

Red Hat Upgrade Guide

Quest Data Quality V3.1.3

Prepared by



Introduction	1
Software Requirements (Auto Install)	3
Pre-Installation Setup	4
Configuration Setup	5
Details about the config file	5
Installation Process	6
Post-Installation Procedure	6

Introduction

This document provides a comprehensive, step-by-step process for upgrading Quest DQ on Red Hat Server 9 from version 3.1.0 to version 3.1.3. It is designed for personnel with technical knowledge of Red Hat Server 9 and Linux Operating Systems.

The guide walks you through the installation steps, ensuring that you have a clear understanding of the process and any prerequisites specific to Red Hat. By following this guide, you will be able to successfully set up the Quest DQ 3.1.3 application and utilize its features in your Red Hat Enterprise Linux 9 environment.

Note: This guide applies only to upgrades from version 3.1.0 to 3.13 and does not support upgrades from version 2.4.6 to 3.1.3.

System Requirements

This section provides the minimum system and mandatory requirements that are needed to install the Quest DQ application in the Linux environment successfully.

Category	Recommended
Operating system	Red Hat Enterprise Linux 9.x (Supported on Azure, AWS, and on-prem VMs) If AWS, use Amazon Machine Image (AMI) (HVM) with SSD Volume Type.
Processor	64-bit processor
Disk Space	100 GB, and it should be /directory or Quest DQ installing folder, and the install user home directory should have at least 5GB

Package	Core and RAM Specifications
Bronze	4 Core 8 GB RAM
Silver	4 Core 8 GB RAM
Gold	8 Core 16 GB RAM
Platinum	16 Cores 32 GB RAM
Titanium	32 Cores 64 GB RAM

Postgres Server Prerequisites (Only in case of DB Isolated Deployments)

Operating System	Red Hat 9.x
CPU Core	4 Cores or more
RAM	8GB or more

- Ensure to take a snapshot of the server before attempting to upgrade
- Dedicated Server:** Quest DQ needs to be installed on a dedicated server.
- Internet Access:**

URL	Purpose	Required during
https://license-ga.dqlabs.cloud	This URL must be whitelisted to activate and manage the validity of the license key.	Required after Installation
https://s3.amazonaws.com	This URL must be whitelisted to allow binaries to be downloaded from the DQLabs repository.	Required only before installation. The file can be downloaded externally and moved to the server if needed
cdn.redhat.com subscription.rhsm.redhat.com	Official RHEL repositories are accessed to download and update the necessary packages Command	These repositories can be limited after updates and before installation
pypi.org pypi.python.org files.pythonhosted.org pythonhosted.org	These URL is required to download the Python packages.	Required after Installation

4. **Ports to be opened:** Ports Used for Internal Communication within the Application:

PostgreSQL	5432 (Mandatory)
Airflow	8080
HTTP	80
HTTPS	443

Software Requirements (Auto Install)

The requirements mentioned below are auto-installed with the script; the user should not manually install any of the software requirements in the DQServer

Services	Version
PostgreSQL	15.14
Python	3.9.18
Java	openjdk 17.0.16
Airflow	2.8.1
Drivers	MSSQL, Oracle, PostgreSQL, MySQL ODBC/JDBC
Spark	3.5.1

Before proceeding with the upgrade, ensure the following:

1. No third-party or external applications should be present on the server.

Pre-Installation Setup

Step 1: Log in to the Application server with appropriate privileges and install wget

```
None
sudo yum install wget -y
```

Step 2: Take a backup of Prerequisites and Erwin-installer from the prior installation location

```
None
sudo mv prerequisites prerequisites_bck
sudo mv Erwin-installer Erwin-installer_bck
```

Step 3: Download the Prerequisite File from S3 into the directory where the application should be installed

```
None
wget
https://s3.us-east-1.amazonaws.com/erwin-2.0/code/linux/application-code/3.1.3/On_premise/Packages_3.1.3/Erwin-Redhat-Upgr
ade-Packages/Erwin-prerequisites.tar
```

Step 4: Execute the command below and verify the checksum is the same

```
None
sha256sum Erwin-prerequisites.tar
```

The above command should return the value
23fa5ffe46e5fbbc6fb6acf0698bb048ff182713c446447ef411a749822d16e9 Erwin-prerequisites.tar.
 If the code fails to match, do not proceed with the deployment.

Step 5: Extract the Downloaded Tar File

```
None
sudo tar -xvf Erwin-prerequisites.tar
```

Step 6: Execute the following commands:

```
None
#Navigate to the p7zip_file folder and Install the p7zip Plugin
cd prerequisites/p7zip_file

#Install the p7zip plugin:
sudo rpm -ivh *.rpm

#Remove the reduntant tar file
sudo rm -rf ~/Erwin-prerequisites.tar

#Return to Prerequisites Folder
cd ~
```

Step 7: Download the Installation File by using the commands below into the directory where the application should be installed

```
None
wget
https://s3.us-east-1.amazonaws.com/erwin-2.0/code/linux/application-code/3.1.3/On_premise/Packages_3.1.3/Erwin-Redhat-Upgr
ade-Packages/Erwin-installer.7za
```

Step 8: Execute the command below and verify that the checksum is the same

```
None
sha256sum Erwin-installer.7za
```

The above command should return the value
bcf7f65be93ae4d9beccf6c837763e634bfbb1060de443d71e2a0aa5f29bb5be Erwin-installer.7za.
 If the code fails to match, do not proceed with the deployment.

Step 9: Extract the Installation File by using the commands below:

```
None
sudo 7za x Erwin-installer.7za
```

Step 10: Remove the redundant zip file after extraction

```
None
sudo rm -rf Erwin-installer.7za
```

Configuration Setup

Step 1: Navigate to the prerequisites directory

```
None

#If the application is installed on home directory
cd prerequisites/

#If the application is installed on a custom directory
cd <custom_directory>/prerequisites/
```

Ensure that the `config.txt` and `Erwin-installer.sh` are in the same directory

```
[dqlabs@Redhat310To313Upgrade prerequisites]$ ls
config.txt  Erwin-installer.sh  p7zip_file
[dqlabs@Redhat310To313Upgrade prerequisites]$
```

Step 2: Edit the `config.txt` file using the command below:

```
None
sudo vi config.txt
```

Step 3: Update the `config.txt` file by clicking “**I**” to get into insert mode

Details about the config file

- 1. Source and Destination Locations:** Please verify that the **SOURCE_LOCATION** path specified in the `config.txt` is accurate and ensure the installation file is downloaded correctly to that location.
 - SOURCE_LOCATION:** Define the source location path.
Example - (/home/ec2-user/Erwin-installer)
 - DESTINATION_LOCATION:** Provide DQ V3.1.0 destination location(Refer DQ V3.1.0 config file)
Example - (/home/ec2-user/erwindq)
- 2. Database Configuration:**
 - DB_SEPARATION:** yes or no
 - yes** - if the deployment is a dual-server architecture
 - no** - if the deployment is a single server architecture

```
LOCAL_REPO=yes

# shellcheck disable=SC2034
SOURCE_LOCATION=/home/dqlabs/Erwin-installer
DESTINATION_LOCATION=/home/dqlabs/app

# Database Configuration
# Set to "yes" if database is on a separate server, "no" for single server installation
DB_SEPARATION=no
```

Step 4: Go to command mode by pressing the escape key. Save and exit the editor using **:wq!** and press Enter

Installation Process

Step 1: Set Permissions and execute the script using the commands below:

```
None

#Grant permissions to the script
sudo chmod 777 Erwin-installer.sh

#Execute the setup script
./Erwin-installer.sh
```

Step 2: After the script has executed, verify that the following output appears at the end of the execution

```
Phase 10: Finalization
Upgrade completed - restarting all services...
Phase 10 completed: Finalization
=====
UPGRADE COMPLETED SUCCESSFULLY!
=====

Upgrade Summary:
• Previous Version: 3.0.4
• New Version: 3.1.3
• Backup Location: /home/dqlabs/app/backup/backup_3.1.3
• Installation Path: /home/dqlabs/app
• Database Mode: Local
• Repository Mode: Local RPM

Services Status:
• DQLabs Server: Running
• Airflow Webserver: Running
• Airflow Scheduler: Running
• HTTP Server: Running
• Livy Service: Enabled

Next Steps:
1. Verify all services are running correctly
2. Test the application functionality
3. Verify database connectivity (local/remote mode)
4. Monitor system performance

Important Notes:
• Database configuration is preserved from fresh installation
• Airflow settings are restored from backup
• No database schema changes are performed

=====
DQLabs 3.1.3 upgrade completed!
=====
```

Note: The execution log will be stored on the server and can be accessed using the following commands:

```
None

#View the logs in 3.1.3 directory
vi script.log
```

Update Postgres Config (Only applicable for DB Separated Deployment)

Step 1: Log in to the Postgres installed Server

```
None

#Switch to root user
sudo -i

#Open the PostgreSQL configuration file
sudo vi /var/lib/pgsql/15/data/postgresql.conf
```

Step 2: Locate the listen_addresses parameter, press i to enter insert mode, update the variable “max_connections” from 100 to 500

```
port = 5432                                # (change requires restart)
max_connections = 100                       # (change requires restart)
#superuser_reserved_connections = 3         # (change requires restart)
unix_socket_directories = '/var/run/postgresql' # comma-separated list of directories
#unix_socket_group = ''                    # (change requires restart)
#unix_socket_permissions = 0777            # begin with 0 to use octal notation
#bonjour = off                             # (change requires restart)
#bonjour_name = ''                         # defaults to the computer name
#                                          # (change requires restart)

# - TCP settings -
# see "man tcp" for details

#tcp_keepalives_idle = 0                    # TCP_KEEPIDL, in seconds;
#tcp_keepalives_interval = 0               # 0 selects the system default
#tcp_keepalives_count = 0                  # TCP_KEEPCNT;
#                                          # 0 selects the system default
#tcp_user_timeout = 0                     # TCP_USER_TIMEOUT, in milliseconds;
#                                          # 0 selects the system default
```

Step 3: Save and exit the file(Press Esc, then type :wq! and hit Enter)

Step 4: Restart Postgres Services


```
None
sudo service postgresql-15 restart
```

Update SSL Config (Only applicable for HTTPS Deployment)

Step 1: Open the ssl.conf file

```
None
#Open the PostgreSQL configuration file
sudo vi /etc/httpd/conf.d/ssl.conf
```

- Step 2: Comment the line “Listen: 443”
- Step 3: Save and exit the file(Press Esc, then type :wq! and hit Enter)
- Step 4: Restart httpd service

```
None
sudo service httpd restart
```

Post-Installation Procedure

After completing the installation and verifying the successful setup of Quest DQ on your RedHat machine, you can now log in to the application and proceed with the final steps. Follow the instructions below:

- Step 1: Launch any supported web browser on your machine
- Step 2: In the address bar of the browser, enter the IP address or DNS name used during installation
- Step 3: The browser will load the Quest DQ application, and you will be presented with the login page

