



Denodo Community

Denodo QuickStart Virtual Machine

Installation and Configuration

Denodo
Revision: 20241210

Copyright© 2024
Denodo Technologies Proprietary and Confidential
denodo.com

Contents

Introduction	3
Installation	4
Hardware Requirements	4
Configuration Overview	4
Oracle VirtualBox	5
Importing the DQVM in Oracle VM VirtualBox	5
Network Adapter configuration in Oracle VM VirtualBox	7
Launching DQVM	10
Microsoft Hyper-V	11
Importing the DQVM in Hyper-V	11
Launching DQVM	15
Network Adapter configuration in Hyper-V	16
Post-configuration tasks	20
Getting the IP of the Virtual Machine	20
IP without pre-configuration in Oracle VirtualBox	21
Getting the IP address from the command line	21
Editing your local Hosts file for using data-server as hostname	22
Accessing the Virtual Machine	24
User Authentication	24
Accessing as “root”	24
How to Access	24
Using a SSH Client (RECOMMENDED)	24
Using the Virtual Machine Console	25
Accessing Through a Web Interface	26
Changing the Keyboard Layout	28
Launching the Training Data Sources	29
List of Data Sources Included in the Virtual Machine	30
Connecting from Denodo to the Training Data Sources	30
Local Installation of Denodo Platform	30
Running Denodo Platform in the DQVM	31
Running the Denodo AI SDK container	32
Appendix: Common Issues	33

Introduction

The **Denodo QuickStart Virtual Machine** (*DQVM* from now on) is a virtual machine based on Alpine Linux (virt) that helps students doing Denodo labs. This virtual machine includes all the different data sources required for completing any on demand training course and tutorial.

By using the virtual machine, students just need to run it and all the different data sources will be available for them in their machines, so they would just need to connect to them using the Denodo Platform.

The steps for using the DQVM are as follows:

1. Install the DQVM in VirtualBox or Hyper-V (see [Installation](#)).
2. Optionally, apply additional configurations if required (see [Post-configuration tasks](#))
3. Login into de DQVM (see [Accessing the Virtual Machine](#))
4. Launch the data sources (see [Launching the Training Data Sources](#))
5. Connect Denodo to the data sources (see [Connecting from Denodo to the Training Data Sources](#))

Installation

The DQVM is a virtual machine that needs to be imported in a virtualization platform.

Software Requirements

The DQVM has been tested in **Oracle VM Virtualbox 5.2.x** and **Hyper-V Manager 10.0.18362.1**.

Hardware Requirements

The DQVM has the following hardware requirements for the virtual machine:

Processor	2 cores or more (minimum of 4 cores if the Denodo containers are going to be launched from the DQVM)
Physical memory (RAM)	By default, the base RAM memory configured for the Virtual Machine is 8 GB . The host Operating System should have that amount of RAM x2.
Disk space	By default, the initial size of the virtual machine is 600 MB but it will need more space because it needs to download the docker images of the data sources (around 25 GB). Note: the VM can extend the internal storage up to 60 GB max.

It is recommended to avoid memory overcommitment. That is, the amount of memory assigned to the virtual machine has to be backed by physical memory. Otherwise, the host operating system will have to swap to disk parts of the virtual machine. This will lead to a decrease in the performance of the Virtual Machine.

Note: if you are going to install Denodo Platform on the same machine, you also have to consider the [Denodo Platform requirements](#).

Network Requirements

The DQVM needs internet connection to the following sites:

- <https://harbor.open.denodo.com/>: for downloading the images of the Denodo Platform (optional)
- <https://github.com/denodo>: for updating automatically the training data sources.
- <https://hub.docker.com/>: for downloading the images of the data sources.

Configuration Overview

Each version of the DQVM has the appropriate downloadable file. Please, you have to download the correct one for your virtualization application. Please, note that when the DQVM is running, it will use an internal IP address which could be different on your machine.

Note: Keep in mind that all Denodo lab descriptions assume that you have the data sources running in **localhost**. When using the DQVM, you have to use **data-server** or the **IP address** of the Virtual Machine instead of localhost. Please review the next chapters to know the IP address of the virtual machine in your environment.

Oracle VirtualBox

This DQVM version is configured to use a specific network adapter and that adapter must be pre-configured to set the specific IP address that will be used in all Denodo Training on demand labs.

In the following sections, we will explain how to configure the Oracle VirtualBox network adapter and how to launch the DQVM.

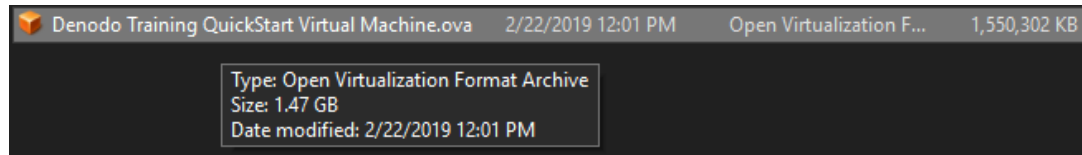
Note: This document assumes that students already have installed Oracle VM VirtualBox (version 5.2.x or newer). If you need to install it, please visit: www.virtualbox.org or [Oracle VM VirtualBox](#)

Importing the DQVM in Oracle VM VirtualBox

These are the steps for importing the DQVM:

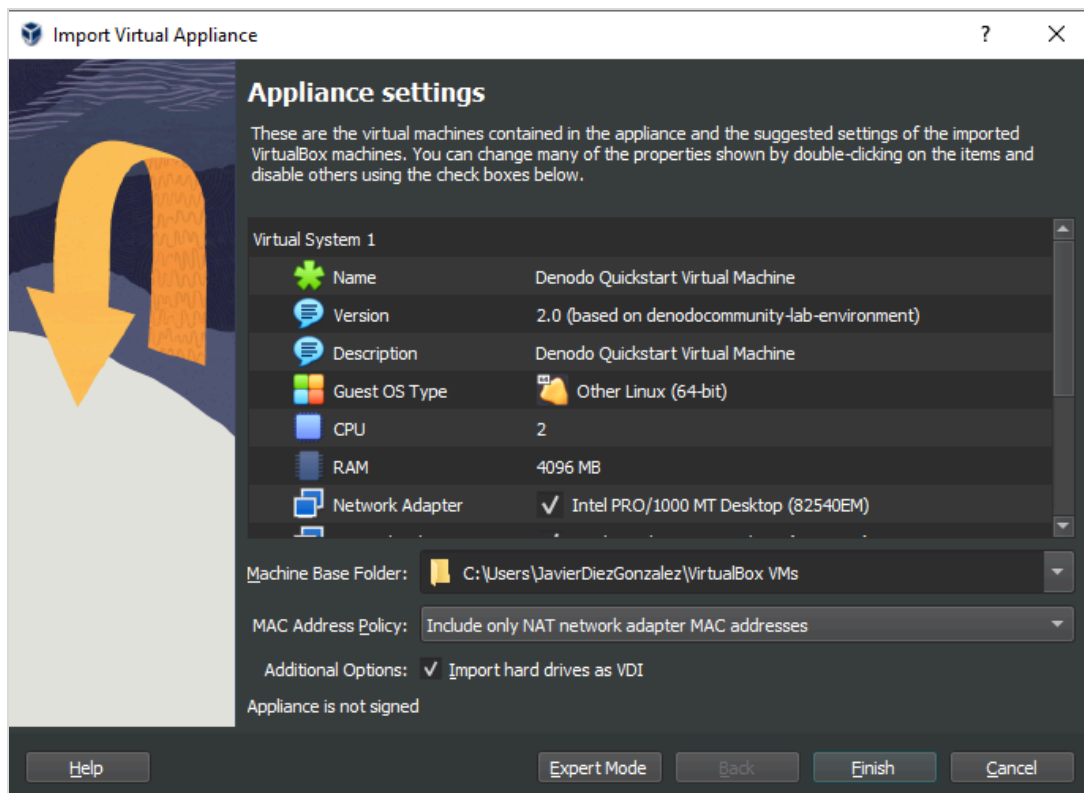
1. You have two main options for importing the DQVM in Oracle VM VirtualBox:

- a. You can double-click over the “Denodo Training QuickStart Virtual Machine.ova” file (if .ova files are opened by default with Oracle VirtualBox in your system).

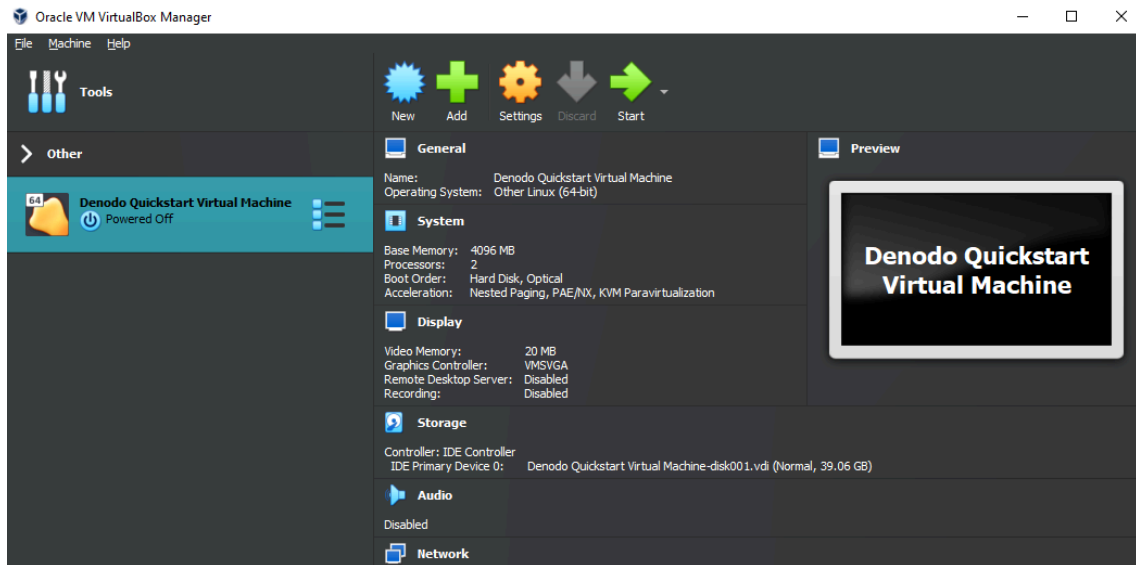


- b. You can right-click on the “Denodo Training Virtual QuickStart Machine.ova” file and select “Open With > VirtualBox Manager”.

2. You will see a new window showing the details of the DQVM (set name as: **Denodo QuickStart Virtual Machine**):



3. Click on “**Finish**” to import the Virtual Machine. When done, you will see it in the left panel with your other virtual machines.

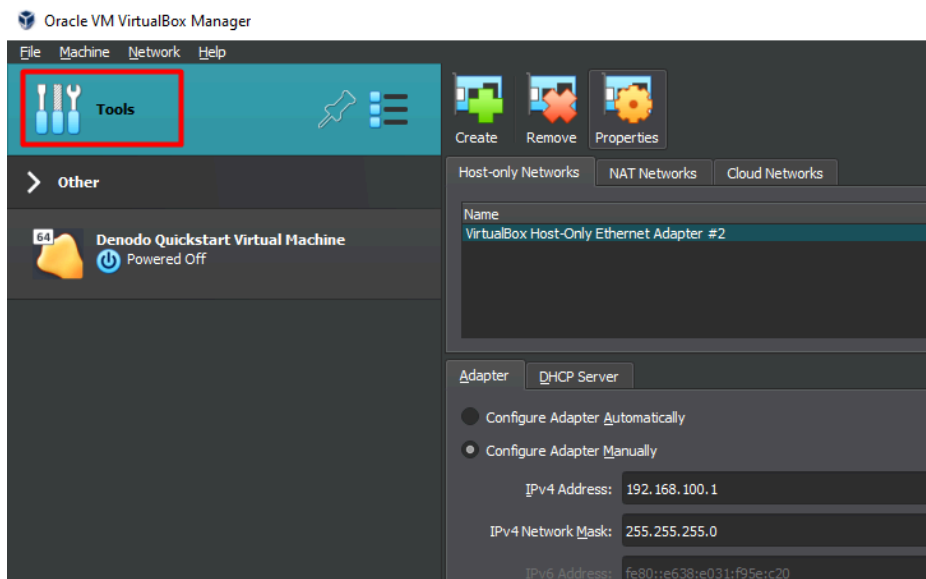


Network Adapter configuration in Oracle VM VirtualBox

Note: These steps are not mandatory for using the Virtual Machine but are recommended.

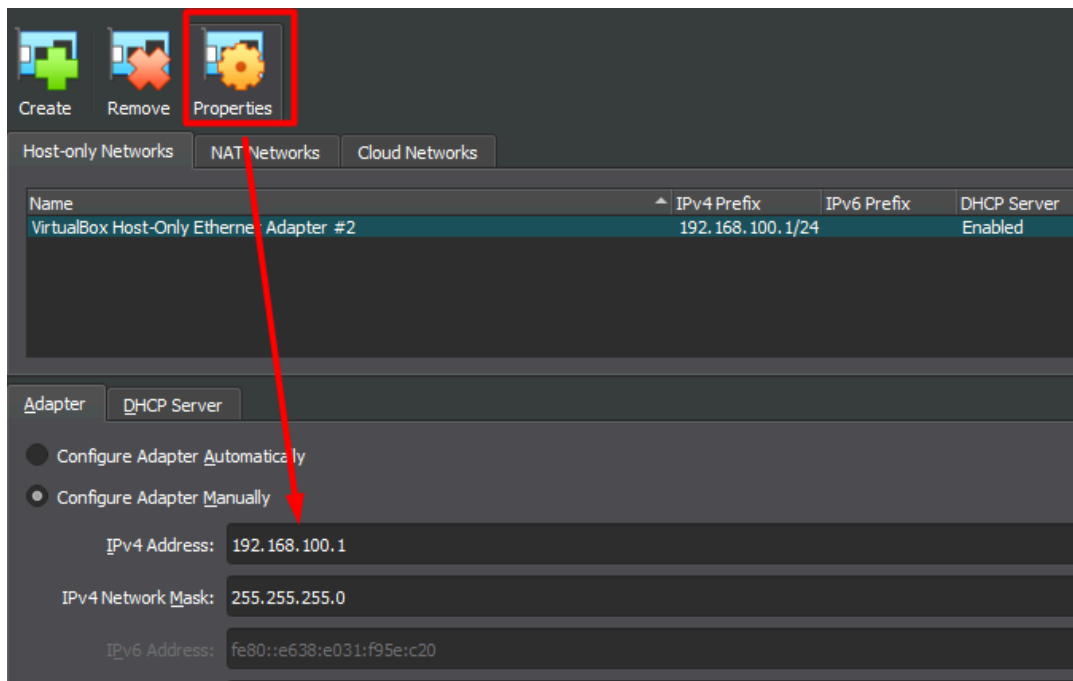
Find below the steps required for students using Oracle VM VirtualBox to configure the IP address of the DQVM:

1. Configure the network for your VirtualBox administrator tool by accessing the “Tools” option (navigating to “File > Host Network Manager” in older versions of VirtualBox).



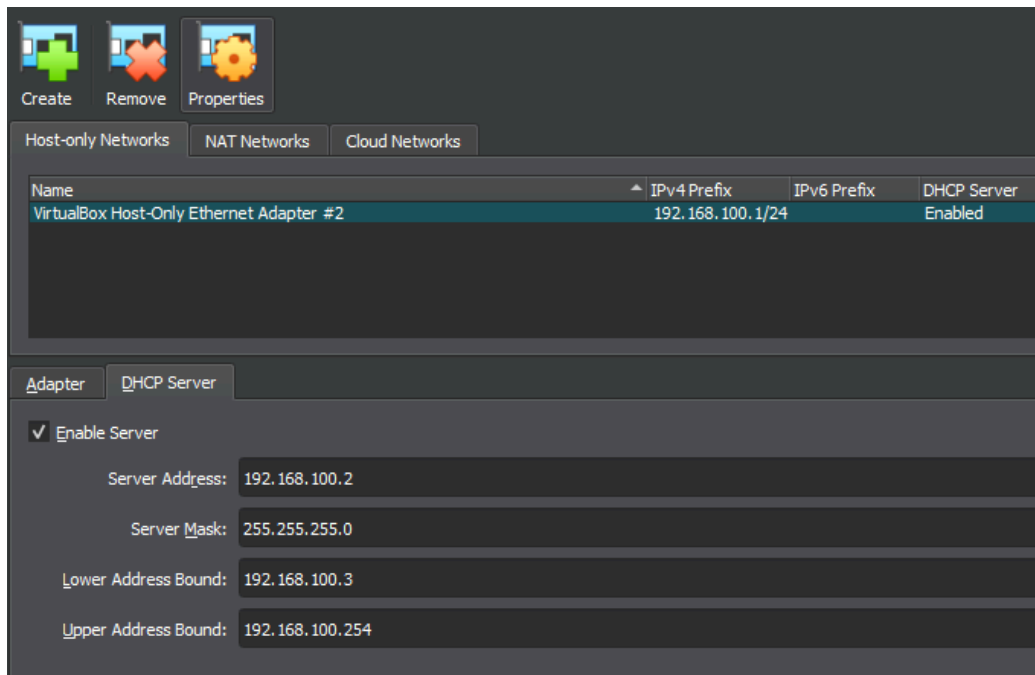
2. Click on the button “Create” to create a new VirtualBox **Host-Only** Network.
 - a. After you click on this button you will see a new “VirtualBox Host-Only Ethernet Adapter #2” listed in this panel.
3. Configure the new adapter just by clicking over it and selecting the “Properties” option:
 - a. Select the option “Configure Adapter Manually” and add the following information:
 - **IPv4 Address:** 192.168.100.1

- **IPv4 Network Mask:** 255.255.255.0

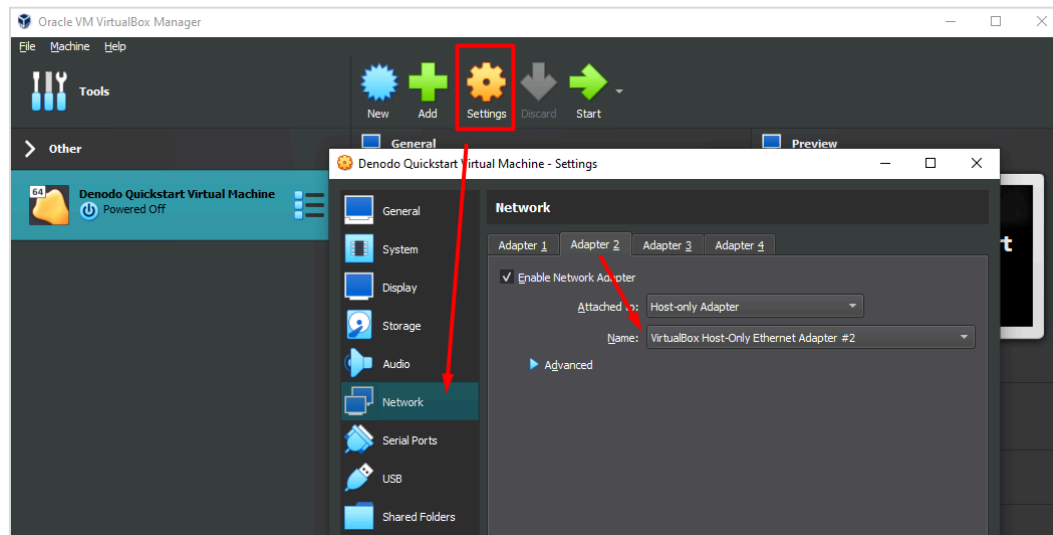


b. Inside the configuration of the adapter, open the “*DHCP Server*” tab and configure the following:

- Select **Enable Server**
- **Server Address:** 192.168.100.2
- **Server Mask:** 255.255.255.0
- **Lower Address Bound:** 192.168.100.3
- **Upper Address Bound:** 192.168.100.254



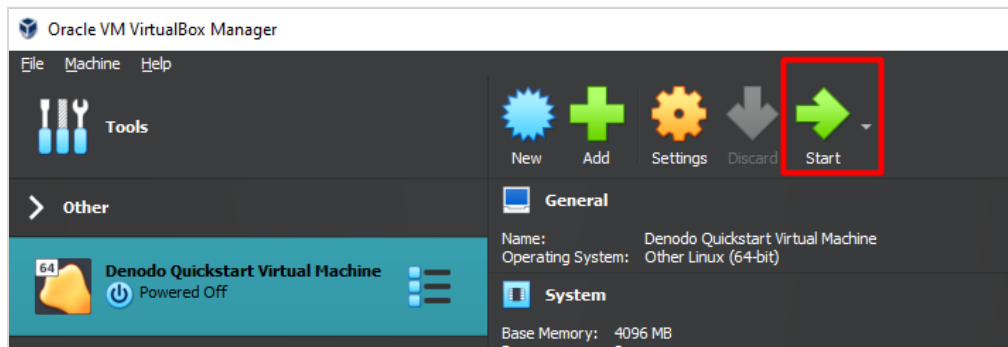
- c. After all the changes have been applied, click on “**Apply**” and then close this panel.
4. Return to the list of virtual machines, select the DQVM, right-click over it, and click on “*Settings*”. Then open the “**Network**” tab:
 - a. Open the “*Adapter 2*” and be sure that is using the following configuration:
 - **Attached to:** Host-only Adapter
 - **Name:** (Here you should select the name of the ethernet adapter you configured at step 2).



b. Then Click on “OK” to save the changes.

Launching DQVM

Once all the configuration steps are done, you can launch the DQVM just simply selecting the Virtual Machine in the left panel and clicking on “**Start**” (green arrow).



Microsoft Hyper-V

In the following sections, we will explain how to configure the Hyper-V Manager and how to launch the DQVM.

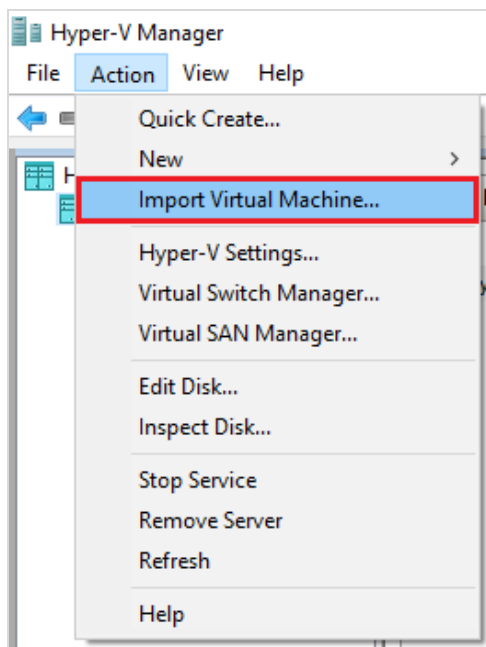
Note: This document assumes that students already have Hyper-V activated on your Windows installation. If you need to install it, please visit:

<https://docs.microsoft.com/virtualization/hyper-v-on-windows/quick-start/enable-hyper-v>

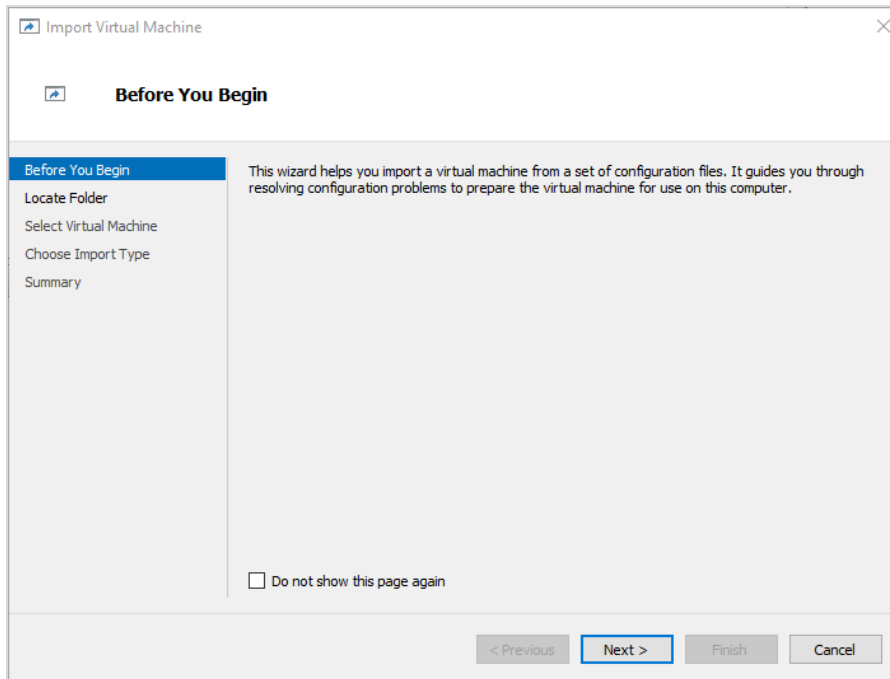
Importing the DQVM in Hyper-V

These are the steps for importing the DQVM:

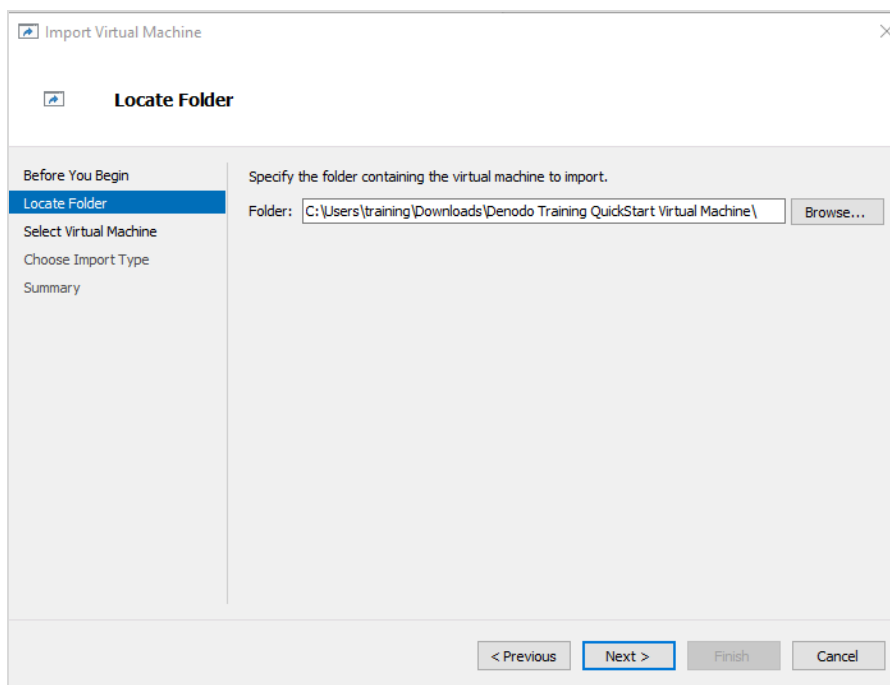
1. Unzip the file Denodo.QuickStart.Virtual.Machine.Hyper-V.zip
2. For importing the DQVM in Hyper-V, you can click on the button “Import Virtual Machine...” located on the menu “Action”.



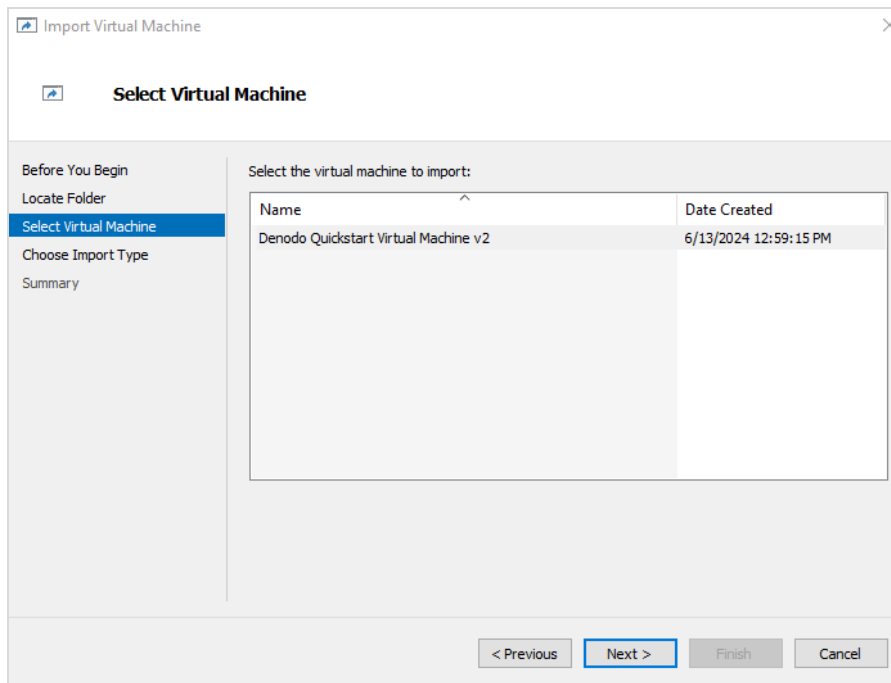
3. In the “Before you begin” page, you can check the content and click on “Next >”:



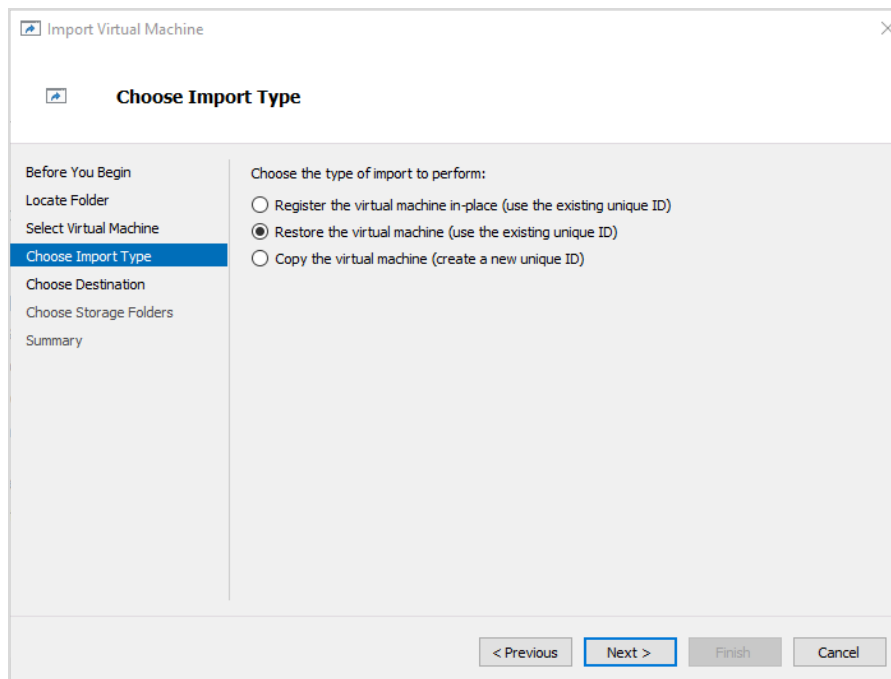
4. On the new page, you have to locate where you have the DQVM files for the Hyper-V. Please, be sure that you have selected the **root path** where you have unzipped the zip file that contains the DQVM.



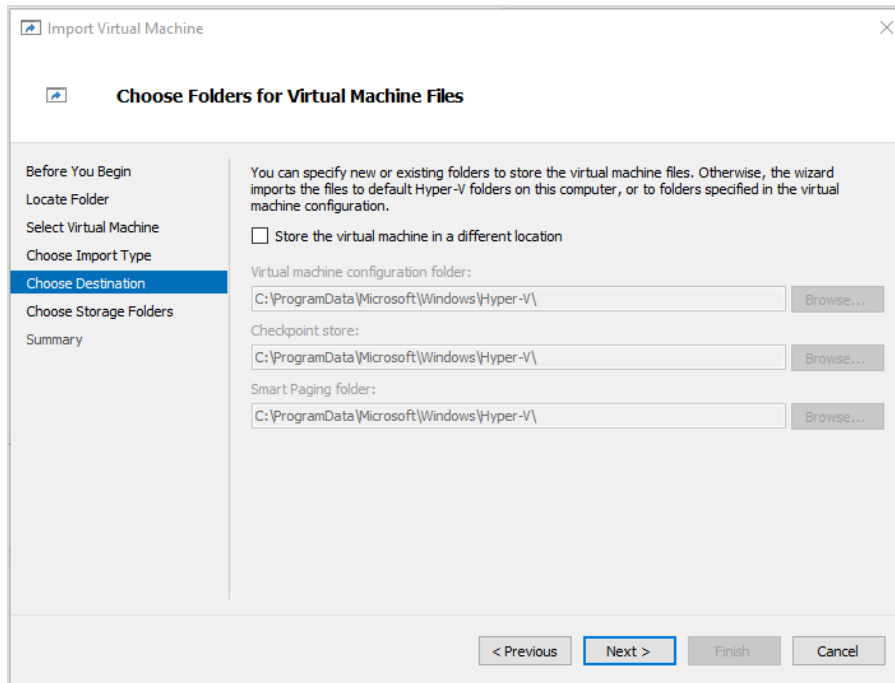
5. Select the DQVM to start the process of importing the Virtual Machine



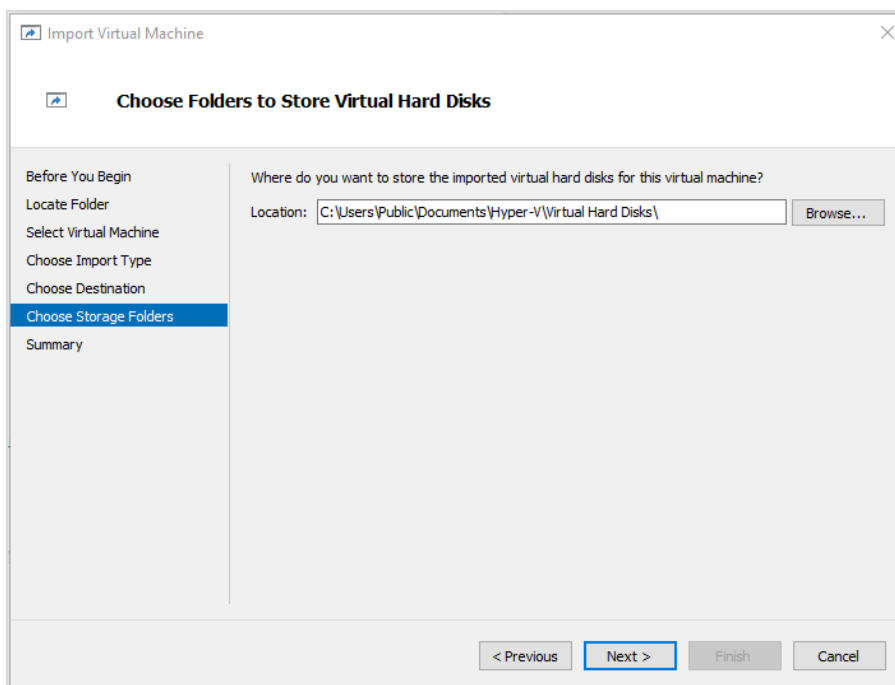
6. Select how you want to import the VM. In these steps, we are going to select “Restore”. The import process will copy the VM to a new path and will maintain the unique ID.



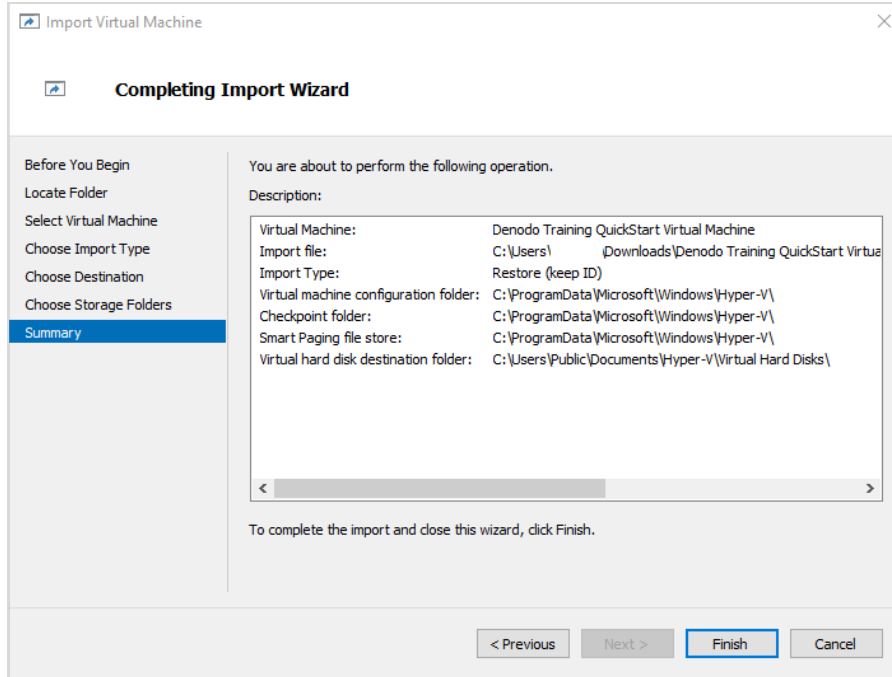
7. In the “Choose Folders for Virtual Machine Files”, you can leave the default paths. Or, if you prefer, you can choose other ones.



8. Also, you can use the default path where the hard disk will be imported, or choose a new one.

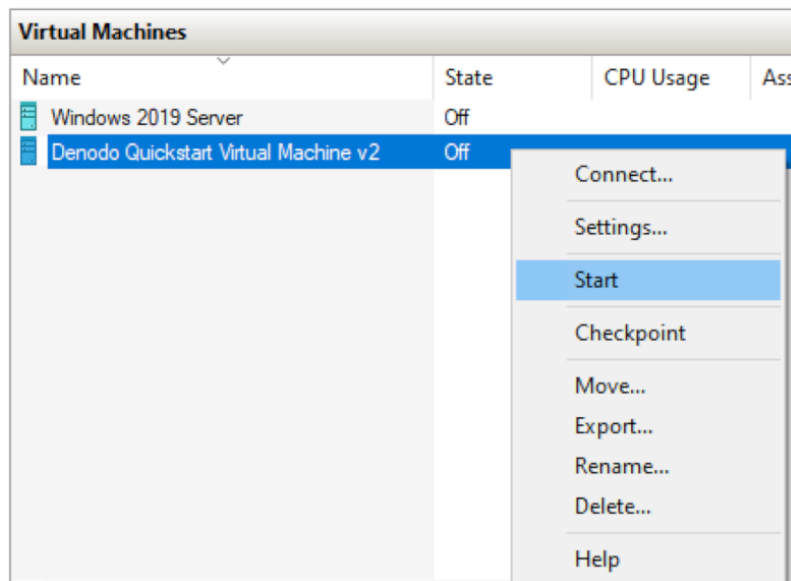


9. In the last step, you can review all the options before the import begins.



Launching DQVM

Once all the configuration steps are done, you can launch the DQVM by simply selecting the Virtual Machine in the left panel and selecting “**Start**” after doing a right-click over the VM.



Network Adapter configuration in Hyper-V

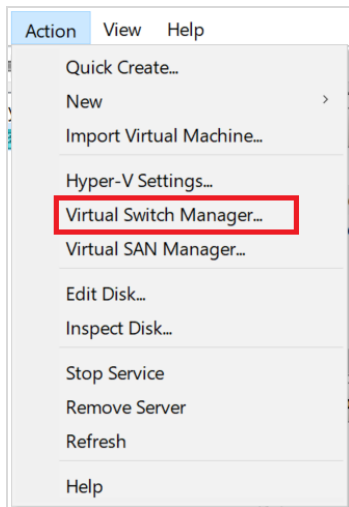
Note: These steps only are necessary if your Hyper-V DQVM does not have configured an IP address after it starts.

By default, the Virtual Machine is using the standard "Default switch" and it should work fine. You only need to perform the steps shown in this section in case networking is not working in your specific installation.

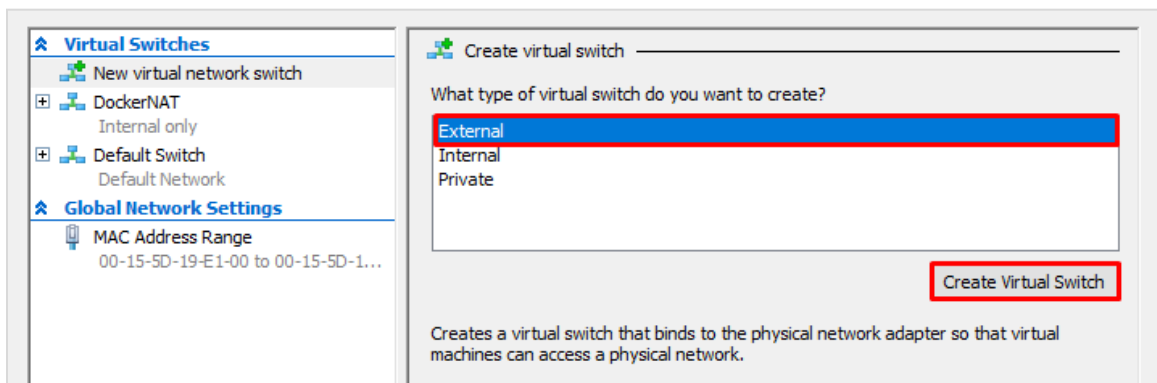
Virtual Machines				
Name	State	CPU Usage	Assigned Memory	
Windows 2019 Server	Off			
Denodo Quickstart Virtual Machine v2	Running	0%	4096 MB	
Checkpoints				
The selected virtual machine has no checkpoints.				
Denodo Quickstart Virtual Machine v2				
Adapter	Connection	IP Addresses		Status
Network Adapter (Dynam...	Default Switch	172.31.35.35; fe80::215:5...		OK

If your Virtual Machine does not have an IP, follow the steps below to create a network switch that can access your network. To that, please, follow these steps:

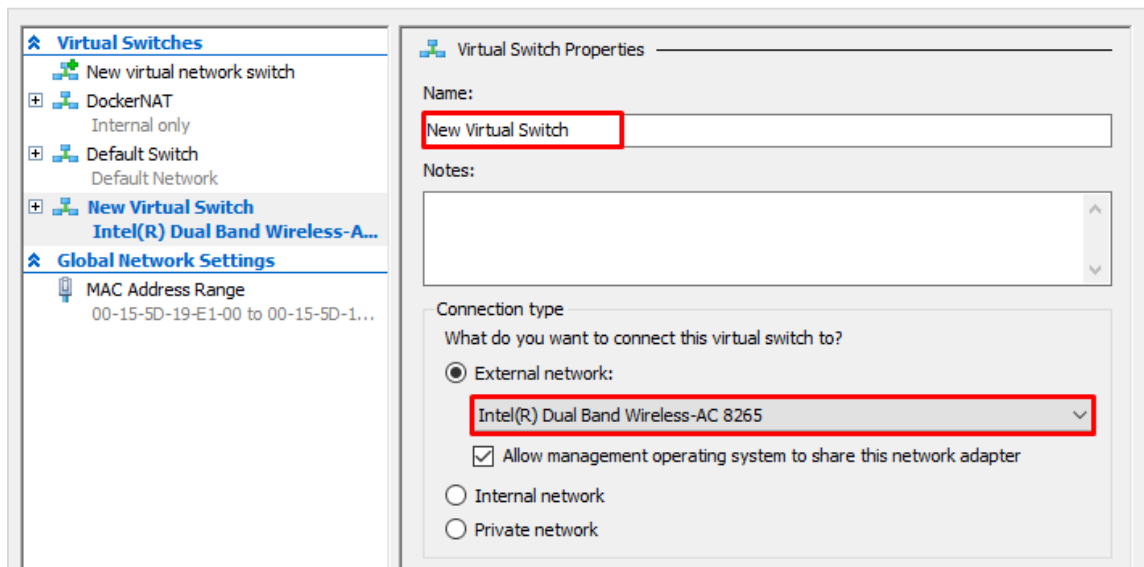
1. Launch your "Hyper-V Manager". Locate the option "Virtual Switch Manager..." under the menu Action.



2. In the “Virtual Switch Manager” window:
 - a. Click on “New virtual network switch”.
 - b. Select “External” as the type of the new virtual switch to create.
 - c. Finally, click on “Create Virtual Switch” to create it.

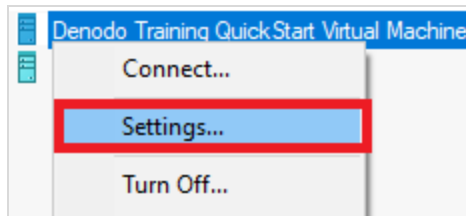


3. In the configuration window of the new virtual switch:
 - a. Give a name to this new virtual switch. In this example, we are going to name it “External Switch”.
 - b. Select “External network” as the “Connection type”, and select the network used by your computer to access the network.
 - c. When done, click on “Ok” to save the changes.

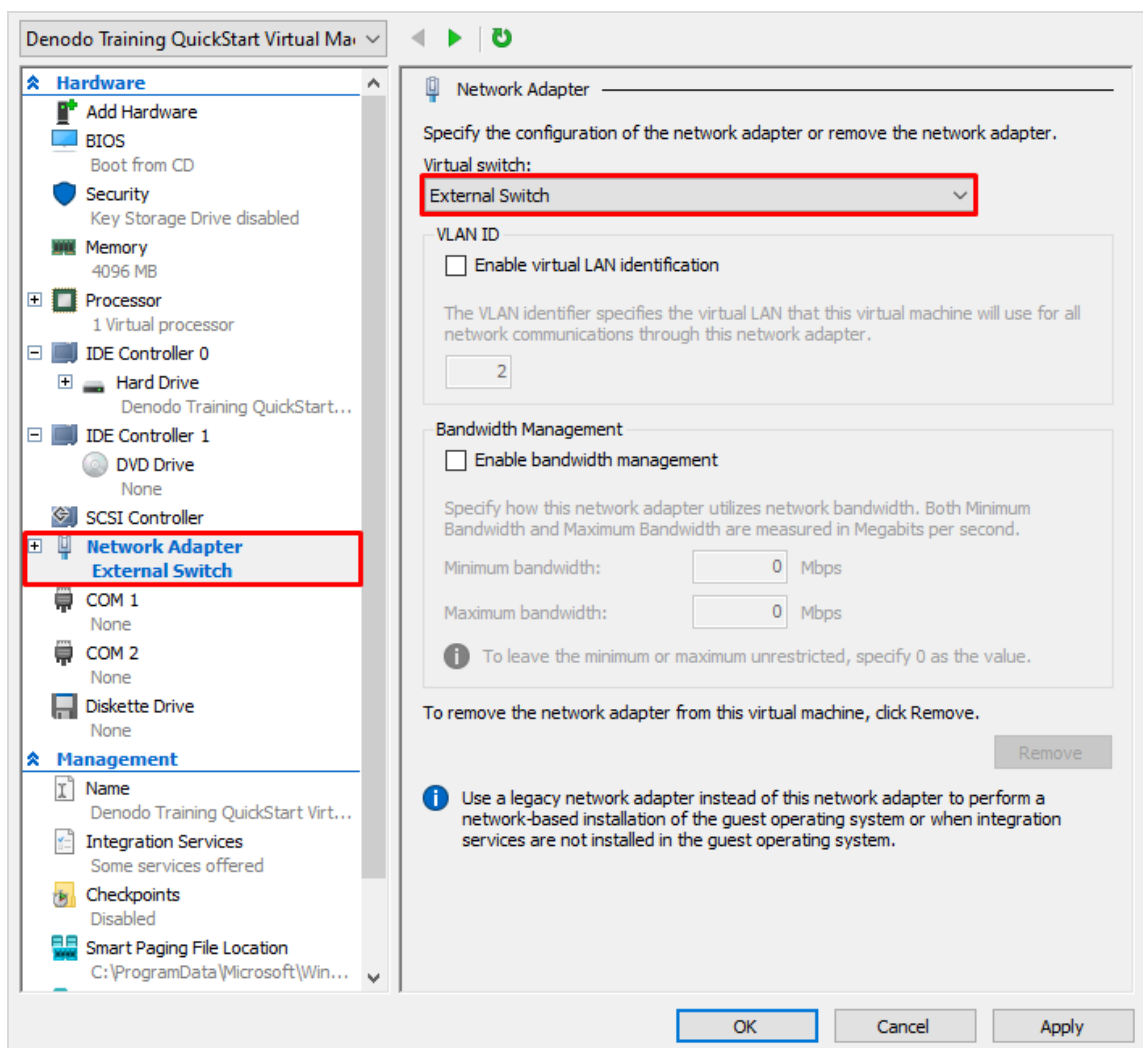


When done, you must configure your DQVM to use your new virtual switch. To do that, please, follow these steps:

1. Right click on the DQVM and select the option “Settings...”:



2. Click on the “Network Adapter” section, and select the new virtual Switch created previously (“External Switch” in this example). Click on “Ok” to save the changes.



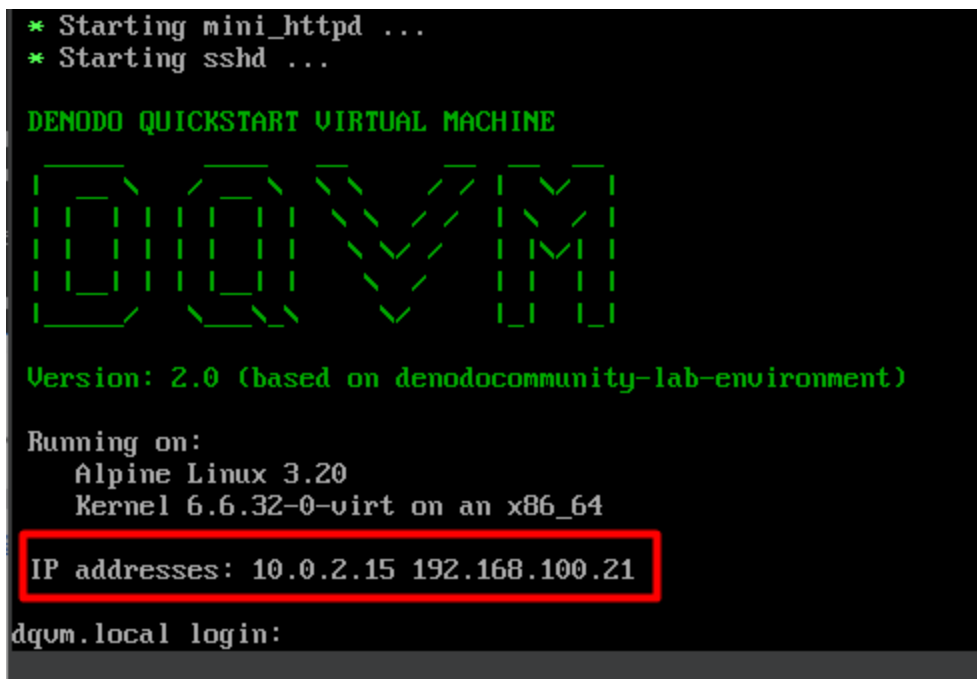
Post-configuration tasks

In this section, some additional configurations are explained. They are not needed for running the DQVM, but you can follow them to simplify the access to the virtual machine.

Getting the IP of the Virtual Machine

Let's see how you can obtain the **IP address** of the virtual machine.

1. When the DQVM is launched a new window is opened. In that window, you can see the startup progress.
2. At some point, you will be asked for login credentials. You can see the IP address in this window.



```
* Starting mini_httpd ...
* Starting sshd ...

DENODO QUICKSTART VIRTUAL MACHINE

DQVM

Version: 2.0 (based on denodocommunity-lab-environment)

Running on:
  Alpine Linux 3.20
  Kernel 6.6.32-0-virt on an x86_64

IP addresses: 10.0.2.15 192.168.100.21

dqvm.local login:
```

3. Please, note that the recommended IP Address to communicate with the VM is 192.168.100.X. If you see only one IP, please, use that to access the VM.

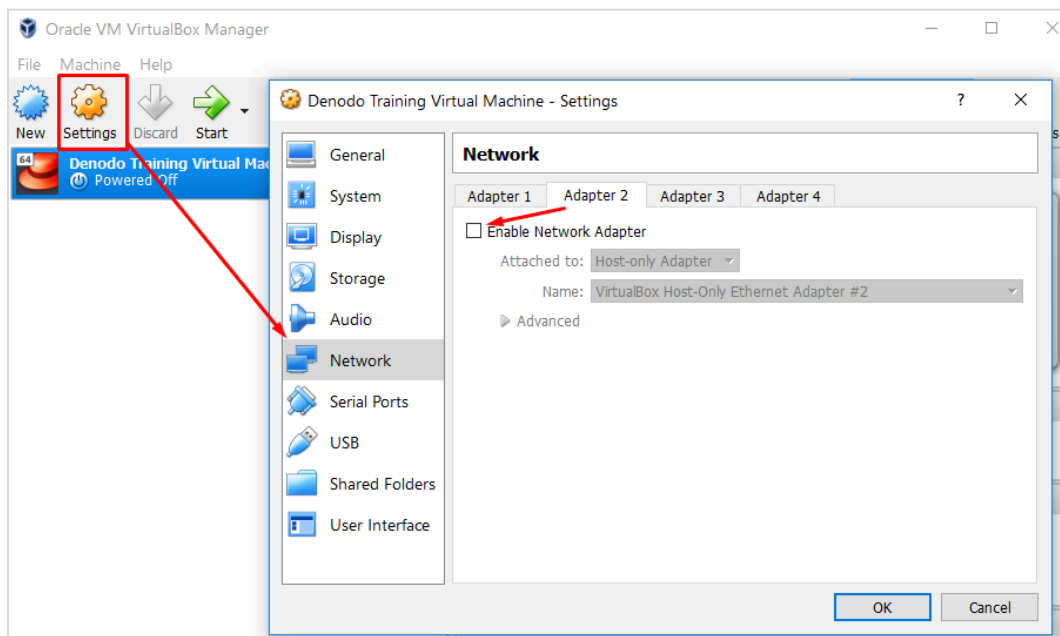
Note: If the field “IP Addresses” is empty after the DQVM startup, please, follow the steps listed in the section [Getting the IP from the command line](#) to get the IP manually.

IP without pre-configuration in Oracle VirtualBox

Although the recommendation for using the DQVM is to configure the network adapter to use the IP address 192.168.100.X, you can also use this virtual machine even without configuring the network adapter.

If you are not going to configure the network adapter, you need to change the network settings of the DQVM by removing the second adapter:

1. Open the **Settings** of the Virtual Machine.
2. Select the **Network** settings.
3. Remove or disable **Adapter 2**



4. Click OK to save the changes.

Getting the IP address from the command line

Another option to get the IP address is to login as root and execute a specific command. You can follow these steps:

- At some point, the console of the Virtual Machine will ask for login credentials. Use the “root” user: root / denodo

- After logged in, you can execute the following command to get the IP of the machine:
 - `ip addr`

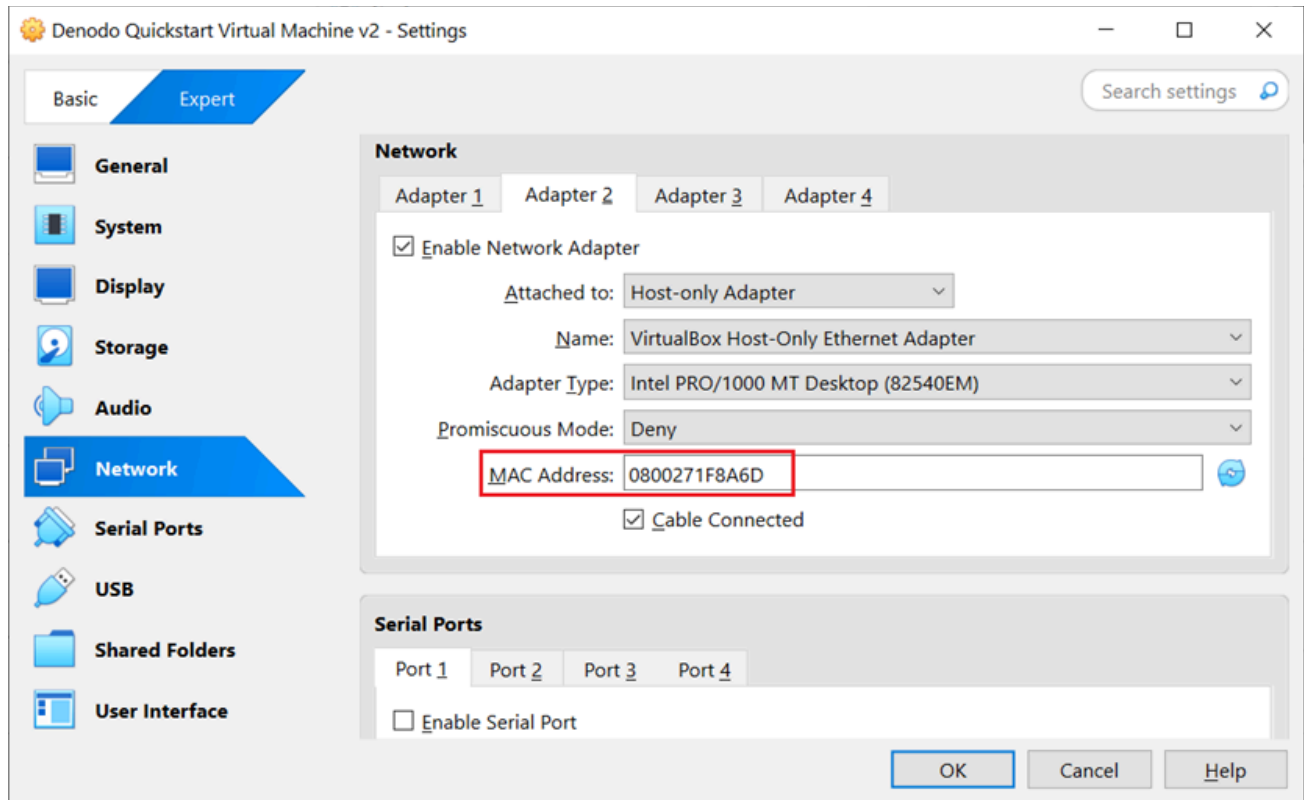
After executing that command, you will get the IP address that you can use for connecting to the data sources for the different training labs. You will get something like the following screenshot, in which you can see the IP (marked in red) you have to use for the labs:

```
dqvm:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 08:00:27:73:30:c3 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe73:30c3/64 scope link
        valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 08:00:27:00:4c:45 brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.21/24 scope global eth1
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe00:4c45/64 scope link
        valid_lft forever preferred_lft forever
4: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN
    link/ether 02:42:40:52:ca:0c brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
        valid_lft forever preferred_lft forever
```

Configure a static IP address for the DQVM

This section describes how to configure the DQVM with a static IP (instead of the default configuration which is DHCP, as we have seen previously).

After importing the DQVM image on your virtualization platform, by default, the DQVM will have two adapters. Adapter #2 is used to access the system externally (e.g. from the Denodo Platform). Find the MAC address of this adapter:



You can follow these steps to configure a static IP address:

1. Log in with the user: *root* / *denodo*

- For this, you can use the console on your virtualization platform. In this case, you may want to change the keyboard setting after you log in for the first time. See the chapter [Changing the Keyboard Layout](#).

- Or you can use an external ssh client:

```
# ssh <current-DQVM-external-IP> -l root
```

- Or Putty:

```
> putty.exe -ssh root@<current-DQVM-external-IP> -pw denodo
```

2. There is a network link related to the virtual machine adapter #2. You can see it from the MAC address (it is *eth1* in this example):

```
# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
```



```

        valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 08:00:27:26:84:28 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.7/24 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe26:8428/64 scope link
        valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 08:00:27:1f:8a:6d brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.148/24 scope global eth1
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe1f:8a6d/64 scope link
        valid_lft forever preferred_lft forever
4: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN
    link/ether 02:42:84:ee:16:54 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
        valid_lft forever preferred_lft forever

```

3. Edit the file `/etc/network/interfaces` and change the `eth1` configuration to:

```

auto eth1
iface eth1 inet static
    address 192.168.56.130
    netmask 255.255.255.0

```

(address/netmask above are examples. Apply your desired address and netmask (gateway and dns-nameservers don't need to be specified here, because this network link is only for incoming network traffic).

4. Optionally, you can add an entry to the `/etc/hosts` file:

```

192.168.56.130    data-server

```

5. Shut down the virtual machine:

```

# poweroff

```

6. When the virtual machine is started again, the console should now show the new IP on adapter #2:

```
DENODO QUICKSTART VIRTUAL MACHINE
DQVM
Version: 2.0 (based on denodocommunity-lab-environment)
Running on:
  Alpine Linux 3.20
  Kernel 6.6.33-0-virt on an x86_64
IP addresses: 10.0.2.7 192.168.56.130
dgvm.local login: _
```

Editing your local Hosts file for using data-server as hostname

You can edit the hosts file of your Operating System for mapping the IP address of the DQVM to “**data-server**” host name. By doing that, it will be easy for you to do the on demand labs, as you will just need to enter “data-server” for any data source connection instead of the whole IP address.

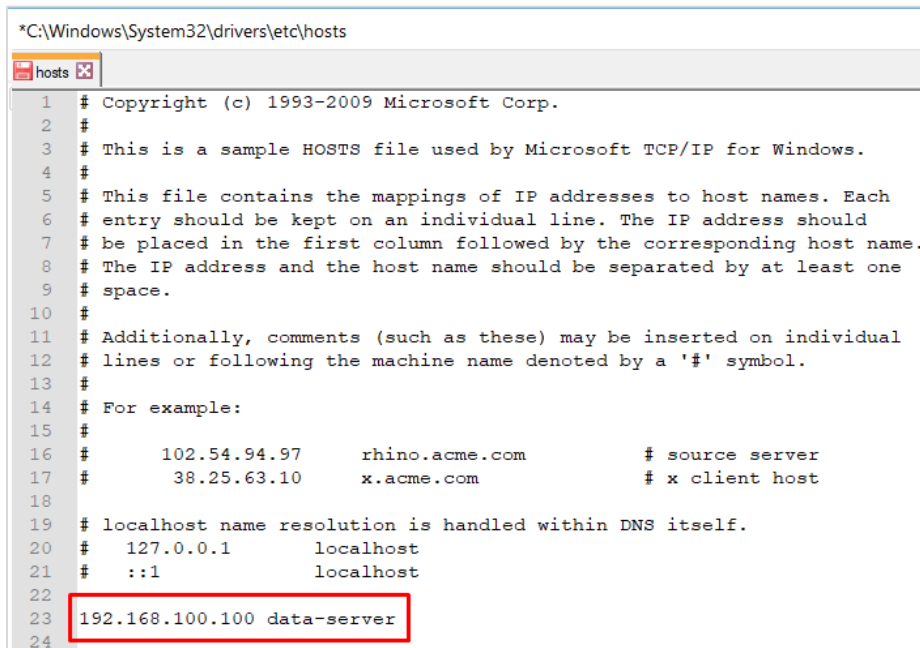
Note: The lab descriptions assume that you have the containers of the data sources running in localhost. When using the DQVM you have to use **data-server** or the **IP address** of the Virtual Machine instead of localhost.

For configuring that, you have to locate the hosts file in your system. Find below the instructions for Windows:

- The hosts file is located at C:\Windows\System32\drivers\etc

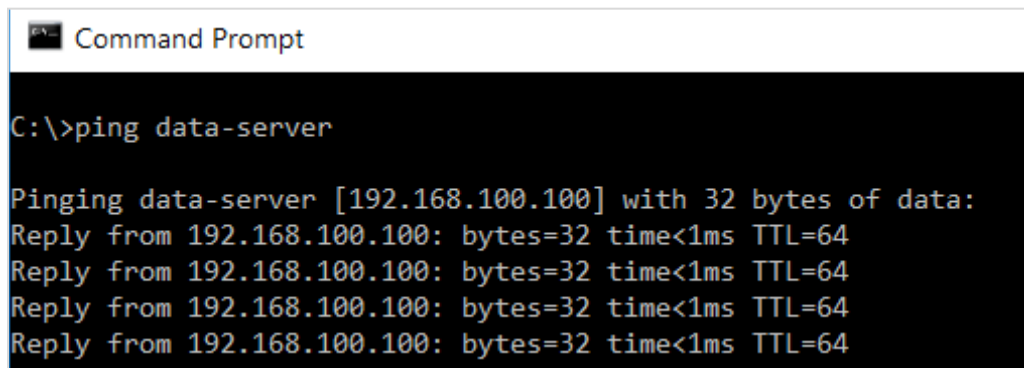
- a. Open the file “**hosts**” (you can use any text editor, just keep in mind you would need admin privileges for editing this file).
- b. Add a new line at the bottom of this file with the following (change 192.168.100.100 with the actual IP address of your DQVM):

192.168.100.100 data-server

A screenshot of a Windows text editor window showing the contents of the hosts file. The title bar indicates the file path is *C:\Windows\System32\drivers\etc\hosts. The file content includes copyright information, a sample HOSTS file description, and mappings for rhino.acme.com, x.acme.com, and localhost. The line '192.168.100.100 data-server' at the bottom is highlighted with a red rectangular box.

```
*C:\Windows\System32\drivers\etc\hosts
hosts
1 # Copyright (c) 1993-2009 Microsoft Corp.
2 #
3 # This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
4 #
5 # This file contains the mappings of IP addresses to host names. Each
6 # entry should be kept on an individual line. The IP address should
7 # be placed in the first column followed by the corresponding host name.
8 # The IP address and the host name should be separated by at least one
9 # space.
10 #
11 # Additionally, comments (such as these) may be inserted on individual
12 # lines or following the machine name denoted by a '#' symbol.
13 #
14 # For example:
15 #
16 #       102.54.94.97       rhino.acme.com       # source server
17 #       38.25.63.10       x.acme.com           # x client host
18
19 # localhost name resolution is handled within DNS itself.
20 #   127.0.0.1       localhost
21 #   ::1             localhost
22
23 192.168.100.100 data-server
24
```

- c. Save the file.
- d. You can confirm that this modification in the hosts file has been done by opening a command-line window and just doing a ping to “data-server”.

A screenshot of a Windows Command Prompt window. The title bar says 'Command Prompt'. The command 'C:\>ping data-server' has been entered. The output shows four successful ping replies from 192.168.100.100 with a time of less than 1ms and TTL of 64.

```
Command Prompt
C:\>ping data-server

Pinging data-server [192.168.100.100] with 32 bytes of data:
Reply from 192.168.100.100: bytes=32 time<1ms TTL=64
Reply from 192.168.100.100: bytes=32 time<1ms TTL=64
Reply from 192.168.100.100: bytes=32 time<1ms TTL=64
Reply from 192.168.100.100: bytes=32 time<1ms TTL=64
```

Accessing the Virtual Machine

User Authentication

You can access this virtual Machine using this user:

- **Login:** denodo
- **Password:** denodo

Please, note that the default “denodo” user is included in the sudoers users, so for executing commands as root you can simply use `sudo ...`

Accessing as “root”

You can access the DQVM by using the “root” user, for example if you need to restart a service or update any configuration. To access “root”, please, follow these steps:

- When the DQVM is launched a new window is opened. In that window you can see the startup progress.
- At some point you will be asked for login credentials. Use the “root” user:
 - **Login:** root
 - **Password:** denodo

How to Access

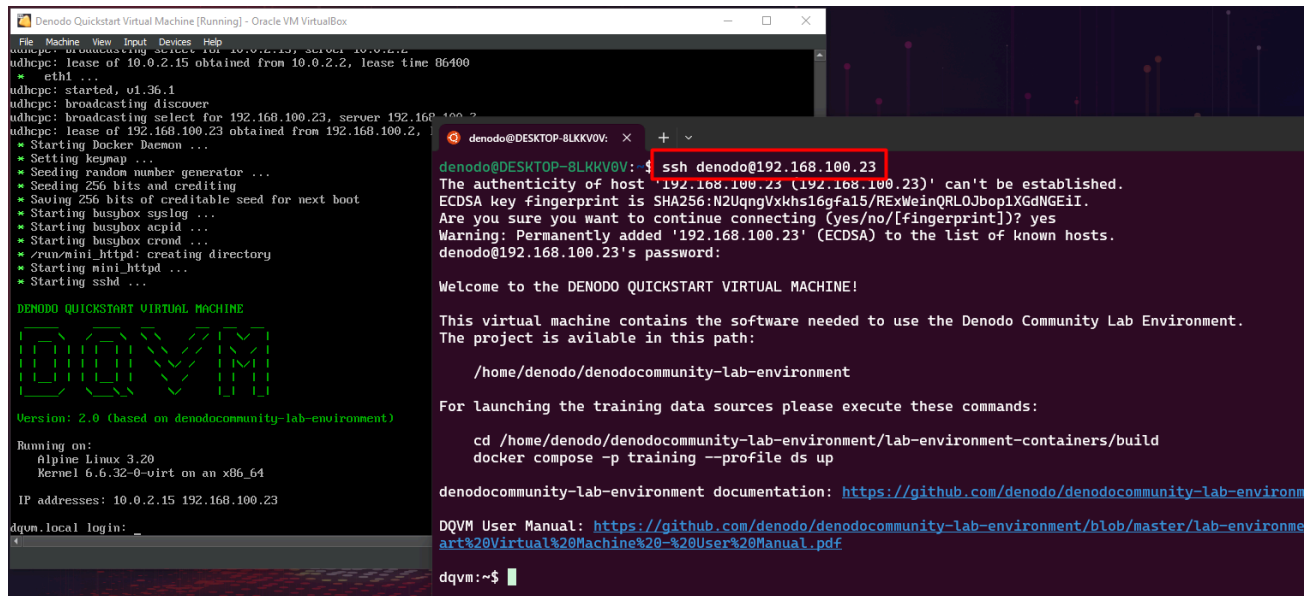
You can access the Virtual Machine using one of these methods.

Using a SSH Client (RECOMMENDED)

Another option for connecting to the Virtual Machine is to use a SSH client. This option is the **RECOMMENDED** one, because you may not have direct access to the console if, for example, the virtual machine is running on a different machine.

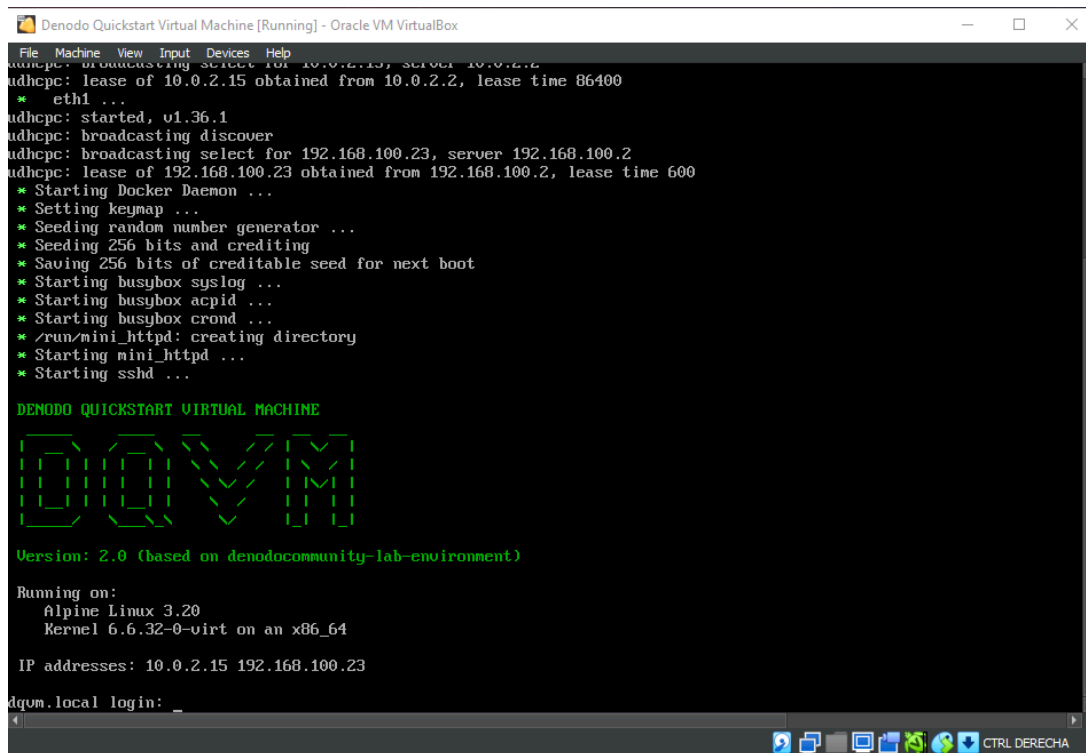
For connecting using SSH, you need a ssh client installed in your system. As an example, you can use Linux on Windows with WSL (Windows Subsystem for Linux) and execute the command:

```
$ ssh denodo@<IP address of VM or data-server>
```



Using the Virtual Machine Console

When the Virtual Machine is started in Virtual Box or Hyper-V, you will see that a new window is opened. In this console you can login to work with the virtual machine:



Accessing Through a Web Interface

The DQVM includes a web-based system administration tool called **Webmin**. Webmin is a web-based server management control panel for Unix-like systems allowing users to configure the virtual machine internals, such as users, services, etc.

In order to use the Webmin web console, you have to enable it executing this command:

- `sudo /usr/bin/expect /home/denodo/conf/webmin.exp`

```
*****
Webmin has been installed and started successfully.

Since Webmin was installed outside the package manager, ensure the
following recommended Perl modules and packages are present:
Perl modules:
- DateTime, DateTime::Locale, DateTime::TimeZone, Data::Dumper
- Digest::MD5, Digest::SHA, Encode::Detect, File::Basename
- File::Path, Net::SSLeay, Time::HiRes, Time::Local, Time::Piece
- lib, open
Packages:
- openssl - Cryptography library with TLS implementation
- shared-mime-info - Shared MIME information database
- tar gzip unzip - File compression and packaging utilities

Use your web browser to go to the following URL and login
with the name and password you entered previously:

  https://dqvm.local:10000

Because Webmin uses SSL for encryption only, the certificate
it uses is not signed by one of the recognized CAs such as
Verisign. When you first connect to the Webmin server, your
browser will ask you if you want to accept the certificate
presented, as it does not recognize the CA. Say yes.

dqvm:~$
```

This installation is only needed the first time you are launching Webmin. If you restart the virtual machine, you only have to start the service executing the following command:

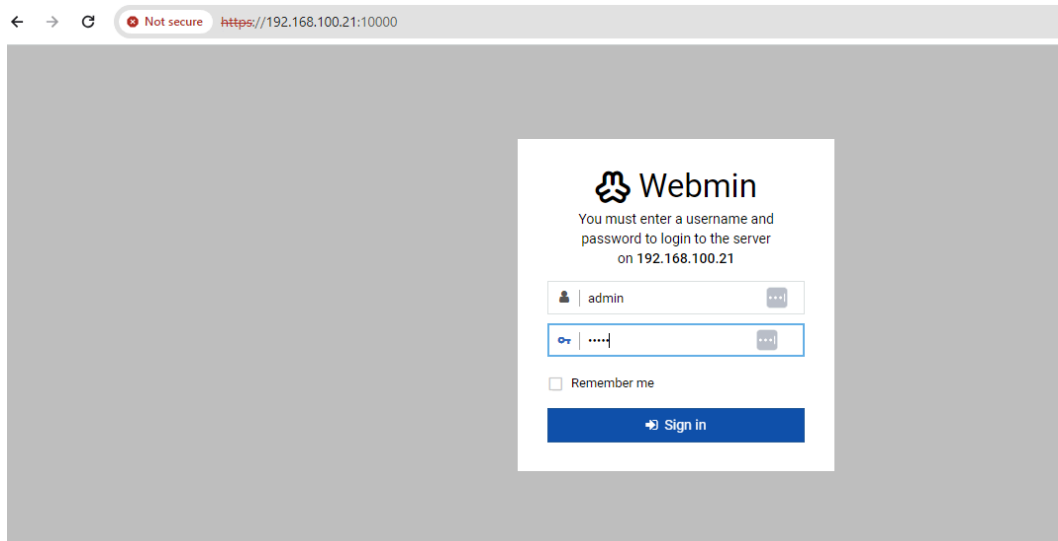
- `sudo rc-service webmin start`

```
dqvm:~$ sudo rc-service webmin start
Starting Webmin server in /opt/webmin-2.105
```

Once it is started, you can access the console from your browser in the URL:

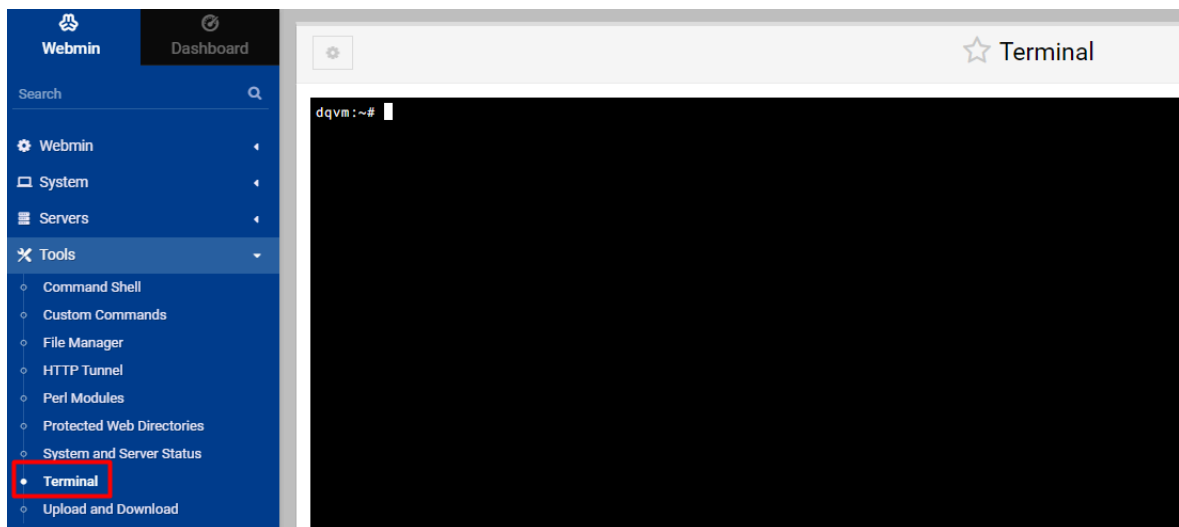
- `https://<IP address of VM or data-server>:10000`

Once the Webmin web console is launched you need to enter the username and password as **admin/admin**

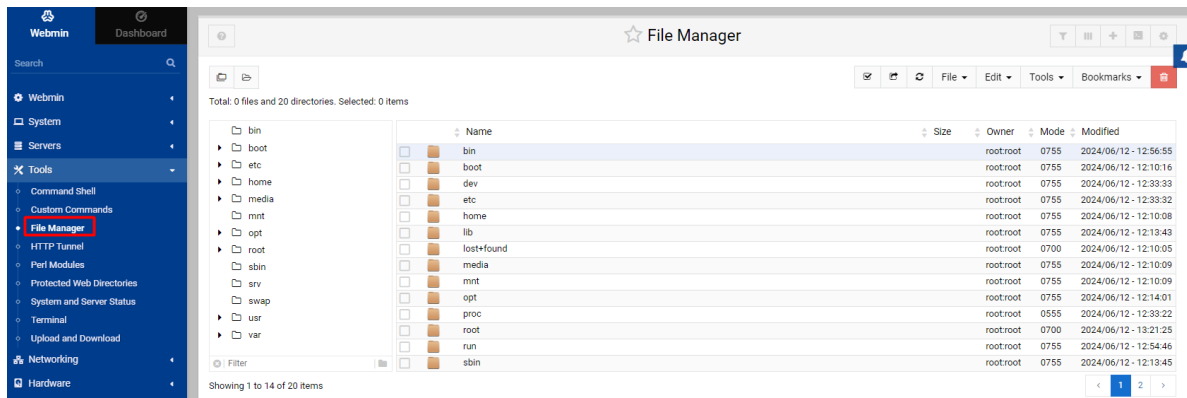


Once the login is successful, you can find a list of useful options available under the “Tools” menu such as:

1. **Terminal** - This module allows you to enter shell commands. Please note this shell uses the user root!



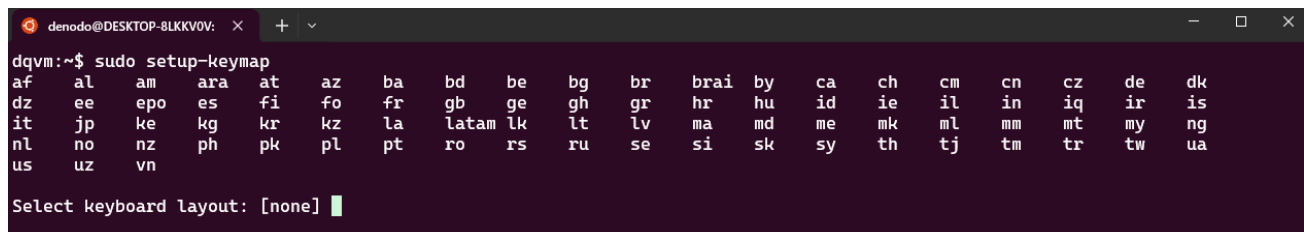
2. **File Manager** - It allows you to view and manipulate files on the DQVM through an HTML interface. You can upload files to the DQVM (or download files to your system) using this tool.



Changing the Keyboard Layout

By default, the DQVM is configured for the **US keyboard**. To change the keyboard layout of the DQVM, access to the Virtual Machine and execute the following:

- `sudo setup-keymap`



A list of keyboard layouts will be shown. You can select the most appropriate for your system. For example, to change the layout to Spanish, you can select:

- Select keyboard layout: es
- Select variant: es

Launching the Training Data Sources

Important: The DQVM requires external connectivity to download and configure the data sources.

The Denodo Community Lab Environment is a Docker Compose script that comprises different environment variables that can be modified based on the requirement.

Once you are connected to the Virtual Machine, you only have to run the following commands for getting the latest version of the Denodo Community Lab Environment and launch the data sources:

```
cd /home/denodo/denodocommunity-lab-environment/  
git pull  
cd lab-environment-containers/build/  
sudo docker compose -p training --profile ds up -d
```

where:

- **-p training** refers to the docker project name. This parameter is optional. If you use it, you can use any label here. This is useful if you want to have multiple versions of the same lab environment. For example, one for training and another one for creating a personal project. For the rest of examples used here we are going to omit this parameter but it is strongly recommended to use it.
- **--profile ds** refers to the profile name. The profiles help you to selectively start services. This is achieved by assigning each service to one or more profiles.
- **up** refers to the command for building, creating and starting containers for the services configured. It aggregates the output of each container (logs), if you want to avoid that, simply add **--detach** or **-d** to start the containers in the background.

Note: it is important to execute the `git pull` command before launching the data sources for downloading the latest updates of the Lab Environment (to use the latest version of the training datasets)

The first time you execute this command, it will download all the required containers from dockerhub:

```
dqvm:/home/denodo/denodocommunity-lab-environment/lab-environment-containers/build# docker compose -p training --profile ds up
WARN[0000] /home/denodo/denodocommunity-lab-environment/lab-environment-containers/build/docker-compose.yml: 'version' is obsolete
[+] Running 5/24
  ⬇ ds-mariadb [ ] Pulling                                14.0s
  ⬇ ds-mongo [ ] Pulling                                  14.0s
  ⬇ ext-apacheds [#####] Pulling                         14.0s
  ⬇ ds-httpd [ ] Pulling                                  14.0s
  ⬇ ds-postgresql [ ] Pulling                             14.0s
  ⬇ ds-tomcat [ ] Pulling                                 14.0s
  ⬇ util-graphql-playground [ ] Pulling                   14.0s
  ⬇ denodo-postgres Pulling                               14.0s
```

That step is only done once. When completed, it will start all the containers of the data sources (you can see the output in the console):

```
denodocommunity-lab-environment-postgres | 2024-06-12 13:48:18.670 UTC [36] LOG: aborting any active transactions
denodocommunity-lab-environment-postgres | 2024-06-12 13:48:18.681 UTC [36] LOG: background worker "logical replication launcher" (PID 43) exited with
exit code 1
denodocommunity-lab-environment-postgres | 2024-06-12 13:48:18.685 UTC [38] LOG: shutting down
denodocommunity-lab-environment-postgres | 2024-06-12 13:48:18.937 UTC [36] LOG: database system is shut down
denodocommunity-lab-environment-postgres | done
denodocommunity-lab-environment-postgres | server stopped
denodocommunity-lab-environment-postgres | PostgreSQL init process complete; ready for start up.
denodocommunity-lab-environment-postgres | 2024-06-12 13:48:18.995 UTC [1] LOG: starting PostgreSQL 12.19 on x86_64-pc-linux-musl, compiled by gcc (Alp
line 13.2.1.git28248389) 13.2.1 20240309, 64-bit
denodocommunity-lab-environment-postgres | 2024-06-12 13:48:18.995 UTC [1] LOG: listening on IPv4 address "0.0.0.0", port 5432
denodocommunity-lab-environment-postgres | 2024-06-12 13:48:18.995 UTC [1] LOG: listening on IPv6 address "::", port 5432
denodocommunity-lab-environment-postgres | 2024-06-12 13:48:19.008 UTC [1] LOG: listening on Unix socket "/var/run/postgresql/.s.PGSQL.5432"
denodocommunity-lab-environment-postgres | 2024-06-12 13:48:19.843 UTC [50] LOG: database system was shut down at 2024-06-12 13:48:18 UTC
denodocommunity-lab-environment-postgres | 2024-06-12 13:48:19.858 UTC [1] LOG: database system is ready to accept connections
denodocommunity-lab-environment-postgres | ==> start apache server for use....
denodocommunity-lab-environment-apacheds |
denodocommunity-lab-environment-mongo | WARNING: MongoDB 5.0+ requires a CPU with AVX support, and your current system does not appear to have that!
denodocommunity-lab-environment-mongo | see https://jira.mongodb.org/browse/SERVER-54407
denodocommunity-lab-environment-mongo | see also https://www.mongodb.com/community/forums/t/mongodb-5-0-cpu-intel-g4650-compatibility/116610/2
denodocommunity-lab-environment-mongo | see also https://github.com/docker-library/mongo/issues/485#issuecomment-891991814
denodocommunity-lab-environment-mongo | /usr/local/bin/docker-entrypoint.sh: line 416: 25 Illegal instruction "${mongodHackedArgs[@]}" --fork
```

List of Data Sources Included in the Virtual Machine

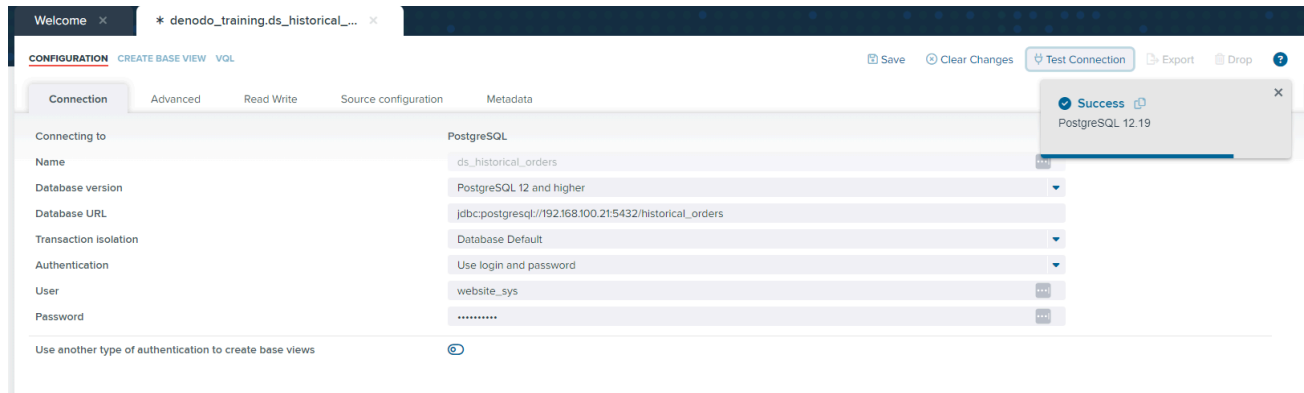
Please check the Denodo Community Lab Environment documentation to see the list of supported data sources:

<https://github.com/denodo/denodocommunity-lab-environment/tree/master/lab-environment-containers#list-of-denodo-common-lab-environment-containers>

Connecting from Denodo to the Training Data Sources

Local Installation of Denodo Platform

If you have Denodo Platform installed locally in your laptop, once the Denodo Community Lab Environment is running in the DQVM, you will be able to access them using the **IP address of the DQVM** (or using **data-server**, if you have configured your system to use that hostname, see [Editing your local Hosts file for using data-server as hostname](#)).



You can find the exact steps for creating the data sources in the lab description of the on-demand training courses or tutorials.

Running Denodo Platform in the DQVM

The Denodo Community Lab Environment allows the launch of the Denodo servers as containers inside the virtual machine. For doing that you need to upload your Denodo License to the Virtual Machine and configure the path to your license:

```
cd /home/denodo/denodocommunity-lab-environment/lab-environment-containers/build/  
vi .env
```

```
#####  
# Denodo #  
#####  
# For Denodo Customers:    DENODO_VERSION=harbor.open.denodo.com/denodo-9/images/denodo-platform:latest  
# For Denodo Express Users: DENODO_VERSION=harbor.open.denodo.com/denodo-express/denodo-platform:latest  
DENODO_VERSION=harbor.open.denodo.com/denodo-express/denodo-platform:latest  
# For Denodo Customers:    DENODO_AI_SDK_VERSION=harbor.open.denodo.com/denodo-connects-9/images/ai-sdk:latest  
# For Denodo Express Users: DENODO_AI_SDK_VERSION=harbor.open.denodo.com/denodo-express/ai-sdk:latest  
DENODO_AI_SDK_VERSION=harbor.open.denodo.com/denodo-express/ai-sdk:latest  
# Full Path to your Denodo Standalone license file  
DENODO_SA_LIC=../res/denodo/license/
```

You have to configure the path modifying the `DENODO_SA_LIC` parameter. You have to enter the full path to your license file. For example,

```
DENODO_SA_LIC=/home/denodo/denodo.lic
```

And finally, execute

```
sudo docker compose -p training --profile denodo --profile ds up -d
```

to start the data sources and the Denodo containers.

In this case you will have to connect to the Denodo web applications from your browser using the **IP address of the DQVM** (or using data-server, if you have configured your system to use that hostname), for example,

<http://data-server:19090/denodo-design-studio/?uri=%2F%2Fvdp:9999%2F#/> to connect to the Design Studio.

For connecting from Denodo to the data sources, you will always have to use the hostname of each container as all of them are running as containers inside the same internal network. You can check the list of hostnames in this table:

<https://github.com/denodo/denodocommunity-lab-environment/blob/master/lab-environment-containers/README.md#list-of-denodo-common-lab-environment-containers> (check the **hostname** column)

Running the Denodo AI SDK container

In case you want to try the **Denodo AI SDK**, please read the [Building an AI Chatbot with Denodo](#) tutorial, which explains how to use a specific environment with some pre-created views in Denodo to be used by the AI SDK and the Sample Chatbot.

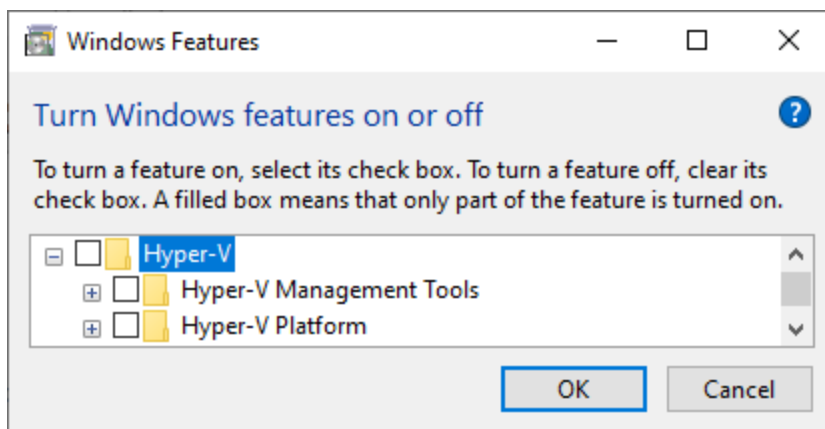
Appendix: Common Issues

The message “VT-x is not available (VERR_VMX_NO_VMX)” is shown when trying to start the DQVM in VirtualBox.

Please, note that you have to enable the Virtualization option in your local machine BIOS. Check the manuals of your BIOS vendor, because this option can have different names based on the BIOS vendor.

If you have enabled the Virtualization option in your local machine BIOS, please, check if you have disabled the “Hyper-V” option. To check that, you can:

- Check it in the Windows Features (Windows button + r, type “OptionalFeatures.exe” and click Ok). Be sure that the Hyper-V options are disabled.



- Or you can open an administrator command console and execute the following command:
 - `bcdedit /set hypervisorlaunchtype off`

Please, note that you have to restart the machine to apply this change. If you have enabled the Hyper-V options, note that you can use the “Microsoft Hyper-V” version of the DQVM.

The message “This kernel requires an x86-64 cpu but only detected an i686 cpu.” is shown when I try to start the DQVM in VirtualBox.

In VirtualBox, ensure that you have selected “Red Hat (64-bit)” as the version of the DQVM. To do that:

1. In VirtualBox, select the DQVM and click on “Settings”.

2. Navigate to the “General Settings” window > “Basic” tab.
3. Ensure that the “Version” is “Red Hat (64-bit)”.

The message “Failed to load R0 module C:\Program Files\Oracle\VirtualBox/VMMR0.r0: SUP_IOCTL_LDR_OPEN failed (VERR_LDR_GENERAL_FAILURE).” is shown when I try to start the DQVM in VirtualBox.

Your VirtualBox installation is affected by the bug [#20627](#). The Windows Hypervisor-enforced Code Integrity (HVCI) feature rejects the VirtualBox component VMMR0.r0 (*).

Ensure to use the latest version of VirtualBox or one of the tested versions of VirtualBox.

A workaround is to disable HVCI aka Memory integrity as follows:

1. On your Windows host, go to Start > Settings > Update & security > Windows Security > Device security > Core isolation details.
2. Turn “off” Memory integrity.
3. Reboot the Windows host.

**The Mongo container fails to start
“/usr/local/bin/docker-entrypoint.sh: line 416: 25 Illegal instruction.
MongoDB 5.0+ requires a CPU with AVX support”**

The CPU used by the virtualization system does not have the AVX extension, which is required for the MongoDB 7.0 data source.

If you are receiving this message, you should change the MongoDB version to 4.4.18 following these steps:

1. Login to the DQVM
2. Stop the data sources:
`~$ sudo docker compose -p training --profile ds down`

3. Edit the .env file of the Lab Environment:

```
~$ cd /home/denodo/denodocommunity-lab-environment/lab-environment-containers/build  
~$ vi .env
```

4. Change the entry NOSQL_DB_VERSION=mongo:7.0.8 to:

```
NOSQL_DB_VERSION=mongo:4.4.18
```

5. Save the .env file

6. Start the data sources again:

```
~$ sudo docker compose -p training --profile ds up -d
```

Thank you again for your continued trust in Denodo, and we look forward to providing you with the industry's best products and services over the coming year. Should you have any feedback or questions, please let us know.

NOTE: *This document is confidential and proprietary of Denodo. No part of this document may be reproduced in any form by any means without prior written authorization of Denodo.*

Visit www.denodo.com Email info@denodo.com twitter.com/denodo
NA & APAC (+1) 877 556 2531 | EMEA (+44) (0) 20 7869 8053 | DACH (+49) (0) 89 203 006 441
Iberia & Latin America (+34) 912 77 58 55

Denodo Training QuickStart Virtual Machine