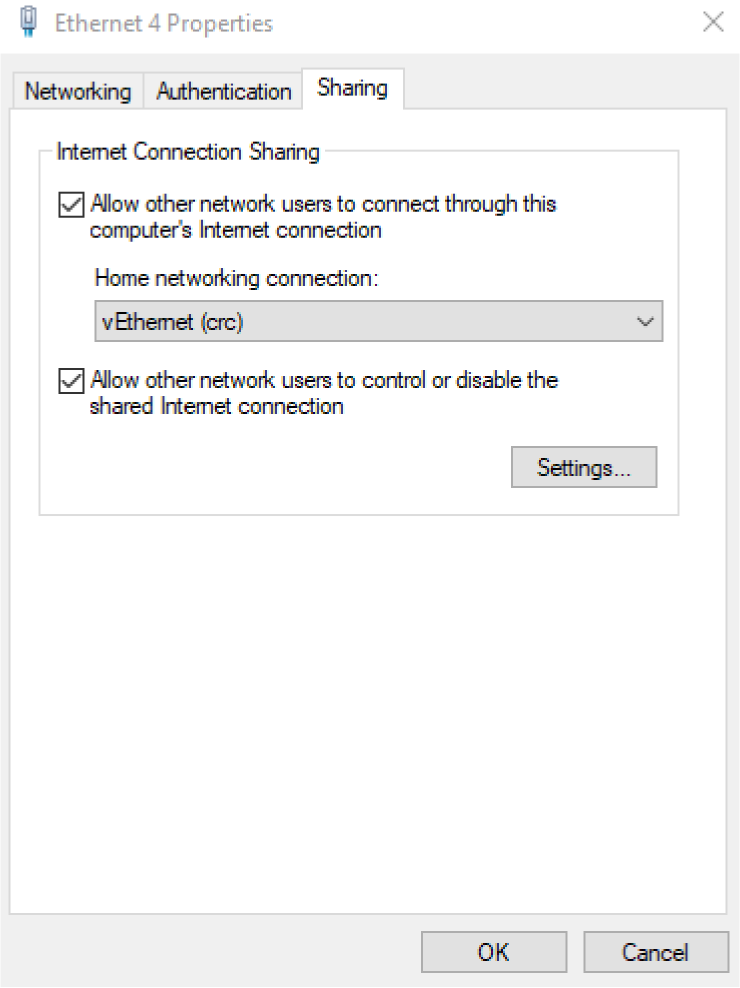
□ Enable virtual LAN identification for management operating system

The VLAN identifier specifies the virtual LAN that the management operating system will use for all network communications through this network adapter. This setting does not affect virtual machine networking.

2

Any hyper-v switch called ‘crc’ will be used instead of the Default switch when starting CRC.

Step 3: Share your WiFi or ethernet adapter’s internet with the ’crc’ switch



Ethernet 4 Properties

Internet Connection Sharing

☑ Allow other network users to connect through this computer's Internet connection

Home networking connection:

vEthernet (crc)

☑ Allow other network users to control or disable the shared Internet connection

The console output of **crc start** will now show you are using the ‘crc’ switch instead of the Default switch.

Have fun with your local CRC!

Openshift Hyper V Windows 10 DevOps