



Denodo Platform Container QuickStart Guide

Revision 20191021

NOTE

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

Copyright © 2020
Denodo Technologies Proprietary and Confidential

[Running Virtual DataPort in a Docker container](#)

[Pre-requisites](#)

Denodo North America & APAC: 525 University Avenue, Suite 31, Palo Alto, CA 94301. USA
Denodo Iberia & Latino América: Montalbán 5, 28014 Madrid, Spain
Denodo EMEA: 21st Floor, Portland House, Bressenden Place, London SW1E 5RS. UK
Denodo DACH: Karlstraße 10, 80333 München. Germany

www.denodo.com

[Installation of Docker](#)

[Loading the image](#)

[Running the image](#)

[Standalone Licenses](#)

[Licenses Managed by a Denodo Solution Manager](#)

[Appendix](#)

[Stopping containers](#)

[Loading metadata](#)

[Saving the container state](#)

[Connecting with the VDP Administration Tool](#)

[Starting additional tools](#)

[Changing Denodo ports](#)

[Changing Denodo settings](#)

[Restricting the number of cores](#)

[Configuring the network](#)

1 RUNNING VIRTUAL DATAPORT IN A DOCKER CONTAINER

1.1 PRE-REQUISITES

1. A compatible OS upon which Docker CE/EE version 18.03+ can be installed.
2. Denodo Platform Container image in tar format.
3. A valid Denodo Platform license, either standalone or managed by an existing Denodo Solution Manager.

1.2 INSTALLATION OF DOCKER

Follow the instructions corresponding to the Operating System that you are using from the [official Docker website](#). For example, the instructions for Ubuntu are available [here](#). Once Docker is installed, you can verify its version by executing the following command from a terminal:

```
$ docker version
Client:
Version:      18.03.1-ce
API version:  1.37
Go version:   go1.9.5
Git commit:   9ee9f40
Built:        Thu Apr 26 07:12:48 2018
OS/Arch:      windows/amd64
Experimental: false
Orchestrator: swarm

Server:
Engine:
Version:      18.03.1-ce
API version:  1.37 (minimum version 1.12)
Go version:   go1.9.5
Git commit:   9ee9f40
Built:        Thu Apr 26 07:22:38 2018
OS/Arch:      linux/amd64
Experimental: true
```

1.3 LOADING THE IMAGE

Once you have downloaded the `denodo-container-7.0-xxxxxxx.zip` file you need to uncompress it to get the Denodo Platform container image in tar format that you can load in Docker with the [docker load](#) command.

For example, you can load the `denodo-platform-latest.tar` file by executing the following command:

```
$ docker load -i denodo-platform-latest.tar
Loaded image: denodo-platform:7.0-latest
```

1.4 RUNNING THE IMAGE

The Denodo Platform container image can be used either with Standalone Denodo Licenses (like the Denodo Express License or an Evaluation License) or with licenses managed by a Denodo Solution Manager.

1.4.1 Standalone Licenses

If you are using a Standalone Denodo License (like the Denodo Express License or an Evaluation License) you have to download the license file and place it in a location in the host OS. You can then run the image by simply executing the following command:

```
$ docker run -d -h <hostname> -p 9999:9999 -p 9997:9997 -p 9996:9996  
-p 9090:9090 -v <path>:/opt/denodo/conf/denodo.lic --rm --name denodo-  
vdpserver denodo-platform:7.0-latest ./denodo-container-start.sh  
--vdpserver
```

where:

- d
Optional. Runs the container in background and print container ID.
- h <hostname>
Sets the internal name for the new container. You need to use the hostname value that you will use to connect to the VDP Server running in this container, and **that hostname must be resolved and reachable from client applications**. You can use localhost to simplify connections from the host only.
- p 9999:9999 -p 9997:9997 -p 9996:9996 -p 9090:9090
Publishes the container ports. The following default ports used by the VDP server are published to the host: 9999, 9997, 9996, 9090. If you want to use a different configuration, read the “Changing Denodo ports” section below.
- v <path>:/opt/denodo/conf/denodo.lic
Specifies the Standalone Denodo License file to use by the Denodo Server, where:
 - <path> - Path for the license file in the host OS, for example:
/home/denodo/denodo.lic
 - /opt/denodo/conf/denodo.lic - Path for the license file in the container (Do not change this)
- rm
Optional. This flag tells Docker to automatically clean up the container and remove the file system when the container exits.
- name denodo-vdpserver
Optional. Name of the container to reference it from your host system.

1.4.2 Licenses Managed by a Denodo Solution Manager

If you are using a Denodo Solution Manager to manage your licenses, you just need to [create a new server](#) in the Solution Manager that will correspond to the VDP Server running in the container.

When registering the VDP Server in the Solution Manager admin tool, it is important that the **host** field matches the hostname that you will assign to the container and the **port** field points to 9999 (the default port that the VDP Server uses inside the container).

You need to create a [Solution Manager configuration file](#) in a location in the host OS. This configuration file must detail the host and port values for the License Manager. The host needs to be resolved by the container to the IP address where the License Manager is running and the port specifies where the License Manager is listening (by default this port is 10091). The container needs to have connectivity to that IP address and port number.

If your Solution Manager is running in a host with name `solution-manager` with the default port configuration, the corresponding `SolutionManager.properties` file is:

```
com.denodo.license.host=solution-manager
com.denodo.license.port=10091
```

Once the configuration file has been created you can run VDP server in a container with the following command:

```
$ docker run -d -h <hostname> -p 9999:9999 -p 9997:9997 -p 9996:9996
-p 9090:9090 -v <path>:/opt/denodo/conf/SolutionManager.properties
--rm --name denodo-vdpserver denodo-platform:7.0-latest ./denodo-
container-start.sh --vdpserver
```

where:

- d
Optional. Runs the container in background and print container ID.
- h <hostname>
Specifies the hostname for the container. This name has to match the one used in the Denodo Solution Manager definition for this VDP Server, and **that hostname must be resolved and reachable from client applications.**
- p 9999:9999 -p 9997:9997 -p 9996:9996 -p 9090:9090
Publishes the container ports. The following default ports used by the VDP server are published to the host: 9999, 9997, 9996, 9090. If you want to use a different configuration, read the “Changing Denodo ports” section below.
- v <path>:/opt/denodo/conf/SolutionManager.properties
Specifies the Solution Manager configuration file to use by the Denodo Server, where:
 <path> - Path for the Solution Manager configuration file in the host OS, for example:
 /home/denodo/SolutionManager.properties
 /opt/denodo/conf/SolutionManager.properties - Path for the SolutionManager.properties file in the container (Do not change this)
- rm
Optional. This flag tells Docker to automatically clean up the container and remove the file system when the container exits.
- name denodo-vdpserver
Optional. Name of the container to reference it from your host system.

2 APPENDIX

2.1 STOPPING CONTAINERS

It is advisable to stop the containers gracefully by using the [docker stop](#) command instead of the [docker kill](#) command. This ensures that running services are stopped and acquired resources freed before stopping the container. This is of particular importance when VDP Server uses a license managed by a Denodo Solution Manager since the license will get locked during a grace period unless the VDP Server releases the license during the proper shutdown. The command to stop the container gracefully is:

```
$ docker stop denodo-vdpserver
```

Where denodo-vdpserver is the name that you assigned to the container.

2.2 LOADING METADATA

Once the container is running you can import metadata in it in two ways:

- copying a VQL export file by using the [docker cp](#) command and execute the [import.sh](#) scripts located in the \$DENODO_HOME/bin directory with the [docker exec](#) command, or
- importing the VQL export file as you would normally do using the VDP Administration Tool's "File > Import" option.

Example of using the command line for loading metadata:

```
$ docker cp metadata.vql denodo-vdpserver:/opt/denodo
$ docker exec denodo-vdpserver ./bin/import.sh --singleuser -f
metadata.vql -h localhost/admin?admin@admin
```

Once you have imported the VQL you can save the container state to persist the changes to the container.

2.3 SAVING THE CONTAINER STATE

If you would like to persist the metadata changes in the container you can use Docker volumes or you can create new Docker images by committing the changes using the [docker commit](#) command.

You can issue the following command to commit:

```
$ docker commit denodo-vdpserver [repository:tag]
```

If your container does not have a name then you need first to get the container id by running `docker ps` and use it instead of the name. If you want to overwrite the original image you can use the same repository:tag which is used to run the container. If you do not provide a repository:tag value, an unnamed image will be created.

2.4 CONNECTING WITH THE VDP ADMINISTRATION TOOL

It is possible to connect to a VDP Server running in a container with a VDP Administration Tool running in the host or in a separate computer, but you must ensure that the computer where you want to run the VDP Administration Tool is able to resolve the name that you assigned to the Docker container (either via DNS or hosts file).

Using the following default connection settings:

```
Login:      admin
Password:   admin
Server URI: //<hostname>:9999/admin
```

Where 9999 represents the published (external) port of the VDP Server container.

2.5 STARTING ADDITIONAL TOOLS

The Denodo container image comes pre-configured to allow you to start the Denodo Data Catalog and/or the Denodo Diagnostic & Monitoring Tool in the same container where you are running the VDP Server.

If you want to start any of these extra tools use the command:

```
$ docker run -d -h <hostname> -p 9999:9999 -p 9997:9997 -p 9996:9996
-p 9090:9090 -v <path>:/opt/denodo/conf/SolutionManager.properties
--rm --name denodo-vdpserver denodo-platform:7.0-latest ./denodo-
container-start.sh --vdpserver --datacatalog --dmt
```

where:

- datacatalog
Optional. Starts the Denodo Data Catalog web application.
- dmt
Optional. Starts the Denodo Diagnostic & Monitoring Tool.

You also can run these two additional tools with a Standalone Denodo License.

These additional applications require a local VDP Server to run, so you can not start them in a separate container without also including a VDP Server.

2.6 CHANGING DENODO PORTS

The Docker commands documented previously where publishing the same ports in the host than the ports used by Denodo Platform inside the container. This was defined with the docker arguments:

```
-p 9999:9999 -p 9997:9997 -p 9996:9996 -p 9090:9090
```

These default ports have the following use in Denodo Platform:

- 9999 - RMI Registry (JDBC, VDP Admin Tool...)
- 9997 - RMI Factory (JDBC, VDP Admin Tool...)
- 9996 - ODBC
- 9090 - Web Container

When running several Denodo Platform containers in the same host these published ports must be different for each container. You can change any of these ports replacing the previous docker argument with the following one:

```
-e FACTORY_PORT=<factory_port> -p <factory_port>:<factory_port> -p <registry_port>:9999 -p <odbc_port>:9996 -p <web_port>:9090
```

Where:

<factory_port>

Represents the host port number that will be used as the RMI factory port required by the JDBC driver. This same port will be used in the host and also inside the container.

<registry_port>, <odbc_port>, <web_port>

Represents the host port number that maps to the respective internal ports: 9999, 9996, 9090. The ports used inside the container for these purposes will keep the default configuration.

2.7 CHANGING DENODO SETTINGS

You can change any Denodo settings in the container image by following standard Denodo Administration procedures. You can open an interactive shell in a denodo container by executing the command:

```
$ docker exec -it denodo-vdpserver bash
```

After completing the changes and exiting the interactive bash session you can save the container state as instructed before.

It's typical to configure JVM memory options to suit your needs. It is recommended that you read the [Denodo Admin and Development Best Practices](#) article section *JVM Configuration* before changing these values.

2.8 RESTRICTING THE NUMBER OF CORES

If you want to limit the number of processors used by the Denodo container, you can use the `--cpuset-cpus` Docker option, that restricts the cores where the docker container is allowed to run. Remember that the range of CPUs set by this parameter cannot surpass the maximum number of CPUs assigned to the Docker daemon.

The following command restricts the execution of VDP Server to the cores from 0 to 3.

```
$ docker run --cpuset-cpus 0-3 -d -h <hostname> -p 9999:9999 -p 9997:9997 -p 9996:9996 -p 9090:9090 -v <path>:/opt/denodo/conf/SolutionManager.properties --rm --name denodo-  
vdpserver denodo-platform:7.0-latest ./denodo-container-start.sh  
--vdpserver
```

2.9 CONFIGURING THE NETWORK

You can assign the container specific IP addresses either from an existing Docker network or by creating a new Docker network. To create a new network the following command can be used:

```
$ docker network create --subnet=172.19.0.0/16 denodonet
```

The container can then be assigned an IP from this new network by using the following command:

```
$ docker run --net denodonet --ip 172.19.0.3 -d -h <hostname> -p  
9999:9999 -p 9997:9997 -p 9996:9996 -p 9090:9090 -v  
<path>:/opt/denodo/conf/SolutionManager.properties --rm --name denodo-  
vdpserver denodo-platform:7.0-latest ./denodo-container-start.sh  
--vdpserver
```