



DH2i DxOdyssey 19.5 Software: with SQL Edge on Docker Quickstart Guide

DH2i Company

Support: +1 (800) 380-5405 ext 2

support.dh2i.com

eFax: +1 970-295-4505

support@dh2i.com

www.dh2i.com

Setting up DxOdyssey and Azure SQL Edge Docker Containers

This document assumes a server located at site A does not have direct network access to site B. The steps below describe how to create DxOdyssey and Azure SQL Edge containers at both sites that can communicate through DxOdyssey tunnels.

Configure DxOdyssey and Azure SQL Edge at Site A

Login to Docker Hub

- `docker login`

Pull a DxOdyssey Docker Image

To pull the latest DxOdyssey Docker image, use the following command.

- `docker pull dh2i/dxo`

Create a Docker Network Bridge

- `docker network create -d bridge dxo-network`

Start the DxOdyssey Container

DxOdyssey uses TCP port 7979 for gateway group administration with the DxOdyssey Client UI. The `-d` flag runs the container detached so that the command line may still be used on the host machine, `-h` assigns the container hostname used by DxOdyssey, `--network` adds the container to the `dxo-network` that was created, `--name` specifies the name within Docker for the container, `-v` adds the path for persistent storage, `-p` is for each port mapping assigned to the container, `DX_LICENSE` sets the license key and `DX_PASSEY` sets a gateway group passkey. The following command example shows port 7979 being mapped to port 17979 for the DxOdyssey Client UI connections. If multiple containers are being run on the same server, then the mapped port needs to be unique for each container.

- `docker run -d -h dxo1 --network=dxo-network --name dxo1 -v dxo1:/etc/dh2i -p 17979:7979 -e "DX_LICENSE=AAAA-BBBB-CCCC-DDDD" -e "DX_PASSEY=P@ssw0rd" dh2i/dxo`

Pull an Azure SQL Edge Image

- `docker pull mcr.microsoft.com/azure-sql-edge:latest`

Start the Azure SQL Edge Container

When starting the Azure SQL Edge container, make sure to specify the `--network` parameter to add the container to the `dxo-network`. The `-v /opt/mssql/bin` parameter will also need to be specified to make the Azure SQL Edge data persistent. Please see Microsoft documentation for more information on Azure SQL Edge specific parameters.

- `Docker run -d --network=dxo-network -v /var/opt/mssql <sql-edge-parameters>`

Connect to the First DxOdyssey Container

- `docker exec -it dxo1 bash`

Set a one-time passkey

Additional gateways may join the gateway group using DxOdyssey tunneling technology. To join another container using NAT, a one-time passkey (OTPK) must be set for the gateway group. Save the GUID output from this command.

Syntax

- `dxcli set-otpk [ttl] [otpk]`

Parameters

Name	Description	Required
Ttl	The time to live	False
Otpk	The one time passkey in base64.	False

Example

- `dxcli set-otpk`

Configure DxOdyssey and Azure SQL Edge at Site B

Login to Docker Hub

- `docker login`

Pull a DxOdyssey Docker Image

Pull a DxOdyssey Docker image onto a server at the second site. This server can be at any location that does not have direct network access to the first site. To pull the latest DxOdyssey Docker image, use the following command.

- `docker pull dh2i/dxo`

Create a Docker Network Bridge

- `docker network create -d bridge dxo-network`

Start the DxOdyssey Container

The OTPK created previously will need to be supplied as a parameter in order for this container to join the existing container. Mapped ports need to be unique for each DxOdyssey Gateway. The following command example shows port 7979 being mapped to port 27979 for the DxOdyssey Client UI connections.

- `docker run -d -h dxo2 --network=dxo-network --name dxo2 -v dxo2:/etc/dh2i -p 27979:7979 -e "DX_LICENSE=AAAA-BBBB-CCCC-DDDD" -e "DX_OTPK= 07509e41-5c17-f931-630b-80b112759979" dh2i/dxo`

Pull an Azure SQL Edge Image

- `docker pull mcr.microsoft.com/azure-sql-edge:latest`

Start the Azure SQL Edge Container

When starting the Azure SQL Edge container, make sure to specify the `--network` parameter to add the container to the `dxo-network`. The `-v /opt/mssql/bin` parameter will also need to be specified to make the Azure SQL Edge data persistent. Please see Microsoft documentation for more information on Azure SQL Edge specific parameters.

- `docker run -d --network=dxo-network -v /var/opt/mssql <sql-edge-parameters>`

Connect to the Second DxOdyssey Container

- `docker exec -it dxo2 bash`

Activate the DxOdyssey License Key

- `dxcli activate-server AAAA-BBBB-CCCC-DDDD`

Join the gateway group

After connecting to the container and activating DxOdyssey, use the saved OTPK GUID to join the gateway group. When prompted to join via NAT proxy, select Yes. When prompted to enter the NAT proxy, use the default value of `match.dh2i.com`. When prompted for the passkey, use the OTPK.

- `dxcli join-gateway-group`

Create a DxOdyssey Tunnel for Azure SQL Edge at Site A

The following command example shows the creation of a tunnel with an origin gateway of the DxOdyssey container at site A (`dxo1`) that accepts connections from any local IP (`0.0.0.0`) on port `11433`, a destination gateway of the DxOdyssey container at site B (`dxo2`) and a destination of the Azure SQL Edge container at site B (`sqledge2`) using the default port of `1433`.

Syntax

- `dxcli add-tunnel <name> <enabled [true|false]> <destination_gateway> <destination_address:destination_port> <origin_gateway,origin_address:origin_port[,address_filter]>|<origin_gateway,origin_address:origin_port[,address_filter]>`

Parameters

Name	Description	Required
<code>name</code>	The name of the tunnel.	True
<code>enabled</code>	Enable the tunnel (true or false).	True
<code>destination_gateway</code>	The name of the destination gateway.	True
<code>destination_address</code>	The IP address of the destination.	True
<code>destination_port</code>	The port number for the destination IP.	True
<code>origin_gateway</code>	The name of the gateway where the listener is active.	True
<code>origin_address</code>	Set to <code>0.0.0.0</code> to allow all IP connections or <code>127.0.0.1</code> for local connections only.	True
<code>origin_port</code>	The port number for the origin gateway.	True
<code>address_filter</code>	The name of the address filter to add to the tunnel.	False

Example

- `dxcli add-tunnel SQL-TUNNEL1 true dxo2 sqledge2:1433 dxo1,0.0.0.0:11433`

Create a DxOdyssey Tunnel for Azure SQL Edge at Site B

The following command example shows the creation of a tunnel with an origin gateway of the DxOdyssey container at site B (`dxo2`) that accepts connections from any local IP (`0.0.0.0`) on port `21433`, a destination gateway of the DxOysey container at site A (`dxo1`) and a destination of the Azure SQL Edge container at site A (`sqledge1`) using the default port of `1433`.

- `dxcli add-tunnel SQL-TUNNEL2 true dxo1 sqledge1:1433 dxo2,0.0.0.0:21433`

Connect the Azure SQL Edge Instances, via Linked Server

The following example shows a TSQL command run on sqledge1 to create a linked server to sqledge2 over DxOdyssey tunnels.

```
--Add linked server via DxO tunnel
EXEC master.dbo.sp_addlinkedserver @server = N'sqledge2', @srvproduct = '', @datasrc='dxo1,11433',
@provider='SQLNCLI'
go
EXEC sp_addlinkedsrvlogin @rmtsrvname='sqledge2', @useself = 'FALSE', @locallogin = NULL, @rmtuser = 'sa',
@rmtpassword='PasswOrd123'
go
EXEC sp_serveroption 'sqledge2', 'rpc out', true;
go
--List the tables in the linked server.
EXEC sp_tables_ex [sqledge2];
go
```

Connect Using the DxOdyssey Client UI

To connect to the containers using the DxOdyssey Client UI, the DxOdysseyClientSetup package will need to be installed on a Windows workstation at either Site A or Site B. Once installed, launch the DxOdyssey Client UI and use the following connection strings to connect to the gateway group.

From Site A:

- Server - <site_a_dxo_container_host_ip>:17979
- Pass Key - <gateway_passkey>

From Site B:

- Server - <site_b_dxo_container_host_ip>:27979
- Pass Key - <gateway_passkey>

References

- [DH2i Support Portal](#)
- [DxOdyssey 19.5 Documentation](#)
- [DxOdyssey v19.5 DxCli Guide](#)