

# Redhat Deployment Guide

## erwin Data Quality V3.1.3

Prepared by



<b>Introduction</b>	<b>1</b>
<b>Software Requirements (Auto Install)</b>	<b>2</b>
PostgreSQL Installation (Only applicable for DB isolated Deployment)	2
Pre-Installation Setup	4
Configuration Setup	5
Details about the config file	5
Verification of Installation	7
Post-Installation Procedure	8

# Introduction

This document provides a comprehensive, step-by-step process for installing DQLabs on RHEL 9. It is designed for personnel with technical knowledge of RHEL 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 RHEL. By following this guide, you will be able to successfully set up the DQLabs application and leverage its features in your RHEL environment.

## System Requirements

This section provides the minimum system and mandatory requirements needed to install the DQLabs 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	Minimum 100 GB, and it should be in the <code>/directory</code> or the DQLabs installation folder, and the install user home directory should have a minimum of 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	Redhat 9.x
CPU Core	4 Cores or more
RAM	8GB or more

1. **Dedicated Server:** DQLabs needs a dedicated server for installation
2. **Internet Access:**

URL	Purpose	Required during
<a href="https://s3.amazonaws.com">https://s3.amazonaws.com</a>	This URL must be whitelisted to allow downloading binaries from the DQLabs repository.	Required only before installation. The file can be downloaded externally and moved to the server if needed
<a href="cdn.redhat.com">cdn.redhat.com</a> <a href="subscription.rhsm.redhat.com">subscription.rhsm.redhat.com</a>	Official RHEL repositories are accessed to download and update the necessary packages Command	These repositories can be limited after updates and before installation
<a href="https://license-qa.dqlabs.cloud">https://license-qa.dqlabs.cloud</a>	To activate the license key for the Product	To activate the product license key: <ul style="list-style-type: none"><li>For online deployments: Activate using the provided key.</li><li>For offline deployments: Request an offline key by sharing the MAC address of the server where the product will be installed.</li></ul>

3. **License key:** A new DQ license key is required for activating the product upon installing the product

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

PostgreSQL	5432 (Mandatory)
Airflow	8080
HTTP	80
HTTPS	443
Livy	8998

5. Postgres Installation: Suppose the Postgres database needs to be isolated from the server on which the application is installed. PostgreSQL 15.14 needs to be installed on the second server, and the user account must be granted all necessary privileges.

Steps to install the PostgreSQL database (Only applicable for DB Separated Deployment)

Follow the Pre-Installation Setup, Configuration Setup, and Installation Process defined below on the database server.

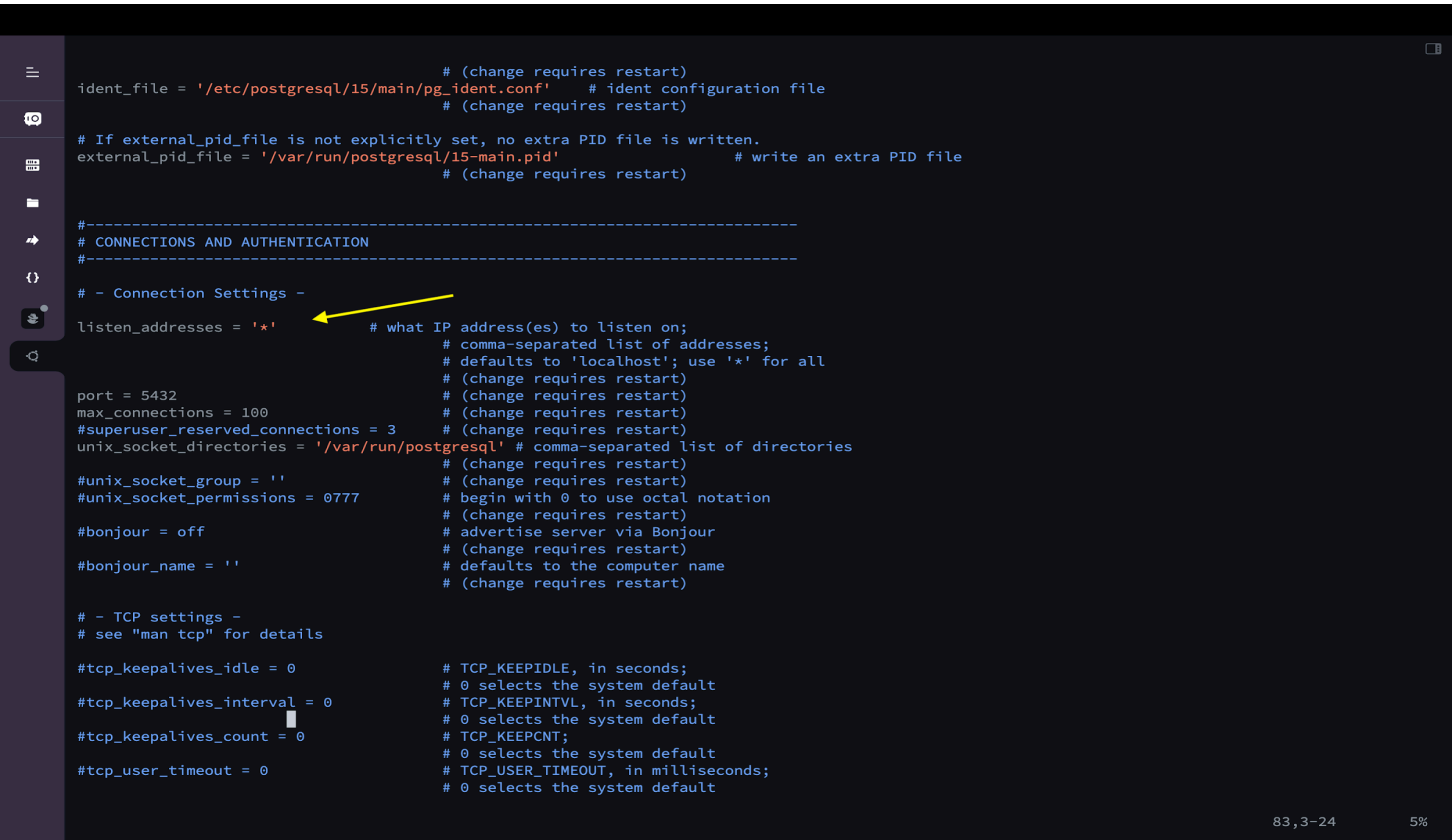
Changes to the PostgreSQL Configuration Directory to accept the PostgreSQL 15 instance remote connections.

Step 1: Update postgresql.conf file

```
None
#Switch to root user
sudo -i

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

- Locate the listen\_addresses parameter, press i to enter insert mode, then uncomment and update its value as shown in the image below
- Update the variable “max\_connections” from 100 to 500



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

Step 2: update pg\_hba.conf file

```
None
sudo vi /var/lib/pgsql/15/data/pg_hba.conf
```

- Add the main server’s IP addresses to allow connections by replacing <Application\_server\_privateIP>, press i to enter insert mode, and copy-paste the line as shown in the image below.

None

```
host all all <Application_server_privateIP>/32 md5
```

```
# "all", "sameuser", "samerole" or "replication" makes the name lose
# its special character, and just match a database or username with
# that name.
#
# This file is read on server startup and when the server receives a
# SIGHUP signal. If you edit the file on a running system, you have to
# SIGHUP the server for the changes to take effect, run "pg_ctl reload",
# or execute "SELECT pg_reload_conf()".
#
# Put your actual configuration here
# -----
#
# If you want to allow non-local connections, you need to add more
# "host" records. In that case you will also need to make PostgreSQL
# listen on a non-local interface via the listen_addresses
# configuration parameter, or via the -i or -h command line switches.


# DO NOT DISABLE!
# If you change this first entry you will need to make sure that the
# database superuser can access the database using some other method.
# Noninteractive access to all databases is required during automatic
# maintenance (custom daily cronjobs, replication, and similar tasks).
#
# Database administrative login by Unix domain socket
local all postgres md5

# TYPE DATABASE USER ADDRESS METHOD

# "local" is for Unix domain socket connections only
local all all md5
# IPv4 local connections:
host all all 127.0.0.1/32 md5
host all all 27.100.26.74/32 md5
host all all 49.206.114.53/32 md5
host all all 100.25.38.189/32 md5
# IPv6 local connections:
host all all ::1/128 md5
# Allow replication connections from localhost, by a user with the
# replication privilege.
local replication all peer
host replication all 127.0.0.1/32 md5
:wq!
```

- Save and exit the file(Press **Esc**, then type **:wq!** and hit **Enter**)
- Restart PostgreSQL: Restart the PostgreSQL service to apply the new configuration

None

```
sudo service postgresql-15 restart
```

By following the above steps, you will have configured your PostgreSQL 15 instance to accept remote connections.

# 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	3.11.11
Java	openjdk 17.0.16
Airflow	2.8.1
Drivers	MSSQL, Oracle, PostgreSQL, MySQL ODBC/JDBC
Spark	3.5.1

Before proceeding with Single Server Deployments, ensure the following:

1. The server is free from any pre-installed applications.
2. PostgreSQL should not be manually installed for single-server deployments.
3. No third-party or external applications should be present on the server.

## Pre-Installation Setup

**Step 1:** Install the wget command using the following command:

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

**Step 2:** 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-Packages/Erwin-prerequisites.tar
```

**Step 3:** Execute the command below and verify the checksum is the same

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

The above command should return the value  
**071a0b80318de20b7aca4c20b1bc6e01d5bf20806e77f088e024c9e1fa356558 Erwin-prerequisites.tar**  
If the code fails to match, do not proceed with the deployment.

**Step 4:** Extract the Downloaded Tar File

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

**Step 5:** 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 6:** Download the Installation File by using the commands below in 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-Packages/Erwin-installer.7za
```

**Step 7:** 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 8: Extract the Installation File by using the commands below:

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

Step 9: Remove the redundant zip file after extraction

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

## Configuration Setup

Step 1: Navigate to the Pre-requisites 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 in the prerequisites folder.

```
dqlabs@ubuntuarun:~$ cd prerequisites/
dqlabs@ubuntuarun:~/prerequisites$ ls
Erwin-installer.sh  config.txt
```

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. **LOCAL\_REPO**=yes or no  
Ensure the Internet is accessible if you choose “no”
2. **User Configuration:**
  - a. **DEDICATED\_USER**: yes or no
    - i. If yes, then update the following:
      1. **SERVICE\_USER**: dquser
      2. **SERVICE\_GROUP**: usergroup
    - ii. If no, then the application will be installed using the installer user and group
3. **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.
  - a. **SOURCE\_LOCATION**: Define the source location path.  
Example - (/home/dqlabs/Dqlabs-installer)
  - b. **DESTINATION\_LOCATION**: Define the destination location path.  
Example - (/usr/src/Dqlabs)

4. **DQLabs Access Information:**



- a. **DNS\_CONFIGURE:** yes or no
  - i. If yes, then update the following:
    - 1. **DNS\_NAME:** e.g., <http://foo.subdomain.com>
    - 2. **SSL\_PROTOCOL:** http or https
  - ii. If no, then update the following:
    - 1. **ACCESS\_MODE:** public or private
    - 2. **SSL\_PROTOCOL:** http or https

5. Administrator Information:

- a. **ADMIN\_EMAIL:** Administrator's email address (Ensure valid email format)  
Example - (admin@dqlabs.ai)
- b. **ADMIN\_PASSWORD:** Administrator's password  
Example - (Dql@b\$)

6. PostgreSQL Configuration:

- i. **PG\_USERNAME=<username>**
- ii. **PG\_PASSWORD=<password>**
- iii. **PG\_DB\_NAME=<dbname>**
- iv. **AIRFLOW\_USERNAME=<username>**
- v. **AIRFLOW\_PASSWORD=<password>**
- vi. **AIRFLOW\_DB\_NAME=<dbname>**

Do not use special characters and spaces in your username, dbname, and passwords; Do not use the same username for both Server and Airflow

7. Installation Options(Mandatory):

- a. **FULL\_INSTALLATION:**
  - i. **yes ->** Install DQLabs Application along with Postgres database.
  - ii. **no ->** Install only the Postgres database.
  - iii. **dqlabs ->** Install DQLabs without a Postgres database.

In case of Postgres Installation on the secondary server, provide the input as no

8. Dual server Deployment (Update only in case of db-separation deployment and if the input to the 7th variable is dqlabs)

- a. **PG\_PORT\_NO=<port no>**
- b. **PG\_HOST=<postgres server IP>**

```
##### Erwin #####

# Do you want to install packages from Local Repo (yes) or RHEL Repo (no)
LOCAL_REPO=yes

### Dedicated Service User Configuration
### Set DEDICATED_USER to "yes" for script will create separete user and group or "no" for Using installer user and group mode
DEDICATED_USER=yes
SERVICE_USER=dqlabsuser
SERVICE_GROUP=dqlabsgroup

# Please provide the Erwin-installer file path as source location (example: /home/ec2-user/Erwin-installer)
SOURCE_LOCATION=/home/dqlabs/Erwin-installer

# Please provide the destination location (example: /usr/src/Dqlabs)
DESTINATION_LOCATION=/home/dqlabsuser/app

# DQLabs access Information

# DNS configuration:
# If a DNS name is available, set DNS_CONFIGURE to yes and provide the DNS value below.
# Otherwise, leave DNS_CONFIGURE as no.
DNS_CONFIGURE=no
DNS_NAME=

# Specify the access mode:
# For public IP, set ACCESS_MODE to 'public'.
# For private IP, set ACCESS_MODE to 'private'.
ACCESS_MODE=public

# Specify the protocol to use (mandatory): http or https.
SSL_PROTOCOL=http

# Login Inputs:
# Provide the administrator email and password for DQLabs access.
ADMIN_EMAIL=admin@dqlabs.ai
ADMIN_PASSWORD=dqlabs

# Postgres credentials to be created and used during DQLabs installation.
PG_USERNAME=dquser
PG_PASSWORD=dqpass
PG_DB_NAME=dqdb

# Airflow credentials and database details.
AIRFLOW_USERNAME=airuser
AIRFLOW_PASSWORD=airpass
AIRFLOW_DB_NAME=airdb

# Installation options (mandatory):
# Enter one of the following values:
# "yes" -> Install DQLabs Application along with Postgres database.
# "no" -> Install only the Postgres database.
# "dqlabs" -> Install DQLabs without Postgres database.
FULL_INSTALLATION=yes

### (Below variables are applicable if FULL_INSTALLATION is set to "dqlabs")
PG_PORT_NO=
PG_HOST=
```

**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

#Navigate to the prerequisites directory
cd prerequisites

#Grant permissions to the script
sudo chmod 755 Erwin-installer.sh

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

Users without password-based authentication should not execute the script.

## Verification of Installation

Once the installation process is complete, it is important to verify that the DQLabs application and related services are running successfully on your Red Hat machine. Perform the following verifications:

**Step 1:** Ensure the following output is shown at the end of the script execution:

```
airflow-webserver is running.
airflow-scheduler is running.
apache2 is running.
dqlabs.service is running.
livy.service is running.
true
*****
All services are running. DQ installation completed successfully!
*****
/home/dqlabs/DQ/dqlabsenv/lib/python3.11/site-packages/drf_yasg/___init____.py:2: UserWarning: pkg_resources is deprecated as an API. See https://setuptools.pypa.io/en/latest/pkg_resources.html. The pkg_resour
ces package is slated for removal as early as 2025-11-30. Refrain from using this package or pin to Setuptools<81.
  from pkg_resources import DistributionNotFound, get_distribution
***** Airflow Token *** YWRtaW46YWRtaW4=
default
dqlabs@m20buntu310Freshinstall:~$
```

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

```
None

#Navigate to prerequisites directory
cd prerequisites

#View the logs
vi script.log
```

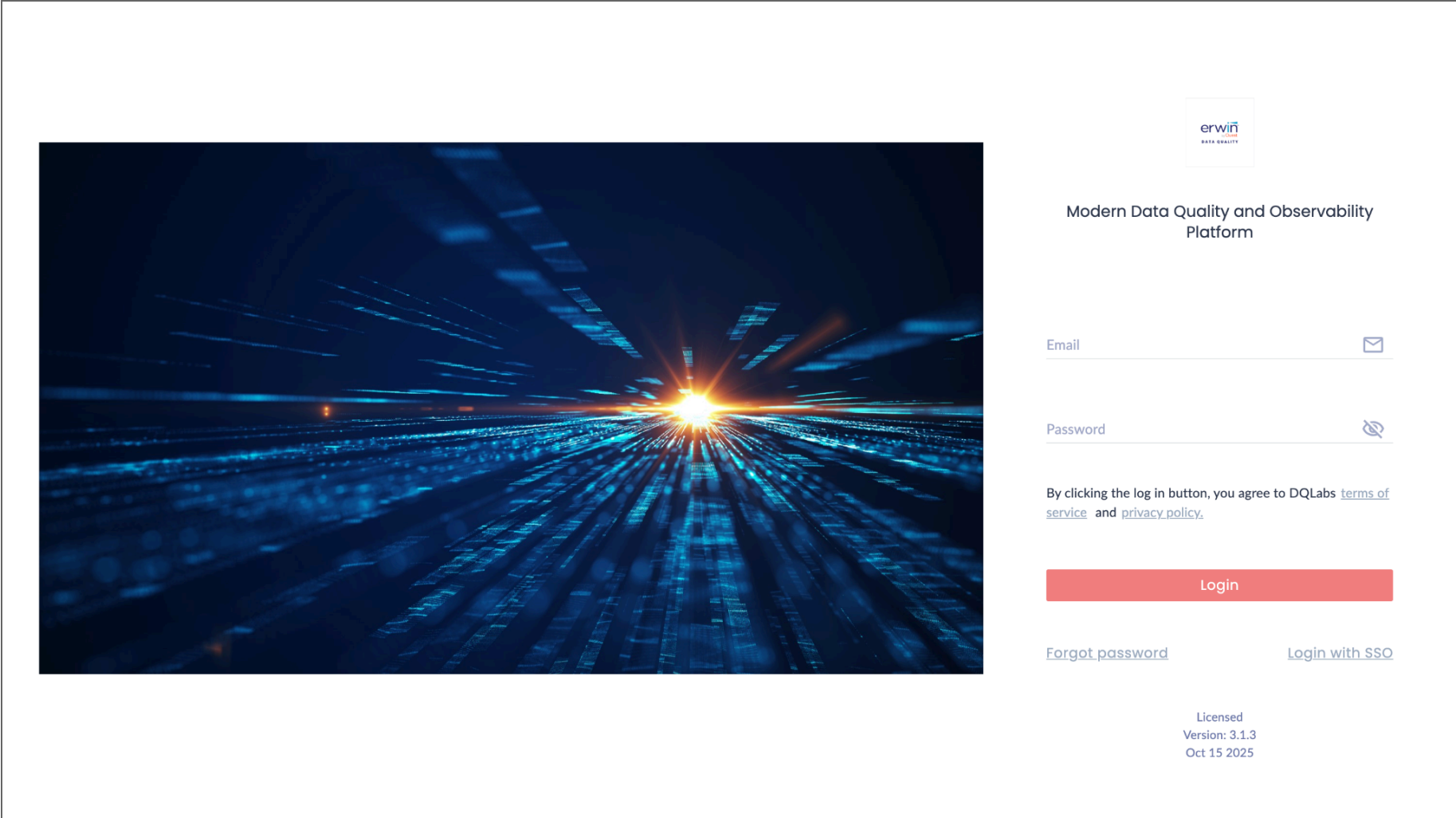
## Post-Installation Procedure

After completing the installation and verifying the successful setup of DQLabs 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 DQLabs application, and you will be presented with the login page



**Step 4:** After logging in to the application, you will be prompted to add the license. Once the license has been activated, the platform is ready to use.

