Ubuntu Deployment Guide

Quest Data Quality V3.1.3

Prepared by

Introduction 1

Software Requirements (Auto Install) 2

PostgreSQL Installation (Only applicable for DB isolated Deployment) 2

Pre-Installation Setup 5

Configuration Setup 6

Details about the config file 6

Installation Process 7

Verification of Installation 8

Post-Installation Procedure 8

Introduction

This document provides a comprehensive, step-by-step process for installing Quest DQ on Ubuntu Server 22.04. It is designed for personnel with technical knowledge of Ubuntu Server 22.04 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 Ubuntu. By following this guide, you will be able to successfully set up the Quest DQ application and leverage its features in your Ubuntu Server 22.04 environment.

System Requirements

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

Category	Recommended
Operating system	Ubuntu Server 22.04 LTS (HVM)
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	Ubuntu 22.04
CPU Core	4 Cores or more
RAM	8GB or more

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

URL	Purpose	Required during
https://s3.amazonaws.com	This URL must be whitelisted to allow binaries to be downloaded from the Quest DQ repository.	Required only before installation. The file can be downloaded externally and moved to the server if needed
Ubuntu/Debian Official Repositories	Official Ubuntu repositories are accessed to download and update the necessary packages Command: `sudo apt-get install wget && sudo apt-get update && sudo apt-get install p7zip-full -y`	These repositories can be limited after updates and before installation
https://license-qa.dqlabs.cloud	To activate the license key for the Product	To activate the product license key: ● For online deployments: Activate using the provided key. ● For offline deployments: Request an offline key by sharing the MAC address of the server where the product will be installed.

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

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)

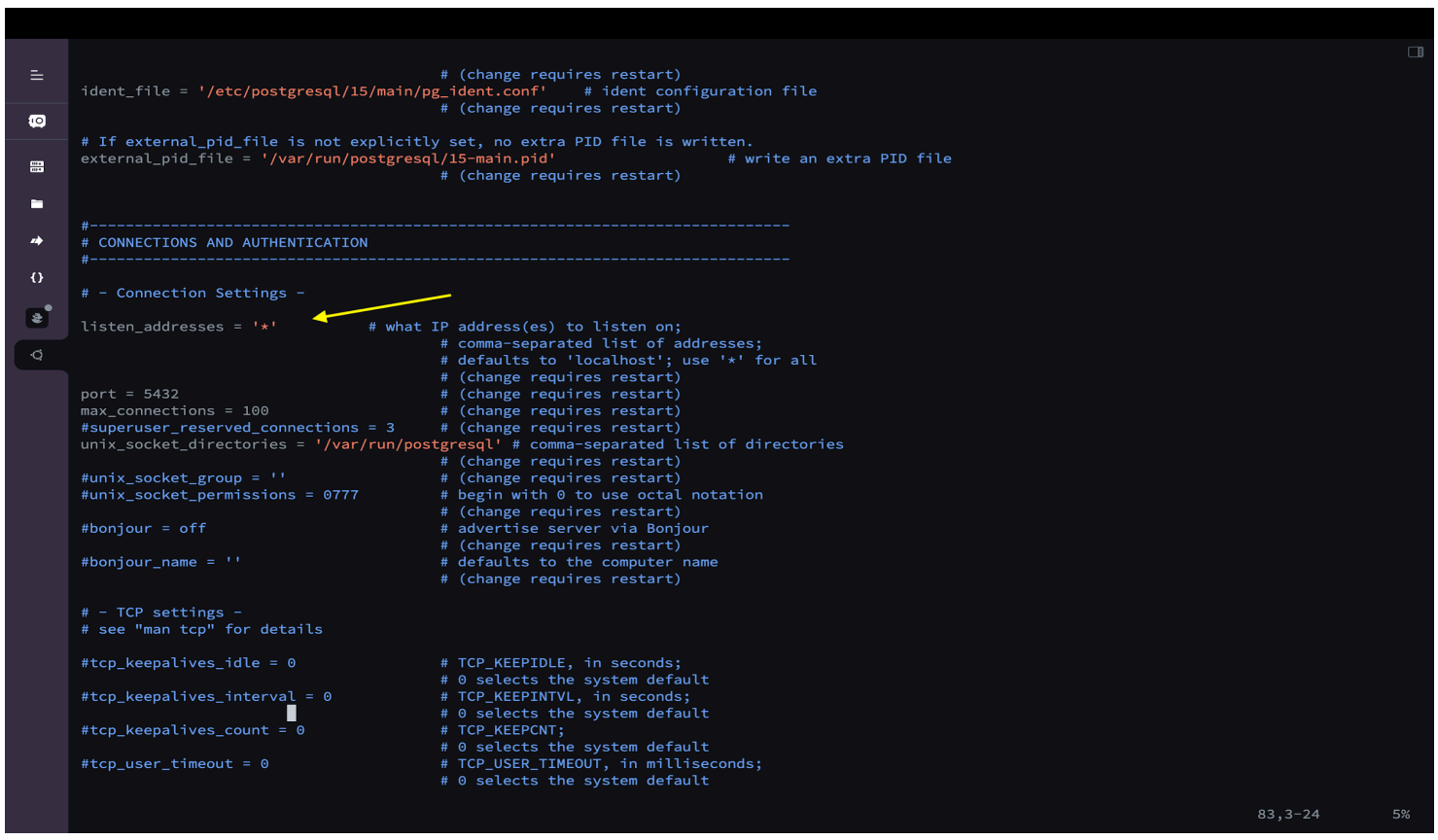
Follow the Pre-Installation Setup, Configuration Setup, and Installation Process defined below on the database server.

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

Step 1: Update postgresql.conf file

```
None

#Open the PostgreSQL configuration file
sudo vi /etc/postgresql/15/main/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`



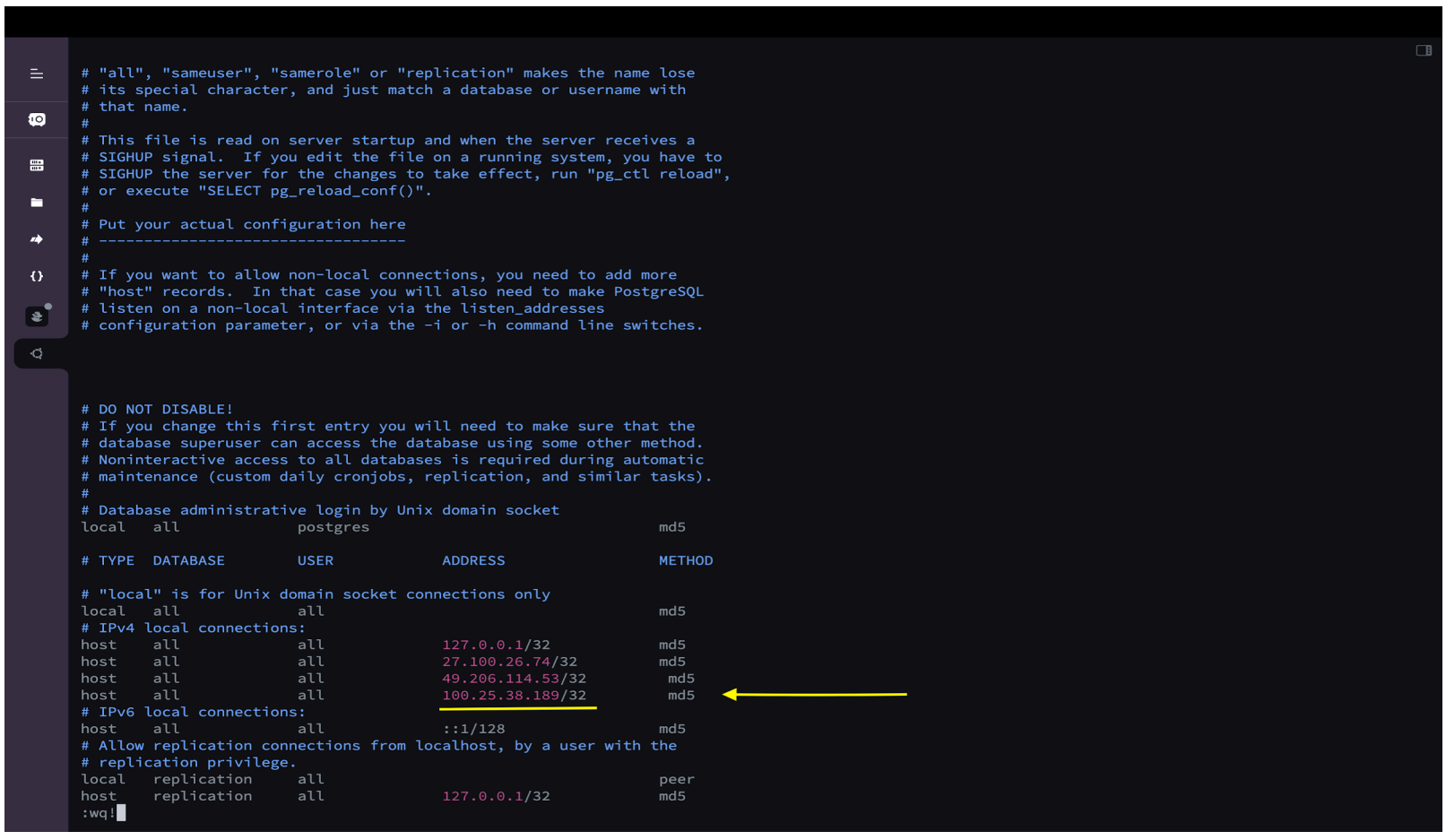
- Save and exit the file(Press `Esc`, then type `:wq` and hit **Enter**)

Step 2: update pg_hba.conf file

```
None
#Open the pg_hba.conf file
sudo vi /etc/postgresql/15/main/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
```



- 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 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	3.10.12
Java	OpenJDK 17.0.16
Airflow	2.8.1
Drivers	MSSQL, Oracle, PostgreSQL, MySQL ODBC/JDBC
Spark	3.5.4

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: Log in to the Application server with appropriate privileges and install wget & the p7zip Plugin using the below command:

```
None
sudo apt-get install wget && sudo apt-get update && sudo apt-get install p7zip-full -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-Ubuntu-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
6238052f4caeb4c4cd7a49b1c3da78e9f32e8699a500b3b67ff94ce2a902a26c 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: Remove the redundant tar file after extraction

```
None
sudo rm -rf Erwin-prerequisites.tar
```

Step 6: 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-Ubuntu-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

1352bb99ba7ae7a7d17bbda53d22a2caca90a6130ffee089f791f787f22f417d 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

```
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/ubuntu/Erwin-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., 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 seperate 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 prerequsites directory
```

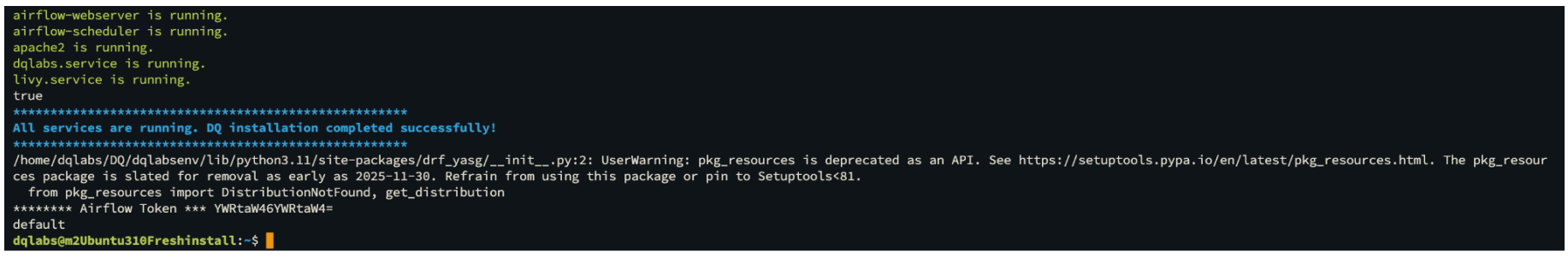
```
cd prerequisites

#Grant permissions to the script
sudo chmod 777 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, verifying that the application and related services are running successfully on your Ubuntu machine is important. Perform the following verifications:

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



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

```
None
#Navigate to prerequisites directory
cd prerequisites

#View the logs
vi script.log
```

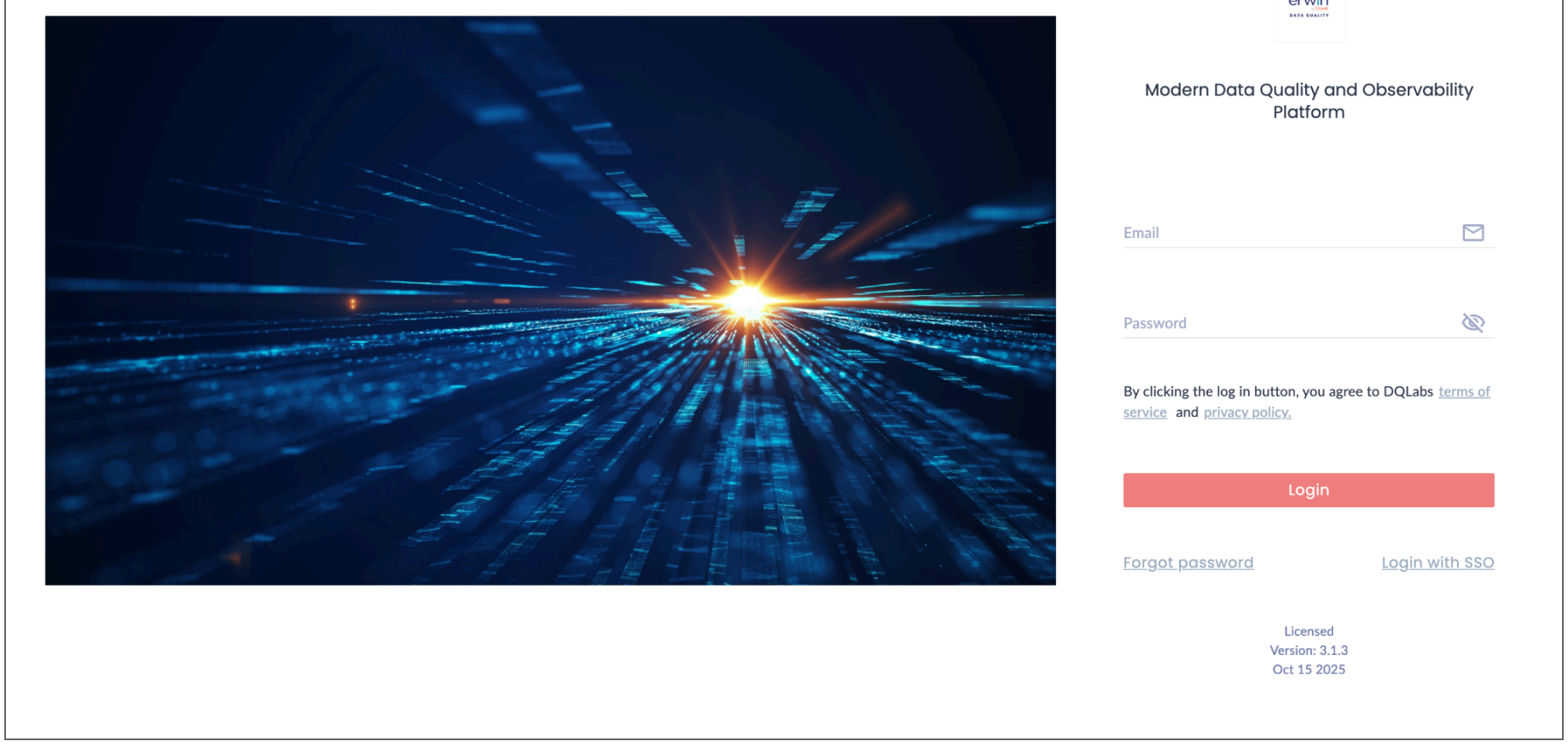
Post-Installation Procedure

After completing the installation and verifying the successful setup of on your Ubuntu 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 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.