Single Node Installation Guide

Qlik Catalog®

August 2022

QLIK.COM

TABLE OF CONTENTS

1.0 Qlik Catalog Overview and System Requirements

This document describes how to install the "single-node" deployment option for Qlik Catalog. Once the installation is complete, upon first login to the Qlik Catalog application, you will be prompted for a license. A license for "**Qlik Catalog**" must be entered.

1.1 Hardware Configuration Requirements

Single Node Recommendations

- ➢ Recommended Minimum Production Configuration
  - 12 Cores
  - 128GB RAM
  - System Drive 250GB
  - Data Drive 3x expected data
  - Ubuntu or RHEL8 deployments: 2gb **/home** directory OR set custom container storage directory according to Podman documentation
  - Ethernet 10GB
  - Virtual Machine or bare metal
- ➢ Minimum POC/Dev Configuration
  - 8 Cores
  - 32GB RAM
  - System partition 250GB
  - Data partition 3x expected data
  - Ubuntu or RHEL8 deployments: 2gb **/home** directory
  - Ethernet 10GB
  - Virtual Machine or bare metal
- ➢ Minimum Supported Screen Resolution: 1366x768px

1.2 Software Configuration Requirements & Support Matrix

Qlik Catalog and Qlik Enterprise Manager supported versions

- Qlik Enterprise Manager May 2021 and above
- Qlik Catalog November 2020 Service Release 1 (4.8.1+)

Qlik Catalog and Qlik Sense supported versions

- QSEoW/QSD May 2021
- Qlik Catalog February 2021 Service Release 2 (4.9.2)
- QSEoW/QSD February 2021 (latest patch) and November 2020 (latest patch)
- Qlik Catalog February 2021 Service Release 1 (4.9.1)
- QSEoW/QSD November 2020 patch 3
- Qlik Catalog February 2021 (4.9)

System Requirements	**Version**
PostgreSQL Metadata Database	Custom Qlik Catalog PostgreSQL 11. Version 11.16 required for first-time installations.
Oracle Metadata Database	Not supported
Apache Tomcat	Tomcat 9. Version 9.0.64+ encouraged for first-time installations.
Java	OpenJDK 8 or JDK 11, minimum version 1.8.0_222
Podman (RHEL 8 or Ubuntu deployments)	3.4.2+
Browsers	
Google Chrome	80.0 or higher
MS Internet Explorer	Not supported
Other browsers not actively tested.	Issues must be reproducible using Chrome to be eligible for a fix.
Operating Systems	
CentOS 7 Red Hat Enterprise Linux 7 Red Hat Enterprise Linux 8 Ubuntu 20.04 LTS Ubuntu 22.04 LTS	CentOS Linux release 7.0 certified on en_US locale RHEL 7 and RHEL 8 installations require a valid Red Hat entitlement subscription. RHEL 7 installations require access to the following repositories: • rhel-7-server-rpm • rhel-7-server-extras-rpms • rhel-7-server-optional-rpms

Additional Requirements

- All JDBC drivers needed for database connectivity
- Ensure port 8080 or 8443 (http or https) is open from user desktops to the Qlik Catalog node

2.0 Installation Prerequisites

The prerequisite software need only be installed before a first-time Catalog installation, not an upgrade. There is also no requirement to upgrade prerequisites (e.g., Tomcat or PostgreSQL) when upgrading Catalog.

NOTE: In all commands below, the user that the command should be "executed as" is in parentheses at the beginning of the line:

- "(sudo)" means the command should be run as a user with sudo permission
- "(qdc)" means the command should be run as the Qlik Catalog *service account* user -- "sudo su - qdc" may be used to become this user
- "(postgres)" means the command should be run as the PostgreSQL superuser -- "sudo su - postgres" may be used to become this user

Note: Outside ports 80/8080 (HTTP) and 443/8443 (HTTPS) must be opened to allow outbound communication to the Internet to allow software to be downloaded.

Prerequisite Installation Script (strongly encouraged)

There is an optional prerequisite installation script which may be used to install the prerequisite packages described in this section. It's use is strongly encouraged. It will automatically download and configure all required prequisite software, including Java, Tomcat and PostgreSQL. Further, it will automatically configure Tomcat for HTTPS.

It is located within the QDCinstaller.zip package and is named ***QDCprereqs.sh***.

- '**sudo'** permission is required to run *QDCprereqs.sh*
- There are two environment variables at the beginning of the script which may be defined by end-users:
  - **QDC_HOME**: the directory where Qlik Catalog will be installed. (default value: /usr/local/qdc)
  - **QDC_SERVICE_ACCOUNT**: the local user account which will be used to run Qlik Catalog. (default value: qdc)
- RHEL 7/8 installations require a valid Red Hat entitlement subscription. This prerequisite script will not run without a subscription.

Note: if changing the values of the Catalog home directory (QDC_HOME) or service account (QDC_SERVICE_ACCOUNT), the altered values must be consistent within the following Qlik Catalog installation scripts:

- QDCprereqs.sh
- QDCinstaller.properties
- QDCpostinstall.sh

To run *QDCprereqs.sh*:

1. Install unzip (if not present)

   RHEL 7/8 & CentOS 7 Deployments

   (sudo)   # sudo yum install -y unzip

Ubuntu Deployments

(sudo) # sudo apt install -y unzip

2. Copy the QDCinstaller.zip file into an installer “working” directory (example: /tmp)

   Example: (**sudo**) # cp QDCinstaller.zip /tmp

3. Unzip QDCinstaller.zip

   (**sudo**) # cd /tmp
   (**sudo**) # unzip QDCinstaller.zip

4. Copy the podium zip file into the QDCinstaller directory created in Step 3

   Example: (**sudo**) # cp podium-4.15-xxxxx.zip /tmp/QDCinstaller/

5. Run QDCprereqs.sh

   (**sudo**) # cd /tmp/QDCinstaller
   (**sudo**) # ./QDCprereqs.sh

6. After QDCprereqs.sh has been run successfully you may skip to Section 3.1 First Time Installation Using the Installer

Manual Prerequisite Installation

Important: During the prerequisite setup process, several items are needed from the Qlik Catalog software distribution, podium-4.15-xxxxx.zip. The instructions below assume the Qlik Catalog software distribution has been unzipped to /tmp:

1. Install unzip (if not present)

   RHEL 7/8 & CentOS 7 Deployments

   (sudo) # sudo yum install -y unzip

   Ubuntu Deployments
   (sudo) # sudo apt install -y unzip

2. Expand the Qlik Catalog software distribution to /tmp

   (sudo) # unzip <replace-path>/podium-4.15-xxxxx.zip -d /tmp/

2.1 Java JDK Installation

Qlik Catalog is supported on the following JDK platforms:

- OpenJDK 8
- OpenJDK 11

- Oracle JDK 8 (license required)

1. Check if JDK exists. If a correct version exists, skip this step.

   (sudo) # java -version

   JDK 8 results:

   ```
   Openjdk version "1.8.0_222"
   OpenJDK Runtime Environment (build 1.8.0_222-b10)
   OpenJDK 64-Bit Server VM (build 25.222-b10, mixed mode)
   ```

   JDK 11 results:

   ```
   openjdk version "11.0.6" 2020-01-14 LTS
   OpenJDK Runtime Environment 18.9 (build 11.0.6+10-LTS)
   OpenJDK 64-Bit Server VM 18.9 (build 11.0.6+10-LTS, mixed mode, sharing)
   ```

2. JDK Installation:
   ➔ OpenJDK: Use package manager for installation:

   OpenJDK 8 installation:

   RHEL 7/8 & CentOS 7 Deployments

   (sudo) # sudo yum install -y java-1.8.0-openjdk-devel

   Ubuntu Deployments

   (sudo) # sudo apt install -y openjdk-8-jdk

   OpenJDK 11 installation:

   RHEL 7/8 & CentOS 7 Deployments

   (sudo) # sudo yum install -y java-11-openjdk-devel

   Ubuntu Deployments

   (sudo) # sudo apt install -y openjdk-11-jdk

Oracle JDK 8 (license required): Download the package directly from Oracle and install.

2.2 Create Service Account and Qlik Catalog Directory

Create a ***service account*** to run Qlik Catalog. Tomcat will be started as this user. Typically, this user is named “qdc” or “qdcsvc”. Throughout the remainder of this document “qdc” will be used -- please replace “qdc” with a different user if so desired. In a similar fashion, “qdc” is also used as a group name.

1. Create a service account to run Qlik Catalog (the user which launches Tomcat)

   (sudo) # sudo groupadd qdc
   (sudo) # sudo useradd -s /bin/bash -g qdc qdc

   Additional step for Ubuntu deployments:

(sudo) mkdir /home/qdc && chown qdc:qdc /home/qdc

2. Optionally, set a password for the service account - this is not needed if “sudo” is used to become this user (e.g., “sudo su - qdc”)

   (sudo) # sudo passwd qdc

3. Create a directory for all Qlik Catalog artifacts, including Tomcat

   (sudo) # sudo mkdir /usr/local/qdc

4. Change ownership of /usr/local/qdc to the service account and group being used

   (sudo) # sudo chown -Rf qdc:qdc /usr/local/qdc

2.3 Tomcat Installation

Qlik Catalog is only supported on Tomcat version 9. Version 9.0.64+ is encouraged for first-time installs. There is no requirement to upgrade Tomcat when upgrading Catalog. If needed, Tomcat upgrade instructions are included elsewhere in this guide.

Qlik strongly encourages the use of SSL with Tomcat for securing Qlik Catalog sessions. Instructions for configuring Tomcat to support SSL connections are provided later in this document.

1. Install wget (if not present) while a sudo capable user

   RHEL 7/8 & CentOS 7 Deployments

   (sudo) # sudo yum install -y wget

   Ubuntu Deployments

   (sudo) # sudo apt install -y wget

2. Become the service account user

   (sudo) # sudo su - qdc

3. Download Apache Tomcat 9.0.64+

   (qdc) $ cd /usr/local/qdc
   (qdc) $ wget https://archive.apache.org/dist/tomcat/tomcat-9/v9.0.64/bin/apache-tomcat-9.0.64.tar.gz

4. Extract the Tomcat file

   (qdc) $ tar -xvf apache-tomcat-9.0.64.tar.gz

5. The resulting directory, for example “/usr/local/qdc/apache-tomcat-9.0.64”, is known as the Tomcat home directory. When configuring the QDCinstaller.properties file in the next section, please set TOMCAT_HOME to this value.

6. Overwrite <tomcat home>/conf/server.xml with the version expanded from the Qlik Catalog zip file or edit the existing server.xml manually.

   Overwrite Instructions (recommended)

   ```
   (qdc)    $ cp /tmp/podium/config/tomcat9-server.xml /usr/local/qdc/apache-tomcat-9.0.64/conf/server.xml
   ```

   OR

   Manual Edit Instructions:

   In the HTTP Connector element, add the bold attributes to turn compression on

   ```
   <Connector port="8080" protocol="HTTP/1.1"
              connectionTimeout="20000"
              useSendfile="false" maxHttpHeaderSize="65536"
              compression="on"
              compressionMinSize="150"
              noCompressionUserAgents="gozilla, traviata"
              compressibleMimeType="text/html,text/xml,text/plain,text/css,text/javas
              cript,application/javascript,application/json,application/xml"
              redirectPort="8443" />
   ```

   In the AccessLogValve element, change the bold attributes prefix, suffix and pattern

   ```
   <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
              prefix="localhost_access" suffix=".log"
              pattern="%h %l %u %t &quot;%r&quot; %s %b %{podiumUser}s %{podiumSession}s [%I]" />
   ```

7. Port 8080 needs to be opened on the Qlik Catalog node firewall for HTTP connections. Port 8443 needs to be opened on the Qlik Catalog node firewall for secure HTTPS connections.

   ➔ It is recommended that Tomcat be configured to ***redirect*** insecure HTTP connections directly to a secure HTTPS session. Instructions for configuring **HTTPS redirect** are provided later in this document.

   RHEL 7/8 & CentOS 7 Deployments

   ```
   (sudo)    # sudo firewall-cmd --zone=public --permanent --add-port=8080/tcp
   (sudo)    # sudo firewall-cmd --zone=public --permanent --add-port=8443/tcp
   (sudo)    # sudo systemctl restart firewalld.service
   ```

   Ubuntu Deployments (if UFW firewall is enabled)

   ```
   (sudo)    # sudo ufw allow 8080,8443/tcp
   ```

8. Setup Apache Tomcat as a service to automatically start when Linux restarts

   Possible edit: if the Tomcat home directory is not /usr/local/qdc/apache-tomcat-9.0.64 or the service user/group is not “qdc,” the file /etc/systemd/system/tomcat.service must be edited after the copy (cp) step below.

   The copy (cp) command below assumes the Qlik Catalog software (a.k.a., podium zip) has been expanded to /tmp -- see Section 2.0. You will not be able to start the service until Qlik Catalog PostgreSQL is later installed, as a dependency exists.

   ```
   (sudo)    # sudo cp /tmp/podium/config/tomcat.service /etc/systemd/system/
   (sudo)    # sudo systemctl daemon-reload
   ```

```
(sudo)    # sudo systemctl enable tomcat.service
```

9. Optional: Configure Tomcat to support HTTPS.

   Configuring Tomcat to support HTTPS may be done now or later.

10. Start Tomcat manually

```
(qdc)    $ cd <tomcat home>
(qdc)    $ ./bin/startup.sh
```

11. Browse to the following URL to verify that Tomcat is running

    http://<Qlik-Catalog-Node-IP-Address-OR-Hostname>:8080

12. Tomcat can be manually stopped at any time

```
(qdc)    # cd <tomcat home>
(qdc)    # ./bin/shutdown.sh
```

13. The Tomcat log can be monitored

```
(qdc)    # tail -F <tomcat home>/logs/catalina.out
```

2.4 PostgreSQL Installation

Qlik Catalog is certified on Qlik Catalog PostgreSQL 11, a customized version of PostgreSQL that supports long schema and table names. First-time Catalog installations must install Qlik Catalog PostgreSQL 11.16. There is no requirement to upgrade PostgreSQL when upgrading Catalog. If needed, PostgreSQL upgrade instructions are included elsewhere in this guide.

To ensure this version is used, the Qlik Catalog PostgreSQL installer has been included in the Qlik Catalog zip file. The directions below describe how to extract and install this custom version of PostgreSQL, and then configure it.

Do **NOT** install PostgreSQL using rpm, yum or otherwise download it from the Internet.

NOTE: If you already have a different version of PostgreSQL installed, please first uninstall it.

NOTE: The below instructions assume that the Qlik Catalog zip file has already been extracted to /tmp -- see Section 2.0.

1. Create a “postgres” user and group

```
(sudo)    # sudo groupadd postgres
(sudo)    # sudo useradd -s /bin/bash -g postgres postgres
```

   Additional step for Ubuntu deployments:

```
(sudo)    # sudo mkdir /home/postgres && chown postgres:postgres /home/postgres
```

2. Add the “postgres” user to the “qdc” group

```
(sudo)    # sudo usermod -aG qdc postgres
```

3. Create directories for executables and data, and change their ownership

   ```
   (sudo)     # sudo mkdir -p /usr/pgsql/qdc11-16
   (sudo)     # sudo chown -R postgres:postgres /usr/pgsql
   (sudo)     # sudo mkdir -p /var/lib/pgsql/11-16/qdc_data
   (sudo)     # sudo chown -R postgres:postgres /var/lib/pgsql
   ```

4. Run the custom Qlik Catalog PostgreSQL installer appropriate for your operating system as the ***postgres*** user:

   ```
   (sudo)          # sudo su - postgres
   ```

   Installer for RHEL7/CentOS7 deployments**:**

   ```
       (postgres)     $ /tmp/podium/thirdParty/qdc_pg11-16_RHEL7-and-CentOS7.bsx
   ```

   Installer for RHEL8/CentOS8 deployments:

   ```
       (postgres)     $ /tmp/podium/thirdParty/qdc_pg11-16_RHEL8-and-CentOS8.bsx
   ```

   Installer for Ubuntu 20.04 deployments:

   ```
       (postgres)     $ /tmp/podium/thirdParty/qdc_pg11-16_Ubuntu20.bsx
   ```

   Installer for Ubuntu 22.04 deployments:

   ```
       (postgres)     $ /tmp/podium/thirdParty/qdc_pg11-16_Ubuntu22.bsx
   ```

5. Create symlinks to executables

   ```
   (sudo)     # sudo ln -s /usr/pgsql/qdc11-16/bin/psql /usr/bin/psql
   (sudo)     # sudo ln -s /usr/pgsql/qdc11-16/bin/pg_dump /usr/bin/pg_dump
   (sudo)     # sudo ln -s /usr/pgsql/qdc11-16/bin/pg_restore /usr/bin/pg_restore
   ```

6. IMPORTANT! Port 5432 needs to be opened on the Qlik Catalog node firewall to allow connections to PostgresSQL. The Qlik Licenses container is dependent upon communication with the PostgreSQL database.

   RHEL 7/8 & CentOS 7 Deployments

   ```
     (sudo)     # sudo firewall-cmd --zone=public --permanent --add-port=5432/tcp

     (sudo)     # sudo systemctl restart firewalld.service
   ```

   Ubuntu Deployments

   ```
     (sudo)     # sudo ufw allow 5432/tcp
   ```

7. Set PostgreSQL to start automatically, then start it.

   Possible edit: if the directories in step 3 were altered, or the user/group is not “postgres”, the file /etc/systemd/system/qdc_pg.service must be edited after the copy (cp) step below.

   The copy (cp) command below assumes the Qlik Catalog software (a.k.a., podium zip) has been expanded to /tmp -- see Section 2.0.

   ```
   (sudo)     # sudo cp /tmp/podium/config/qdc_pg.service /etc/systemd/system/
   (sudo)     # sudo systemctl daemon-reload
   (sudo)     # sudo systemctl enable qdc_pg.service
   ```

```
(sudo)    # sudo systemctl start qdc_pg.service
```

2.5 Container Platform & Node.js Installation

Qlik Catalog requires a container platform to run containers. Platform is determined by the operating system upon which Catalog is deployed:

RHEL7 & CentOS 7: *Docker*

RHEL8, Ubuntu 20.04, and Ubuntu 22.04: *Podman*

Node.js is required for integration with Qlik Sense.

1. Install Node.js (which includes npm)

   RHEL 7/8 & CentOS 7 Deployments

   ```
   (sudo)    # curl -sL https://rpm.nodesource.com/setup_14.x | sudo bash -
   (sudo)    # sudo yum install -y nodejs
   ```

   Ubuntu Deployments

   ```
   (sudo)    # curl -sL https://deb.nodesource.com/setup_14.x | sudo -E bash -
   (sudo)    # sudo apt install -y nodejs
   ```

2. Install Container Platform

   CentOS 7 Deployments: *Docker Community Edition* (CE)

   Docker CE installation instructions for CentOS 7

   1. Install Docker, set it to start automatically, and start it:

      ```
      (sudo)    # sudo yum install -y yum-utils device-mapper-persistent-data lvm2
      (sudo)    # sudo yum-config-manager --add-repo
      https://download.docker.com/linux/centos/docker-ce.repo
      (sudo)    # sudo yum install -y docker-ce

      (sudo)    # sudo systemctl enable docker.service
      (sudo)    # sudo systemctl start docker.service
      ```

   2. Add the service user to the “docker” group

      ```
      (sudo)    # sudo usermod -aG docker qdc
      ```

   3. Install Docker Compose

      ```
      (sudo) sudo curl -L "https://github.com/docker/compose/releases/download/1.23.2/docker-
      compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
      ```

      ```
      % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                     Dload  Upload   Total   Spent    Left  Speed
      100   617  100   617    0     0   2114      0 --:--:-- --:--:-- --:--:--  2127
      100 11.2M  100 11.2M    0     0  13.9M      0 --:--:-- --:--:-- --:--:-- 32.9M
      ```

      ```
      (sudo)    # sudo chmod +x /usr/local/bin/docker-compose
      ```

4. Test Docker and Docker Compose -- this should be done as the service account user

(sudo) # sudo su - qdc

(qdc) $ docker ps

```
CONTAINER ID   IMAGE   COMMAND   CREATED   STATUS   PORTS   NAMES
```

(qdc) $ docker-compose version

```
docker-compose version 1.23.2, build 1110ad01
docker-py version: 3.6.0
CPython version: 3.6.7
OpenSSL version: OpenSSL 1.1.0f  25 May 2017
```

RHEL 7 Deployments: *Docker Enterprise*

- ***Docker Enterprise*** is the officially supported Docker platform for RHEL 7.
- A Red Hat entitlement subscription is required to access the RHEL repository containing the Docker Enterprise engine package.
- Official RHEL 7 Docker Enterprise installation instructions may be found here: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux_atomic_host/7/html-single/getting_started_with_containers/index#using_the_docker_command_and_service

Docker Enterprise installation instructions for RHEL 7

1. Install Docker, set it to start automatically, and start it:

(sudo) # subscription-manager repos --enable=rhel-7-server-rpms
(sudo) # subscription-manager repos --enable=rhel-7-server-extras-rpms
(sudo) # subscription-manager repos --enable=rhel-7-server-optional-rpms

(sudo) # yum install docker device-mapper-libs device-mapper-event-libs
(sudo) # systemctl enable docker.service
(sudo) # systemctl start docker.service

(sudo) # groupadd docker
(sudo) # chown -R root:docker /run/docker /etc/sysconfig/docker /etc/docker /var/lib/docker /usr/bin/docker /usr/share/bash-completion/completions/docker /usr/libexec/docker
(sudo) # chmod -R 770 /var/lib/docker

(sudo) # systemctl restart docker.service

2. Add the service user to the "docker" group

(sudo) # sudo usermod -aG docker qdc

3. Install Docker Compose

(sudo) sudo curl -L "https://github.com/docker/compose/releases/download/1.23.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

```
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                               Dload  Upload   Total   Spent    Left  Speed
100   617  100   617    0     0   2114      0 --:--:-- --:--:-- --:--:--  2127
100 11.2M  100 11.2M    0     0  13.9M      0 --:--:-- --:--:-- --:--:-- 32.9M
```

```
(sudo)    # sudo chmod +x /usr/local/bin/docker-compose
```

4. Test Docker and Docker Compose -- this should be done as the service account user

```
(sudo)   # sudo su - qdc

(qdc)    $ docker ps

    CONTAINER ID    IMAGE    COMMAND    CREATED    STATUS    PORTS    NAMES

(qdc)    $ docker-compose version

         docker-compose version 1.23.2, build 1110ad01
         docker-py version: 3.6.0
         CPython version: 3.6.7
         OpenSSL version: OpenSSL 1.1.0f  25 May 2017
```

RHEL 8 Deployments: *Podman*

- ***Podman*** is the officially supported container platform for RHEL 8.
- *Podman* is a "daemonless" container platform

Podman installation instructions for RHEL 8

```
(sudo)    # yum module install -y container-tools
```

Test Podman command as service account user

```
(sudo)    # sudo su - qdc
(qdc)     $ podman info
```

Ubuntu 20.04 & Ubuntu 22.04 Deployments: *Podman*

- ***Podman*** is the supported container platform for Ubuntu 20.04 & Ubuntu 22.04 deployments.
- *Podman* is a "daemonless" container platform

Podman installation instructions for **Ubuntu 20.04**

```
(sudo)    # apt install curl gnupg2 -y
(sudo)    # echo 'deb
http://download.opensuse.org/repositories/devel:/kubic:/libcontainers:/stable/xUbuntu_20.04/
/' > /etc/apt/sources.list.d/devel:kubic:libcontainers:stable.list
(sudo)    # wget -nv
https://download.opensuse.org/repositories/devel:kubic:libcontainers:stable/xUbuntu_20.04/R
elease.key -O- | apt-key add -
(sudo)    # apt-get update -qq -y
(sudo)    # apt-get -qq --yes install podman
```

Test Podman command as service account user

```
(sudo)    # sudo su - qdc
```

(qdc) $ podman info

Podman installation instructions for **Ubuntu 22.04**

(sudo) # apt-get -y install podman

Test Podman command as service account user

(sudo) # sudo su - qdc

(qdc) $ podman info

3.0 Qlik Catalog Software Installation

The Qlik Catalog installer is a shell script, ***QDCinstaller.sh***, that is guided by a properties file containing configuration, ***QDCinstaller.properties***. The same properties file can be used both for the initial install as well as later upgrades.

The shell script and properties files are included in a zip file, **QDCinstaller.zip**. Required installation parameters are first configured in the properties file. Then, the shell script is run and begins by confirming the installation parameters.

Password Encryption

Passwords may be encrypted at any time using a utility. It will use stdin to prompt for the password and output the encrypted password to stdout.

```
$ unzip -j podium-4.15-xxxxx.zip podium/lib/podium-encrypt-util.jar -d .
$ java -cp podium-encrypt-util.jar com.nvs.core.utils.PodiumEncrUtil
```

NOTE: In all commands below, the user that the command should be “executed as” is in parentheses at the beginning of the line:

- “(sudo)” means the command should be run as a user with sudo permission
- “(qdc)” means the command should be run as the Qlik Catalog service account user -- “sudo su - qdc” may be used to become this user
- “(postgres)” means the command should be run as the PostgreSQL superuser -- “sudo su - postgres” may be used to become this user

Note: if changing the values of the Catalog home directory (QDC_HOME) or service account (QDC_SERVICE_ACCOUNT), the altered values must be consistent within the following Qlik Catalog installation scripts:

- QDCprereqs.sh
- QDCinstaller.properties
- QDCpostinstall.sh

3.1 First-time Installation Using the Installer

Execute the following steps to perform a first-time install:

1. The installer must be run as the Qlik Catalog service account:

   (sudo) # sudo su - qdc

2. Unzip QDCinstaller.zip into a working directory

3. Copy the Qlik Catalog software zip file (e.g., podium-4.15-xxxxx.zip) into the working directory

4. Change directory to the working directory. It should contain the following:

```
podium-4.15-xxxxx.zip  QDCinstaller.properties  QDCinstaller.sh  QDCinstaller.txt   QCA.txt. upgrade-scripts (directory)
```

5. Edit the installation parameters in **QDCinstaller.properties** -- additional documentation is present in that file. Ensure the following are set -- the defaults should work for most. PODIUM_RELEASE_FILE is set to the current podium zip file. The properties SUPERUSER_NAME and SUPERUSER_PASSWORD are for Qlik Catalog PostgreSQL and by default are both "postgres".

   ⊄ INSTALL_TYPE -- **SINGLE**
   ⊄ QDC_HOME
   ⊄ TOMCAT_HOME
   ⊄ PODIUM_RELEASE_FILE
   ⊄ WEBAPP_NAME
   ⊄ POSTGRES_HOSTNAME
   ⊄ POSTGRES_IPADDRESS -- programmatically initialized to local IP address; override if necessary
   ⊄ CATALOG_IPADDRESS -- likely same as POSTGRES_IPADDRESS
   ⊄ SUPERUSER_NAME
   ⊄ SUPERUSER_PASSWORD
   ⊄ JAVA_HOME -- programmatically initialized; override if necessary
   ⊄ PODIUM_BASE

6. Interactive & Non-Interactive installation modes:

   The Qlik Catalog installer shell script may be run in interactive mode or non-interactive “silent” mode.

   Interactive installation allows users to confirm installer actions in a step-by-step manner. To run the installer in interactive mode simply run: ./QDCinstaller.sh

   Non-Interactive or “silent” installation allows users to deploy QDC in a scripted manner which does not require any user interaction. Details for running the installer in non-interactive mode are noted later in this document.

7. Run the shell script. You may abort the script at any time by entering control-C. The installer will begin by asking you to confirm the data you entered in QDCinstaller.properties -- it will also validate the parameters. It will expand the Qlik Catalog software zip and create all necessary configuration files. It will then setup the database (if PostgreSQL is used).

   ```
   (qdc)     $ ./QDCinstaller.sh
   ```

8. When complete, Tomcat is automatically started. It may be started and shutdown using the following:

   ```
   (qdc)     $ <tomcat home>/bin/startup.sh
   (qdc)     $ <tomcat home>/bin/shutdown.sh
   ```

   The log can be viewed as follows:

   ```
   (qdc)     $ tail -F <tomcat home>/logs/catalina.out
   ```

9. Open a browser and go to the following URL to validate that you can display the homepage. If a WEBAPP_NAME other than “qdc” was specified in QDCinstaller.properties, please replace “qdc” with the alternative webapp name.

http://<QDC node hostname or IP address>:8080/qdc

10. Attempt to login for the first time (user: podium, password: nvs2014!) and a prompt will appear to enter a license key. Enter the provided key and click register.

3.1.1 Post-Installation Tasks

The installer deployed four containers which are used by Qlik Catalog:

- Licenses
- Engine
- DCAAS
- Data Rest Connector

These containers should be configured to **auto-start during system boot**. (In fact, the Licenses container MUST be running in order to login to Qlik Catalog).

A **post-installation script** which simplifies the process of configuring the containers to run as services is included, and its use is strongly encouraged. It is located within the QDCinstaller.zip package and is named ***QDCpostinstall.sh***:

> Note: if changing the values of the Catalog home directory (QDC_HOME) or service account (QDC_SERVICE_ACCOUNT), the altered values must be consistent within the following Qlik Catalog installation scripts:
>
> - QDCprereqs.sh
> - QDCinstaller.properties
> - QDCpostinstall.sh

There are two environment variables at the beginning of the script which may be defined by end-users:

- **QDC_HOME**: the directory where Qlik Catalog has been installed. (default value: /usr/local/qdc)
- **QDC_SERVICE_ACCOUNT**: the local user account which is used to run Qlik Catalog. (default value: qdc)

NOTE: '**sudo'** permission is required to run *QDCpostinstall.sh*

To run *QDCpostinstall.sh*:

1. Switch to a user account with 'sudo' privileges

2. Navigate to the QDCinstaller working directory (used above to run the Catalog installer)

   Example: `$ cd /tmp/QDCinstaller`

3. Modify variables if non-default **location** and/or **service account** was used during installation.

   Example: `$ sudo vi QDCpostinstall.sh`

4. Run QDCpostinstall.sh using an account with 'sudo'

   (sudo) `$ sudo ./QDCpostinstall.sh`

Manual Container Service Configuration

If you did not run QDCpostinstall.sh, then do the following.

a) If the Qlik Catalog Qlik Sense integration directory is not /usr/local/qdc/qlikcore or the service user/group is not “qdc,” the file **/etc/systemd/system/qlikContainers.service** must be edited after the copy (cp) step below

b) If deploying to RHEL 8, Ubuntu 20.04, or Ubuntu 22.04 which use Podman to manage containers you must edit **/etc/systemd/system/qlikContainers.service** after the copy (cp) step below:

   a. **REMOVE** line 13 completely:

      Requires=docker.service

   b. EDIT line 14 and remove the following text: **docker.service**

The copy (cp) command below assumes the Qlik Catalog software (a.k.a., podium zip) has been expanded to /tmp in the prerequisites section.

```
(sudo)    $ sudo cp /tmp/podium/config/qlikContainers.service /etc/systemd/system/
(sudo)    $ sudo systemctl daemon-reload
(sudo)    $ sudo systemctl enable qlikContainers.service
```

Congratulations! Qlik Catalog software installation is now complete.

The installer created two databases: the Qlik Catalog metadata database (podium_md) and the distribution database (podium_dist). The default users (roles) for these databases are podium_md and podium_dist, both with a default password of “nvs2014!”.

Important Notes

- Any **JDBC drivers** for your licensed RDBMS should be placed in the directory called out by the following core_env property. This directory is preferred over placing drivers in $TOMCAT_HOME/webapps/qdc/WEB-INF/lib, where they will be overwritten on upgrade and where they may interfere with other libraries.

```
# An alternate directory to WEB-INF/lib for JDBC driver jars.
# May also be set directly, for a given driver, on table
# podium_core.pd_jdbc_source_info, column alt_classpath.
# Restart required. Default: not set
jdbc.alternate.classpath.dir=/usr/local/qdc/jdbcDrivers
```

- If a JDBC driver is particularly complicated and consists of multiple jars (e.g., the Simba Google Big Query driver has dozens of jars), it can be further isolated into its own sub-directory. If you do this, you must run a SQL statement as follows (default password is “nvs2014!”; **update** path and name):

```
psql podium_md -U podium_md -c "update podium_core.pd_jdbc_source_info set alt_classpath =
     '/usr/local/qdc/jdbcDrivers/simbaBigQuery' where sname = 'BIGQUERY';"
```

REMINDER: Reboot Procedure

When the Qlik Catalog server is restarted, several required processes must be started.

The following are **autostarted** services. To manually restart these services:

- PostgreSQL: (sudo) # sudo systemctl restart qdc_pg.service
    - test: (sudo) # psql
- RHEL7/CentOS 7 Deployments using Docker
    - Docker: (sudo) # sudo systemctl restart docker.service
    - test: (sudo) # sudo docker ps

The following should be **autostarted** services, if configured correctly above. To manually restart these services:

- Qlik Licenses & Engine Containers:
  (sudo) # sudo systemctl restart qlikContainers.service

  Check for Running Containers:
    - *Docker*:
      (sudo) # sudo docker inspect -f '{{.State.Running}}' qlikcore_qix-engine_1
      (sudo) # sudo docker inspect -f '{{.State.Running}}' licenses
    - *Podman*:
      (qdc) $ podman inspect -f '{{.State.Running}}' catalog-pod-engine
      (qdc) $ podman inspect -f '{{.State.Running}}' catalog-pod-licenses
- Tomcat: (sudo) # sudo systemctl restart tomcat.service

If the following were not configured to be autostarted services, they must be manually restarted after reboot. First, become the service user: (sudo) # sudo su - qdc

- Qlik Licenses & Engine Containers: (qdc) $ cd /usr/local/qdc/qlikcore && ./launch_qlikContainers.sh
    - Test using *Docker*:
      (qdc) $ docker inspect -f '{{.State.Running}}' qlikcore_qix-engine_1
      (qdc) $ docker inspect -f '{{.State.Running}}' licenses
    - Test using *Podman*:
      (qdc) $ podman inspect -f '{{.State.Running}}' catalog-pod-engine
      (qdc) $ podman inspect -f '{{.State.Running}}' catalog-pod-licenses
- Tomcat: (qdc) $ /usr/local/qdc/apache-tomcat-9.0.64/bin/startup.sh

3.2 Upgrade of Qlik Catalog

NOTE: Beginning with Qlik Catalog May 2021, **Apache Tomcat 9 is mandatory**. Prior installations using Tomcat 7 must be migrated. The installer will disallow use of any version of Tomcat other than version 9. Please see the section "Migrating to or Upgrading Tomcat 9". Notably, at the end, the installer should be run **WITHOUT** the upgrade option “-u”. Please review the release notes and the below guidance before proceeding.

You should re-use the QDCinstaller.properties file from the initial install, updating the value for PODIUM_RELEASE_FILE.

The installer script has an upgrade mode, which also performs a backup of the WEB-INF/classes directory. Execute the following steps to perform an upgrade of Qlik Catalog June 2019 and later:

1. The installer must be run as the Qlik Catalog service account:

   ```
   (sudo)    # sudo su - qdc
   ```

2. Stop Tomcat. Ensure it is no longer running.

   ```
   (qdc)    $ cd <tomcat home>
   (qdc)    $ ./bin/shutdown.sh
   (qdc)    $ ps -ef | grep Boot
   ```

3. **Backup** any manually copied Java library jars from <tomcat home>/webapps/qdc/WEB-INF/lib.

4. **Backup** the PostgreSQL databases, in case the upgrade must be reverted.

   ```
   (sudo)    # pg_dump -U postgres --format=c --file=<backupFileName1> podium_md
   (sudo)    # pg_dump -U postgres --format=c --file=<backupFileName2> podium_dist
   ```

5. Unzip QDCinstaller.zip into a working directory

6. Copy the Qlik Catalog software ZIP file (e.g., podium-4.15-xxxxx.zip) into the working directory

7. Change directory to the working directory. It should contain the following:

   ```
   podium-4.15-xxxxx.zip  QDCinstaller.properties  QDCinstaller.sh  QDCinstaller.txt  QCA.txt  upgrade-scripts (directory)
   ```

8. Edit the installation parameters in QDCinstaller.properties -- additional documentation is present in that file

   - Only the following are used for upgrade: INSTALL_TYPE, QDC_HOME, TOMCAT_HOME, PODIUM_RELEASE_FILE, WEBAPP_NAME, POSTGRES_HOSTNAME, POSTGRES_IPADDRESS, CATALOG_IPADDRESS, and database SUPERUSER_NAME and SUPERUSER_PASSWORD.

9. Run the shell script **with the "-u" argument**. You may abort the script at any time by entering control-C. The installer will begin by asking you to confirm the data you entered in QDCinstaller.properties -- it will also validate the parameters. It will expand the Qlik Catalog software ZIP and update the webapp. A **backup** of WEB-INF/classes is automatically made in <tomcat home>/backups. The file WEB-INF/classes/log4j.xml (or log4j2.xml) is automatically restored during upgrade.

   ```
   (qdc) # ./QDCinstaller.sh -u
   ```

10. **Restore** any manually copied Java library jars to <tomcat home>/webapps/qdc/WEB-INF/lib. If files were restored, restart Tomcat.

11. The Qlik Licenses container must be setup as a service to automatically start when Linux restarts.

    Possible edit: If the Qlik Catalog Qlik Sense integration directory is not /usr/local/qdc/qlikcore or the service user/group is not qdc, the file /etc/systemd/system/qlikContainers.service must be edited after the copy (cp) step below.

    The copy (cp) command below assumes the Qlik Catalog software (a.k.a., podium zip) has been expanded to /tmp -- the prerequisites section.

(sudo) # sudo cp /tmp/podium/config/qlikContainers.service /etc/systemd/system/
(sudo) # sudo systemctl daemon-reload
(sudo) # sudo systemctl enable qlikContainers.service

12. If previously using the QVD Import feature then **remove the existing *qlikcore.service* file**. The Qlik Engine container required for the QVD Import feature has been included in the unified *qlikContainers.service* file configured in step 11.

    (sudo) # sudo rm /etc/systemd/system/qlikcore.service

3.3 Non-Interactive ("Silent") Installation

Note: If upgrading from a Catalog version prior to 4.7 non-interactive mode is not allowed by the installer due to mandatory upgrade scripts which must be run manually.

To run the installer in non-interactive mode:

1. Edit the installation parameters in **QDCinstaller.properties** -- additional documentation is present in that file. Ensure the following are set -- the defaults should work for most. PODIUM_RELEASE_FILE is set to the current podium zip file. The properties SUPERUSER_NAME and SUPERUSER_PASSWORD are for Qlik Catalog PostgreSQL and by default are both "postgres".
   - INSTALL_TYPE -- **SINGLE**
   - QDC_HOME
   - TOMCAT_HOME
   - PODIUM_RELEASE_FILE
   - WEBAPP_NAME
   - POSTGRES_HOSTNAME
   - POSTGRES_IPADDRESS
   - CATALOG_IPADDRESS
   - SUPERUSER_NAME
   - SUPERUSER_PASSWORD
   - JAVA_HOME
   - PODIUM_BASE

2. The following options must be specified following the QDCinstaller.sh command:

   - -s (silent)
   - -a (Accept Qlik Customer Agreement)

   Example: ./QDCinstaller.sh -s -a

NOTE: Invoking the "-a" option indicating acceptance of the Qlik Customer Agreement (QCA) is required to run the installer in silent mode. By selecting this installation option the user agrees to the following:

> BY DOWNLOADING, INSTALLING, OR OTHERWISE USING QLIK PRODUCTS, THE CUSTOMER ACKNOWLEDGES AND AGREES THAT THE USE OF ALL QLIK PRODUCTS IS SUBJECT TO THE TERMS AND CONDITIONS OF THE QLIK CUSTOMER AGREEMENT (QCA) FOUND ON https://qlik.com. ANY SUCH USE WILL CONSTITUTE CUSTOMER'S ACCEPTANCE AND RESULT IN A BINDING AND LEGALLY ENFORCEABLE AGREEMENT BETWEEN THE CUSTOMER AND THE QLIK ENTITY IDENTIFIED IN TABLE 1 OF THE AGREEMENT ("QLIK"). IF YOU ACCEPT THESE TERMS ON BEHALF OF ANY CORPORATION, PARTNERSHIP, OR OTHER ENTITY, YOU REPRESENT AND WARRANT THAT YOU ARE AUTHORIZED TO LEGALLY BIND SUCH ENTITY TO THIS AGREEMENT AND SUCH ENTITY IS DEEMED THE CUSTOMER HEREUNDER.

4.0 Qlik Catalog Software Installation Reference

All following sections are not part of the base installation runbook and as such are treated in isolation.

The **core_env.properties** file contains all the primary and secondary configuration parameters necessary for desired operation. The file is found at <tomcat home>/conf/core_env.properties. It is self-describing in that the explanation for all parameters is included in the file itself. Please see the file and modify the parameters as needed. Once Qlik Catalog is launched, you can edit the file and then use the button in the admin section to refresh core_env.properties if any changes are made, which prevents having to restart Tomcat for these changes.

4.1 Enabling SAML

This section covers configuring Catalog to use one of two SAML providers: Microsoft Azure Active Directory or Okta. If using a different SAML provider, please review this section and apply the same approach to your provider.

Note for Other SAML Providers

Some organizations prohibit the HTTP-based exchange of metadata between their Identity Provider (typically on-premise) and Service Provider (Catalog). If so, Catalog can be configured to consume a local, exported copy of the Identity Provider (IdP) metadata by setting a core_env property to reference a local file:

```
saml.metadata.provider=file:///usr/local/qdc/idp.xml
```

Further, Catalog's Service Provider metadata can be exported to an XML file for hand delivery to the Identity Provider:

```
curl -k
https://<REPLACE_WITH_FULL_CATALOG_HOSTNAME>:8443/qdc/saml2/service-
provider-metadata/catalog > spExport.xml
```

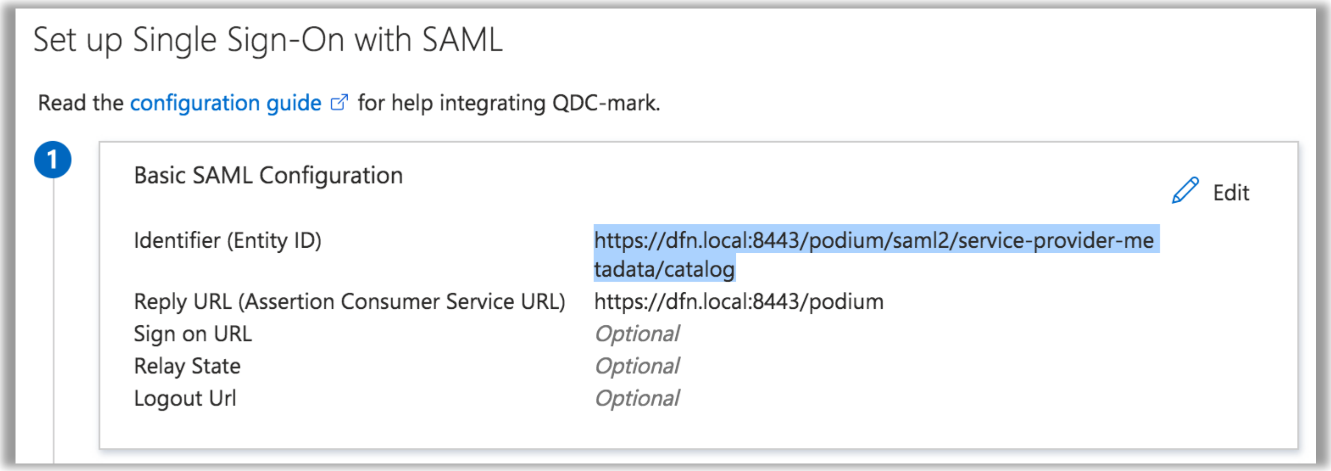
The hostname in the above URL should be that used by end-users to access Catalog.

4.1.1 Enabling SAML using Microsoft Azure Active Directory (MS AAD)

(1) Qlik Catalog Tomcat must first be configured for HTTPS (see section below) -- new installs since May 2022 are automatically configured with HTTPS.

(2) Two properties must be set in the MS AAD “Single Sign-On with SAML” “Basic SAML Configuration” dialog – replace <HOSTNAME> with the Qlik Catalog hostname:

- **Identifier (Entity ID)** – https://<HOSTNAME>:8443/qdc/saml2/service-provider-metadata/catalog
- **Reply URL** – https://<HOSTNAME>:8443/qdc



Set up Single Sign-On with SAML

Read the configuration guide for help integrating QDC-mark.

1 Basic SAML Configuration

Identifier (Entity ID)	https://dfn.local:8443/podium/saml2/service-provider-metadata/catalog
Reply URL (Assertion Consumer Service URL)	https://dfn.local:8443/podium
Sign on URL	*Optional*
Relay State	*Optional*
Logout Url	*Optional*

(3) The following three core_env properties must be set (see notes immediately following):

```
authentication.mode=SAML

# this will log out of Qlik Catalog and MS AAD
# replace <tenant-id> and <HOSTNAME> (and possibly 8443 and qdc)
logout.url=https://login.windows.net/<tenant-
id>/oauth2/logout?post_logout_redirect_uri=https%3A%2F%2F<HOSTNAME>%3A8443%2Fqdc%
2Flogged-out

# replace <tenant-id> and <app-id>
saml.metadata.provider=https://login.microsoftonline.com/<tenant-
id>/federationmetadata/2007-06/federationmetadata.xml?appid=<app-id>
```

Notes on core_env properties:

- property "saml.metadata.provider" is taken from MS AAD setup, "**App Federation Metadata Url**"
- property "logout.url" was formerly "saml.logout.url" – it applies to both manually initiated logout via the menu, as well as session timeout
- property "logout.url", when configured for login.windows.net, now supports "post_logout_redirect_uri" to allow redirection from MS back to the Catalog logged-out page
- property "saml.entity.baseurl" is no longer set (May 2021 change)
- property "saml.keystore.path" is no longer set, and there is no longer a need to download the Base64 Certificate and add it to samlKeystore.jks – in fact, samlKeystore.jks is no longer present (May 2021 change)
- do **NOT** copy "**Logout URL**" from MS AAD into the core_env property "logout.url"
- there are only two valid choices for "logout.url": /logged-out (log out of only Qlik Catalog) or https://login.windows.net/<tenant-id>/oauth2/logout (log out of MS AAD and Qlik Catalog)
- **IMPORTANT**: when logging in to the UI, use URL "https://<HOSTNAME>:8443/qdc" and not "https://<HOSTNAME>:8443/qdc/login"

4.1.2 Enabling SAML using Okta

Instructions below are a reference with examples. Modifications will be required for client-specific SAML authentication and client environment. In this example setup, ***Okta*** is used as the Identity Provider (IDP) while podium is the Service Provider (SP).

1. Log in to your Okta organization as a user with administrative privileges. You can create a free Okta Developer Edition organization with your own email here: https://www.okta.com/developer/signup/.

2. Click on the blue Admin button on the top right corner.

3. Click on the Add Applications shortcut in the right panel.

4. Click on the green Create New **App** button.

5. In the dialog that opens, select the **SAML 2.0** option, then click the green **Create** button

6. In Step 1 General Settings, enter the application name (e.g., HostName SAML Application) in App name field, then click the green Next button.

7. In Step 2 Configure SAML, paste the URL below into the “Single Sign On URL" field – replace <HOSTNAME> with the Qlik Catalog hostname:

   https://<HOSTNAME>:8443/qdc

   Then, paste the URL below into the “Audience URI (SP Entity ID)” [old] or “Audience Restriction” [new] field:

   https://<HOSTNAME>:8443/qdc/saml2/service-provider-metadata/catalog

   Then, check the box “Allow this app to request other SSO URLs” and paste the URL below into the “Requestable SSO URLs” field:

   https://<HOSTNAME>:8443/qdc/login/saml2/sso/catalog

8. In Step 3 Feedback click the checkbox next to the text This is an internal application that we created then click the green **Finish** button.

9. You will now see the Sign On section of your newly created Spring Security SAML application

10. Copy the Identity Provider metadata link and paste it in the core_env.properties saml.metadata.provider. Example:

    saml.metadata.provider=https://dev-248822.okta.com/app/exk2z8xylfcOt2tRg4x7/sso/saml/metadata

11. You can also add additional users in the **People** section. All these users will need to be added to podium application as well with the same username.

12. Open the core_env.properties and add this line to it.
    authentication.mode=SAML

13. Restart the Qlik Catalog application (Tomcat).

There are now two ways to log into Qlik Catalog using Okta SAML Provider:

1. Log in to https://<HOSTNAME>:8443/qdc as usual. It will redirect you to Okta IDP from where you will have to authenticate using username/password. After successful authentication it will redirect to qdc.

   Important! A user with the same username must previously exist in Qlik Catalog.

2. Log in to your Okta account and on the home page, click on the application icon you just created. This will login to the Qlik Catalog application using the account you signed in with. If in the admin console, use the “My end user dashboard” from the top-right corner.

4.2 Tomcat SSL Configuration

Configure Tomcat to Support SSL (HTTPS)

NOTE: As of the May 2022 release, if the optional prerequisites script QDCprereqs.sh is used, Tomcat is ***automatically configured for HTTPS*** (including the generation of a self-signed certificate), and requests on port 8080 are automatically redirected to 8443.

The following steps are for customers that did not use the May 2022 (or more recent) QDCprereqs.sh script.

1. **Generate Keystore** - Use 'keytool' command to create a self-signed certificate. Optionally change the password from "changeit".

   Example:

   ```
   (qdc) # keytool -genkey -noprompt -validity 3650 -alias qdc -keyalg RSA -dname 'CN=Qlik Catalog,O=Qlik Technologies Inc.' -ext SAN=DNS:$(hostname) -ext EKU:c=serverAuth -keystore /usr/local/qdc/qdc.jks -storepass changeit -keypass changeit
   ```

   The above created a certificate (with alias 'qdc') located at '/usr/local/qdc/qdc.jks' (the filename that was provided in the keytool command).

   Certification Details
   Use same 'keytool' command to list the certificate’s details:

   Example:

   ```
   (qdc) # keytool -list -v -keystore /usr/local/qdc/qdc.jks -storepass changeit
   ```

2. **Add Connector in server.xml** - Locate your Tomcat’s server configuration file at $TOMCAT_HOME\conf\server.xml; modify it by adding a connector element to support https connections as follows:

   Under the existing Connector, which begins "<Connector port="8080"", add the following:

   ```
   <Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true" maxHttpHeaderSize="65536"
           server="Unknown Application Server"
           maxThreads="150" scheme="https" secure="true"
           clientAuth="false" sslProtocol="TLS"
           keystoreFile="${qdc.home}/qdc.jks"
           keystoreType="JKS"
           keystorePass="changeit"
           keyPass="changeit" />
   ```

   Note: The passwords (for both 'keystorePass' and 'keyPass') must be the same passwords given when the certificate was generated.

   Make sure that redirect port is available for the connector with the default (http) port you are using. For example, default server.xml may show:

   ```
   <Connector connectionTimeout="20000" port="8080" protocol="HTTP/1.1" redirectPort="8443"/>
   ```

HTTPS Redirect Configuration: 8080 is the http port and 8443 is the https port:

1. Start the server

2. Browse http://qdc-node-hostname:8080/qdc

3. If the application doesn’t redirect to https, create file $TOMCAT_HOME/webapps/qdc/WEB-INF/web.xml by either:

```
(qdc)    $ cp /tmp/podium/config/tomcat9-web.xml /usr/local/qdc/apache-tomcat-9.0.64/webapps/qdc/WEB-INF/web.xml
```

or manually adding the following content:

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://java.sun.com/xml/ns/j2ee"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd"
   version="2.4" id="WebApp_ID">
   <session-config>
      <session-timeout>60</session-timeout>
   </session-config>
   <listener>
      <listener-class>org.springframework.web.context.request.RequestContextListener</listener-class>
   </listener>
   <security-constraint>
      <web-resource-collection>
         <web-resource-name>Entire Application</web-resource-name>
         <url-pattern>/*</url-pattern>
      </web-resource-collection>
      <user-data-constraint>
         <transport-guarantee>CONFIDENTIAL</transport-guarantee>
      </user-data-constraint>
   </security-constraint>
</web-app>
```

4. Restart Tomcat

4.3 Configuration Recommendations for Qlik Catalog Installations

Single Node Settings for QVDs	
Hardware	
Memory	128G
CPU	32
core-env.properties file	
External.job.runner.spawn	true
external.job.runner.spawn.count	8
external.job.runner.enable	true
external.job.runner.process.rotation.period.hours	24
max.pool.size	96
hadoop.job.poolsize*	5
use.single.receiving.mapper	true

Single Node Settings for non-QVDs	
Hardware	
Memory	128G
CPU	32
core-env.properties file	
External.job.runner.spawn**	true
External.job.runner.spawn.count**	8
external.job.runner.enable	true
external.job.runner.process.rotation.period.hours	24
max.pool.size	96
hadoop.job.poolsize*	32
use.single.receiving.mapper	true

*If ingesting both QVDs and non-QVDs hadoop.job.poolsize must be set to 5
**These settings are mandatory for single-node which only runs in local mode; they are required for multi-node when using Spark for Prepare (due to user context being shared).

4.4 Enabling NextGen XML

As of the May 2022 release, NextGen XML (along with Tomcat HTTPS) is automatically enabled for first-time installs that use QDCprereqs.sh.

NextGen XML Containers

- To use the 'next-generation' XML support in Qlik Catalog, the Catalog installer (QDCinstaller.sh) installs two Qlik Sense containers: DCaaS (a connector lookup service) and a REST Connector (that parses XML files and converts them to flattened data).
- The NextGen XML Docker containers must be setup as a service to automatically start when Linux restarts.

The copy (cp) command below assumes the Qlik Catalog software (a.k.a., podium zip) has been expanded to /tmp -- the prerequisites section.

IMPORTANT:

a) If the Qlik Catalog NextGen XML directory is not /usr/local/qdc/dcaasIntegration or the service user/group is not “qdc,” the file **/etc/systemd/system/nextgen-xml.service** must be edited after the copy (cp) step below

b) If deploying to RHEL 8, Ubuntu 20.04, or Ubuntu 22.04 which use Podman to manage containers you must edit **/etc/systemd/system/nextgen-xml.service** after the copy (cp) step below:

a. **REMOVE** line 13 completely:

Requires=docker.service

b. EDIT line 14 and remove the following text: **docker.service**

```
(sudo)    # sudo cp /tmp/podium/config/nextgen-xml.service /etc/systemd/system/
(sudo)    # sudo systemctl daemon-reload
(sudo)    # sudo systemctl enable nextgen-xml.service
```

Qlik Catalog Configuration

Qlik Catalog must be configured to support NextGen XML.

- **core_env.properties**

  The core_env.properties file is located here: $TOMCAT_HOME/conf/core_env.properties.

  Locate the following property and replace $CATALOG_IPADDRESS with the IP address of the Qlik Catalog server:

  ○ base.xml.callback.url= http://$CATALOG_IPADDRESS:8082/qdc-xmlstore

  Note: the following two NextGen XML properties are also present in the core_env.properties file and are set automatically by the Qlik Catalog Installer:

- enable.new.xml.ingestion=true
- dcaas.connector.staging.dir=<QDC_HOME>/dcaasIntegration/dcaas-connector-staging

Modify Tomcat Web Server Configuration

NOTE: As of the May 2022 release, if the optional prerequisites script QDCprereqs.sh is used, the NextGen XML feature and Tomcat are ***automically configured to work together*** and this section can be ignored.

Manual configuration changes to the Tomcat web server are also required. Follow the instructions below to modify the Tomcat server.xml file to support NextGen XML:

1) Using a text editor, open **$TOMCAT_HOME/conf/server.xml:**

```
(qdc)   $ vi /usr/local/qdc/apache-tomcat-9.0.64/conf/server.xml
```

2) Add a dedicated Service and Connector

A dedicated service and connector are setup to listen on port 8082. The connector is configured to only allow localhost / local subnet connections (e.g., 127.0.0.1, 192.168.*.*, 172.16.*.*), thereby prohibiting outside access.

This new Service should come after the existing Service and is likely near the end of the document, near the closing </Server> tag. Here is the XML:

```
  <!-- The qdc-xmlstore directory is a temporary location for serving XML files to the DCaaS REST connector.
By default, it is configured
       to allow localhost and private IP access only. To allow access from an external host, add the host's IP
address with a pipe '|' separator
       to the end of the 'allow' value. For instance,
       allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1|...|11.111.3.241" -->

    <Service name="qdc-xmlstore">
        <Connector port="8082" protocol="org.apache.coyote.http11.Http11NioProtocol"
            connectionTimeout="20000" />
        <Engine name="Catalina" defaultHost="localhost">
            <Host name="localhost" appBase="qdc-xmlstore" unpackWARs="true" autoDeploy="true">
                <Context docBase="${qdc.home}/dcaasIntegration/qdc-xmlstore" path="/qdc-xmlstore">
                    <Valve className="org.apache.catalina.valves.RemoteAddrValve"
                        allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1|10\.\d+\.\d+\.\d+|172\.1[6-9]\.\d+\.\d+|172\.2[0-
9]\.\d+\.\d+|172\.3[0-1]\.\d+\.\d+|192\.168\.\d+\.\d+" />
                </Context>
            </Host>
        </Engine>
    </Service>
```

Port 8082 can be changed -- if changed, make sure to update core_env property base.xml.callback.url. The above Context is standard for NextGen XML and prohibits outside access.

4) Save the changes to server.xml and **restart Tomcat**

5) Restart NextGen XML docker containers:

```
(sudo)  # sudo systemctl restart nextgen-xml.service
```

6) To verify that Docker and its containers are running:

- Enter this on a command line: docker ps (or podman ps)
- Expected result: you should see the dcaas and data-connector-rest containers in the list of active containers.

Note: If SELinux is enabled (or other security framework), you may need to open port 50060 to allow routing between the two containers:

```
(sudo)  # sudo firewall-cmd --zone=public --permanent --add-port=50060/tcp
```

```
(sudo)  # sudo systemctl restart firewalld.service
```

4.5 Logging

As of May 2021, logging configuration changes, for single-node installations, are now made in file WEB-INF/classes/log4j2.xml rather than log4j.xml. Formerly, a logger entry looked like:

```
<logger name="com.nvs.ui.interceptor.LoggingInterceptor">
    <level value="info" />
</logger>
```

Now, in log4j2.xml, it looks like:

```
<Logger name="com.nvs.ui.interceptor.LoggingInterceptor" level="info" additivity="false">
   <AppenderRef ref="Console" />
</Logger>
```

4.6 Hive JDBC Driver

As of May 2021, the Hive JDBC driver is no longer included by default for single-node installations.

If you want to load from a Hive JDBC source, or publish to Hadoop+Hive, you will need to get the appropriate Hive JDBC driver for your cluster. The example below allows a single-node installation to interact with Hive running on a CDH 5.16.2 cluster.

Copy the JDBC driver jar(s) to a dedicated directory (also add the three log4j2 jars from WEB-INF/lib):

```
$ pwd
/usr/local/qdc/jdbcDrivers/hive-jdbc-cdh5.16.2

$ ls -la
total 51360
drwxr-xr-x   6 qdc   staff   204B Jan 29 17:25 .
```

```
drwxr-xr-x  7 qdc  staff   238B Jan 29 16:31 ..
-rw-r--r--  1 qdc  staff    23M Jan 29 16:30 hive-jdbc-1.1.0-cdh5.16.2-standalone.jar
-rw-r--r--  1 qdc  staff   285K Jan 29 17:21 log4j-api-2.13.2.jar
-rw-r--r--  1 qdc  staff   1.6M Jan 29 17:25 log4j-core-2.13.2.jar
-rw-r--r--  1 qdc  staff    23K Jan 29 17:18 log4j-slf4j-impl-2.13.2.jar
```

There is already a "HIVE" JDBC driver defined in the Admin “Database Drivers” section of the UI. It must be updated to point to the above directory:

```
update podium_core.pd_jdbc_source_info
    set alt_classpath='/usr/local/qdc/jdbcDrivers/hive-jdbc-cdh5.16.2' where sname='HIVE';
```

A Tomcat restart is required.

4.7 Migrating to or Upgrading Tomcat 9

Beginning with the May 2021 release, only Apache Tomcat 9 is supported. The installer will prohibit other versions. If using Tomcat 7, please first initiate a migration to Tomcat 9 before installing this release. Then, when installing, the upgrade option (-u) is ***NOT*** used.

These instructions may also be used to upgrade from an older version of Tomcat 9 to a newer version.

Step	**Sample Commands**
Shutdown and rename old Tomcat 7 or 9	cd /usr/local/qdc (or cd /usr/local/podium) ./apache-tomcat-<OLD_VERSION>/bin/shutdown.sh mv apache-tomcat-<OLD_VERSION> old-apache-tomcat
Download and expand Tomcat 9 - **NOTE**: **adjust version 9.0.64 to use latest 9.0.x series**	wget https://archive.apache.org/dist/tomcat/tomcat-9/v9.0.64/bin/apache-tomcat-9.0.64.tar.gz tar -xf apache-tomcat-9.0.64.tar.gz rm apache-tomcat-9.0.64.tar.gz
Copy core_env.properties from old Tomcat 7 to new Tomcat 9	cp old-apache-tomcat/conf/core_env.properties apache-tomcat-9.0.64/conf/
If migrating from Tomcat 7: Extract server.xml from podium.zip and copy to new Tomcat	unzip -j podium-4.15-xxxxx.zip podium/config/tomcat9-server.xml -d . mv ./tomcat9-server.xml apache-tomcat-9.0.64/conf/server.xml
If upgrading Tomcat 9: Copy server.xml from old Tomcat 9 to new Tomcat 9	cp old-apache-tomcat/conf/server.xml apache-tomcat-9.0.64/conf/ If the old Tomcat 9 was configured for HTTPS, and the keystore (jks file) was stored in the old Tomcat directory, migrate it to the new Tomcat directory, and update conf/server.xml to reference it. Consider

Step	Sample Commands
	placing the keystore file in a non-Tomcat directory such as /usr/local/qdc/keystore.
Configure QDCinstaller.properties for Tomcat 9	Whether using an existing QDCinstaller.properties file from a previous install, or configuring one for the first time, ensure that it is updated to point to Tomcat 9: TOMCAT_HOME=/usr/local/podium/apache-tomcat-9.0.64
Finally, when the installer is run, **do *NOT* specify upgrade mode (-u)**, as some files should be created as if it were a first-time install.	./QDCinstaller.sh
After installation, restore webapp web.xml, then restart Tomcat	cp old-apache-tomcat/webapps/qdc/WEB-INF/web.xml apache-tomcat-9.0.64/webapps/qdc/WEB-INF/

At this point, Tomcat 9, if newly installed, will support only HTTP on port 8080.

Verify successful Qlik Catalog startup and basic functionality.

Additional configuration will be required to configure HTTPS on port 8443, apply security headers, etc. If Tomcat 7 used HTTPS, the keystore (jks file) containing the public-private keypair should be copied to Tomcat 9 and conf/server.xml updated. For more information, see **Configure Tomcat to Support SSL**

In addition, Tomcat 7 may have been configured as a service. It should be disabled. Tomcat 9 may be configured as a service to automatically start. For more information, see setting up **Tomcat as a service**

4.8 Disabling Tomcat Redirect / SSL

As of the May 2022 release, when using the optional prerequisites script QDCprereqs.sh, Tomcat is automatically configured for HTTPS (including the generation of a self-signed certificate), and requests on port 8080 are automatically redirected to 8443.

In certain situations, these may need to be disabled (e.g., when the SSL connection is terminated at a load balancer). Perform either or both of the following and then restart Tomcat.

Disable Redirect from 8080 to 8443

Edit <tomcat home>/webapps/qdc/WEB-INF/web.xml

Remove the entire "<security-constraint>" element.

Disable HTTPS Port 8443

First, disable the redirect (see above).

Edit <tomcat home>/conf/server.xml

Remove the entire "<Connector>" element which begins "<Connector port="8443""

4.9 Upgrading PostgreSQL

These instructions may be used to upgrade to the latest version of Qlik Catalog custom PostgreSQL. Reminder: there is no requirement to upgrade PostgreSQL when upgrading Catalog. If a PostgreSQL upgrade is planned, please do so before upgrading Catalog.

Step	Sample Commands
Shutdown Tomcat	./bin/shutdown.sh
Backup databases. Default password for podium_md and podium_dist is "nvs2014!" For licenses is "licenses". For postgres is "postgres". If "ERROR: out of shared memory", see note below table.	pg_dump -U podium_md --format=c --file=podium_md.sqlc podium_md pg_dump -U podium_dist --format=c --file=podium_dist.sqlc podium_dist pg_dump -U licenses --format=c --file=licenses.sqlc licenses
Find the Linux PostgreSQL service file name. Will be something like "qdc_pg.service" or "qdc_pg-11.2.service" -- **USE THIS NAME FOR NEXT TWO STEPS**	ls -la /etc/systemd/system/
Stop the service and verify PostgreSQL is not running	sudo systemctl stop qdc_pg-11.2.service ps -ef | grep "postgres:"

Remove the service	sudo systemctl disable qdc_pg-11.2.service sudo rm /etc/systemd/system/qdc_pg-11.2.service
Prepare for new PostgreSQL -- **adjust version as needed**	sudo mkdir -p /usr/pgsql/qdc11-16 sudo chown -R postgres:postgres /usr/pgsql sudo mkdir -p /var/lib/pgsql/11-16/qdc_data sudo chown -R postgres:postgres /var/lib/pgsql
Install new PostgreSQL -- **adjust versions as needed**	cp podium-4.15-xxxxx_Linux_X64.zip /tmp/ cd /tmp unzip podium-4.15-xxxxx_Linux_X64.zip sudo su - postgres /tmp/podium/thirdParty/qdc_pg11-16_RHEL7-and-CentOS7.bsx **OR** /tmp/podium/thirdParty/qdc_pg11-16_RHEL8-and-CentOS8.bsx **OR** /tmp/podium/thirdParty/qdc_pg11-16_Ubuntu.bsx **exit**
Recreate symlinks	sudo rm /usr/bin/psql sudo ln -s /usr/pgsql/qdc11-16/bin/psql /usr/bin/psql psql --version sudo rm /usr/bin/pg_dump sudo ln -s /usr/pgsql/qdc11-16/bin/pg_dump /usr/bin/pg_dump pg_dump --version sudo rm /usr/bin/pg_restore sudo ln -s /usr/pgsql/qdc11-16/bin/pg_restore /usr/bin/pg_restore pg_restore --version
Recreate Linux service	sudo cp /tmp/podium/config/qdc_pg.service /etc/systemd/system/ sudo systemctl daemon-reload sudo systemctl enable qdc_pg.service

Ensure Tomcat service depends on new PostgreSQL service -- two references should be to "qdc_pg.service"	sudo vi /etc/systemd/system/tomcat.service
Start the service and verify PostgreSQL is running	sudo systemctl start qdc_pg.service ps -ef | grep "postgres:"
Restore podium_md. Default password for podium_md is "nvs2014!" For postgres is "postgres".	INTERNAL_DB_PASSWORD=nvs2014! psql template1 -U postgres -c "create role podium_md with encrypted password '$INTERNAL_DB_PASSWORD' createdb login;" psql template1 -U podium_md -c "create database podium_md;" pg_restore -U podium_md --dbname=podium_md podium_md.sqlc
Restore podium_dist. Default password for podium_dist is "nvs2014!" For postgres is "postgres". The last step may produce an error/warning that can be ignored.	INTERNAL_DB_PASSWORD=nvs2014! psql template1 -U postgres -c "create role podium_dist with encrypted password '$INTERNAL_DB_PASSWORD' superuser createdb login;" psql template1 -U podium_dist -c "create database podium_dist;" psql podium_dist -U podium_dist -f /tmp/podium/config/podium_dist_su_ddl.txt pg_restore -U podium_dist --dbname=podium_dist podium_dist.sqlc psql podium_dist -U podium_dist -f /tmp/podium/config/podium_dist_ddl.txt psql podium_dist -U postgres -c "ALTER USER podium_readonly WITH ENCRYPTED PASSWORD '$INTERNAL_DB_PASSWORD' LOGIN NOSUPERUSER;"
Restore licenses. Default password for licenses is "licenses" For postgres is "postgres".	psql template1 -U postgres -c "create user licenses with password 'licenses'" psql template1 -U postgres -c "create database licenses owner licenses" pg_restore -U licenses --dbname=licenses licenses.sqlc
Proceed to upgrade Catalog	

When performing a pg_dump or pg_restore of podium_dist, if you see the following:

```
pg_dump: [archiver (db)] query failed: ERROR:  out of shared memory
HINT:  You might need to increase max_locks_per_transaction.
pg_dump: [archiver (db)] query was: LOCK TABLE
podium_demo_muabbas.pd_role_str_parent IN ACCESS SHARE MODE
```

then PostgreSQL must be configured with additional resources:

Step	Sample Commands
Stop PostgreSQL -- adjust version as needed	sudo systemctl stop qdc_pg-11.13.service
Become postgres user and edit configuration file -- adjust version as needed	sudo su - postgres vi /var/lib/pgsql/11-13/qdc_data/postgresql.conf

Increase these two settings, then quit vi	shared_buffers = 2GB
	max_locks_per_transaction = 4096

Exit postgres user	exit
Start PostgreSQL -- adjust version as needed	sudo systemctl start qdc_pg-11.13.service

About Qlik

Qlik is on a mission to create a data-literate world, where everyone can use data to solve their most challenging problems. Only Qlik’s end-to-end data management and analytics platform brings together all of an organization’s data from any source, enabling people at any skill level to use their curiosity to uncover new insights. Companies use Qlik products to see more deeply into customer behavior, reinvent business processes, discover new revenue streams, and balance risk and reward. Qlik does business in more than 100 countries and serves over 48,000 customers around the world.

qlik.com