erwin Data Intelligence – erwin Data Quality

Windows Deployment Guide

Legal Notices

Contact erwin

Understanding your Support

Review support maintenance programs and offerings.

Registering for Support

Access the erwin support site and click Sign in to register for product support.

Accessing Technical Support

For your convenience, erwin provides easy access to "One Stop" support for erwin Data Intelligence (erwin DI), and includes the following:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- erwin Support policies and guidelines
- Other helpful resources appropriate for your product

For information about other erwin products, visit http://erwin.com/.

Provide Feedback

If you have comments or questions, or feedback about erwin product documentation, you can send a message to distechpubs@erwin.com.

erwin Data Modeler News and Events

Visit www.erwin.com to get up-to-date news, announcements, and events. View video demos and read up on customer success stories and articles by industry experts.

Contents

Introduction

This document provides a comprehensive, step-by-step process for installing DQLabs in a Windows environment. It is designed for personnel with technical knowledge of Windows Operating Systems. This guide provides step-by-step instructions for installation, ensuring you understand the process and any Windows-specific prerequisites. By following it, you can successfully set up the DQLabs application and utilize its features in your 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	Windows Server 2022 64-bit
Processor	64-bit processor
Disk Space	Minimum 100 GB (C drive is not recommended, and ensure the disk is dedicated only to DQLabs)
Nested Virtualization	Enabled

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

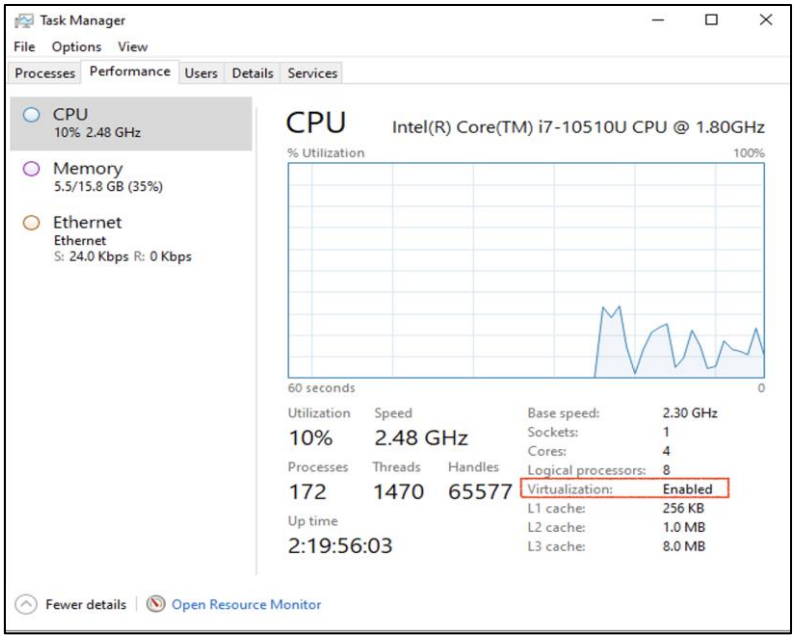
Postgres Server Prerequisites (Only in case of DB Isolated Deployments)

Operating System	Windows Server 2022 64-bit
CPU Core	4 Cores or more

RAM	8GB or more

Virtualization Checks

- To check if Virtualization is enabled, go to Start > Task Manager> Performance and check for virtualization enabled as shown below:



- If the Windows 2022 server is hosted on an Azure VM and Virtualization is not enabled, run the below command from PowerShell to enable it.

```
Enable-WindowsOptionalFeature -Online -FeatureName VirtualMachinePlatform
```

PS: For non-Azure servers, please reach out to your server admins to enable the Virtualization

Pre-requisites

1. **Dedicated Server**: DQLabs needs a dedicated server for installation (Windows OS should be up to date).
2. **Internet Access**:

URL	Purpose	Required during
https://license-qa.dqlabs.cloud	This URL must be whitelisted to activate and manage the validity of the license key. In case of an Offline license, raise a request with the MAC ID of the server	Required after Installation

https://s3.us-east-1.amazonaws.com/erwin-2.0/code/windows/application-code/3.1.0/Dqlabs-offline-installer-py-3.11.zip	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

3. **License key**: A new DQ license key is required for activating the product upon installing the product
4. **User:** Install using a Service account, and the DQLabs application must be maintained under the same service account. Services must be restarted only from the installed user account.
5. **Administrator Privileges**: The user should be able to run Windows PowerShell ISE with administrator privileges
6. **Dedicated Drive**: Any drive other than C
7. **Ports to be opened**: Ports Used for Internal Communication within the Application:

PostgreSQL	5432 (Mandatory)
Airflow	8080
HTTP	80
HTTPS	443

Software Requirements (Manual)

PostgreSQL Installation (Only applicable for DB isolated Deployment)

Skip this Step if the application and PostgreSQL should be installed on the same Server.

Follow the steps below to install PostgreSQL on your secondary server:

Step 1: Log in to the Secondary Server and ensure the user account is provided with the necessary privileges.

Step 2: Download the binary and extract the zip file.

Details about the config file

Step 3: Open the CONFIG.txt file with administrator access and update the following:

Server Configuration

A. **drive**= Define the drive letter where PostgreSQL has to be installed
Example - (F)
B. **dqlabsserverip**= Define the DQLabs application server private IP for communications
Example - (10.10.10.2)
C. **PSQL_Host**= Define the Postgres database server private IP (If postgres needs to be installed on the same server, provide the same server IP)
D. **PostgresMasterusername**=<Postgres_username>

Example – postgres

E. PostgresMasterPassword=<Postgres_password>
Example – postgres

Note: If the Postgres server is installed using the DQLabs script, the username and password input should be 'postgres'.

DQLabs Access Information

F. **DNS_NAME:**nodns

G. **ACCESS_MODE**: public or private

   a. Public: When the server is connected to the internet, it is public. Here, the user will be able to connect to the DQLabs application outside the network.
   b. Private: When the server is connected to the intranet, it is private. Here, the user will be able to connect to the DQLabs application using the organization network.

H. **SSL_PROTOCOL:** http or https
If the server is SSL-certified, provide the input as https, else http

Note: If the user has provided the DNS_name, ACCESS_MODE needs to be empty.

Administrator Information

I. **ADMIN_EMAIL:** Administrator's email address (Ensure valid email format)
Example - (admin@dqlabs.ai)

J. **ADMIN_PASSWORD**: Administrator's password
Example - (Dql@b$)

Postgres database credentials

(Do not use special characters and spaces in your username, dbname, and passwords)

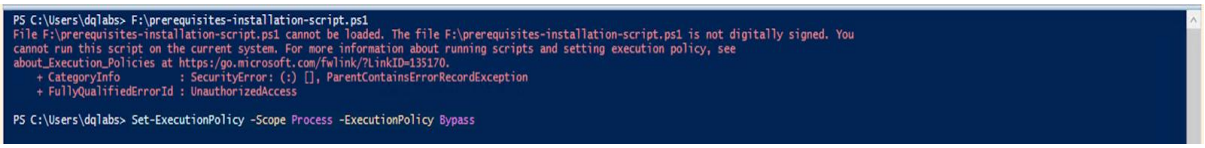
K. **PG_USERNAME=**<Postgres_username>
L. **PG_PASSWORD=**<Postgres_password>
M. **PG_DB_NAME=**<databse_name>
N. **PG_PORT_NO=**5432
O. **AIRFLOW_USERNAME=**airflow_user
P. **AIRFLOW_PASSWORD=**airflow
Q. **AIRFLOW_DB_NAME=**airflow_db
R. **DATABASE_ONLY_INSTALLATION=**yes
yes -> Install only the Postgres database
no -> Install DQLabs application along with Postgres database
dqlabs -> Install DQLabs without Postgres database

Step 4: Open PowerShell ISE as an administrator. (Press the Win key and search for PowerShell ISE > Right Click on Windows PowerShell ISE and click on "Run as administrator").

Step 5: In the PowerShell ISE window, click on File > Open. Locate the prerequisites installation script.

Step 6: Click the “Run Script” button in PowerShell (as shown in the reference image below). If the script throws a digitally not signed error, run the following command and select yes to all.

```
Set-ExecutionPolicy -Scope Process -ExecutionPolicy Bypass
```



Step 7: Run the script. The user gets a prompt to confirm the installation type, confirm the type, and provide the input as yes.

Software Requirements (Auto Install)

The below-mentioned requirements are auto-installed with the script, the user should not manually install any of the software requirements in the DQServer. Manually installed software may conflict with the script, leading to failure.

Services	Version
PostgreSQL	15.11
Python	3.11.9
IIS Server	NA
Airflow	2.8.1
Drivers	MSSQL, Oracle, PostgreSQL, MySQL ODBC/JDBC
Java	11 Open JDK 64-bit
Spark	3.5.1

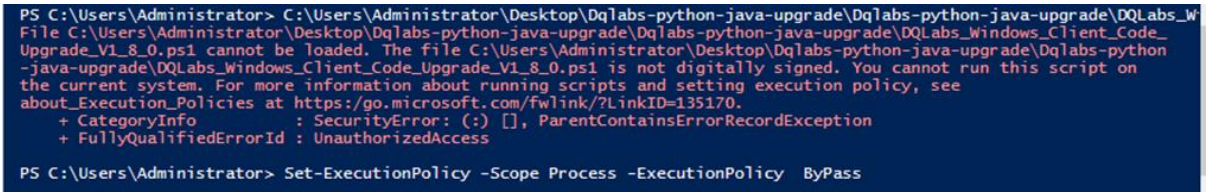
Before proceeding with Single Server Deployments, ensure the following:

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

Please note that at any point during the execution of Scripts, if the user faces the below error, execute the below command:




```
#Execute the below command to digitally sign the script
Set-ExecutionPolicy -Scope Process -ExecutionPolicy Bypass
```

Step 1: Log in to the application server and download the binary to the server.

```
https://s3.us-east-1.amazonaws.com/erwin-2.0/code/windows/application-code/3.1.0/Dqlabs-offline-installer-py-3.11.zip
```

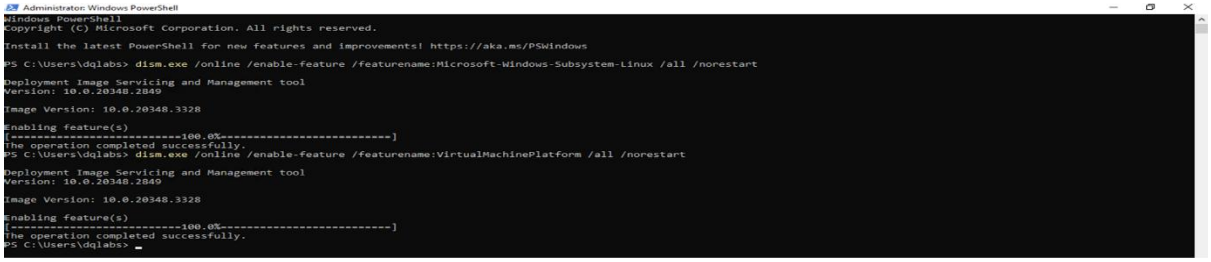
Step 2: Move the downloaded file to the drive where DQLabs should be installed (Other than C) and extract the Zip file.

WSL and Ubuntu Install

Step 3: Enable the Windows Subsystem for Linux using the following steps

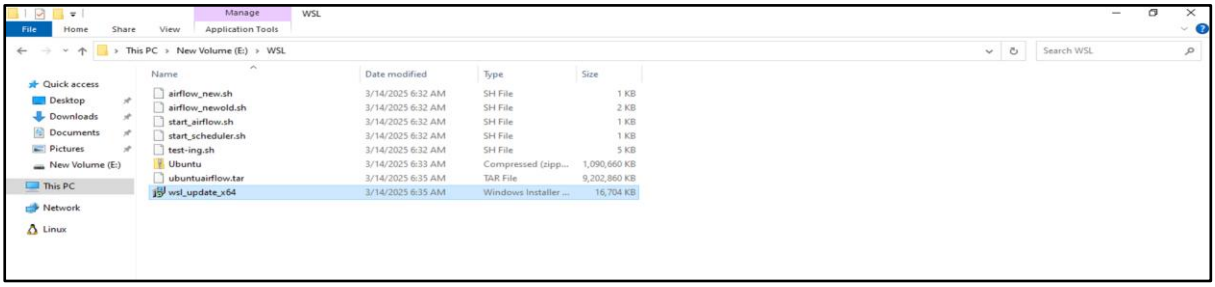
- Open Command Prompt as an Administrator (Start menu > Command Prompt > Run as Administrator) and run the following commands one after the other.

```
dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart


dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart
```

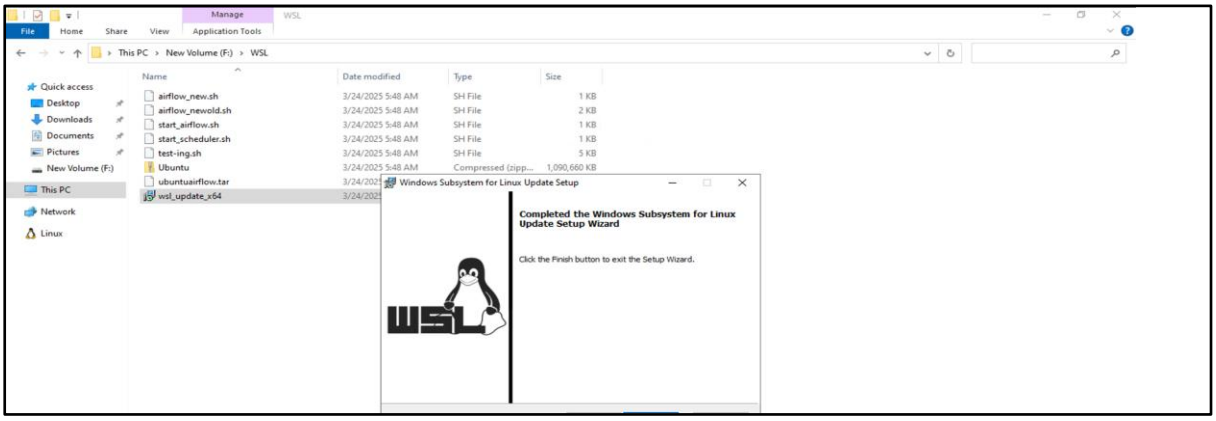


Step 4: Restart your machine to complete the WSL installation.

Step 5: Navigate to the extracted directory > WSL.



Step 6: Double-click on wsl_update_X64 to install > Next > click Finish to complete the installation.



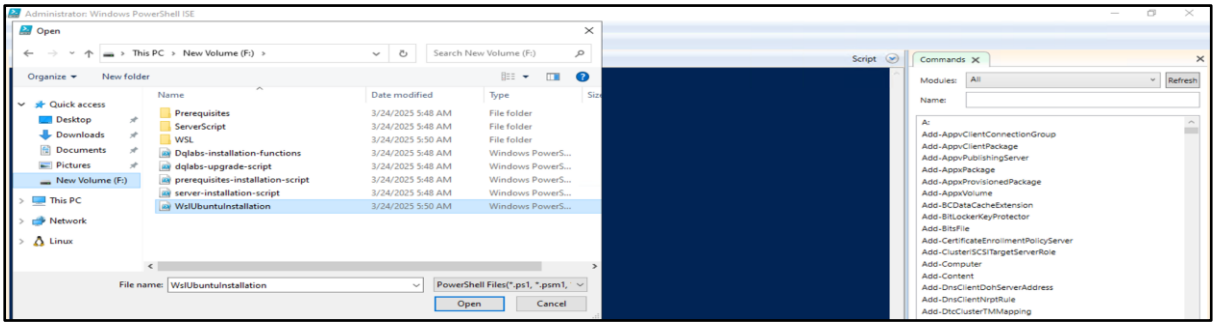
Step 7: Installation steps for Ubuntu

Online Installation

- Open the command prompt as an administrator.
- Run the command **wsl –update**.

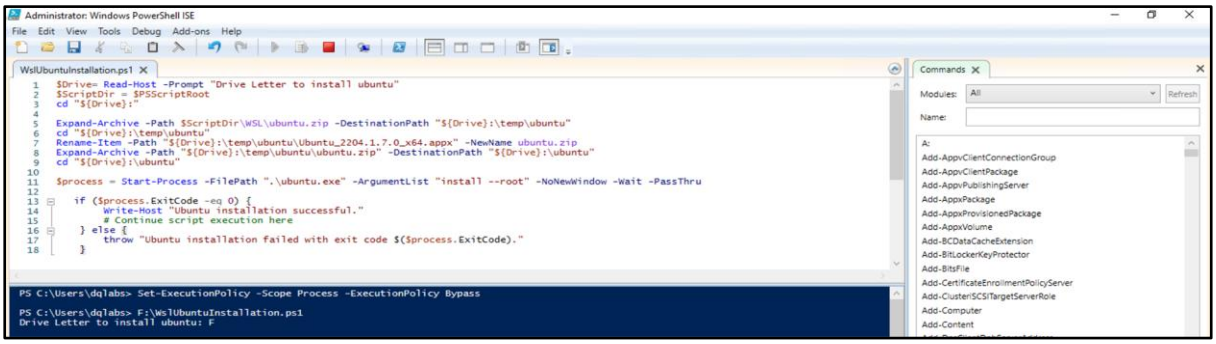
Offline Installation

- Open “PowerShell ISE” as an administrator (Press Win key and search for “PowerShell ISE”)
- In the PowerShell ISE window, click on File > Open and open the WslUbuntuInstallation script from the extracted directory



Note: The user should not change the location of the script from the zip file.

- Enter the Disk Drive name where the DQLabs application will be installed (For Example, F as shown in the image above). The best practice is to install it on any drive other than C.



- Once Ubuntu is installed, the user will get a "Ubuntu installation successful" message

Configuration Setup

Step 1: Open the CONFIG.txt file with administrator access

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

Server Configuration

A. **drive=** Define the drive letter where DQLabs has to be installed(Any drive other than C, do not use colon)
   Example - (F)
B. **dqlabsserverip=** Define the DQLabs server private IP
   Example - (10.10.10.2)
C. **PSQL_Host=** Define the Postgres database server private IP (If PostgreSQL needs to be installed on the same server, provide the same server private IP)
D. **PostgresMasterusername=**<Postgres_username>
   Example - postgres
E. **PostgresMasterPassword=**<Postgres_password>
   Example – postgres

Note: If the Postgres server is installed using the DQLabs script, the username and password input should be 'postgres'

DQLabs Access Information

F. **DNS_NAME**:e.g., http://foo.subdomain.com
G. **ACCESS_MODE**: public or private
   a. Public: When the server is connected to the internet, it is public. Here, the user will be able to connect to the DQLabs application outside the network.
   b. Private: When the server is connected to the intranet, it is private. Here, the user will be able to connect to the DQLabs application using the organization network.

H. **SSL_PROTOCOL:** http or https
   If the server is SSL-certified, provide the input as https, else http

Note: If the user has provided the DNS_name, ACESS_MODE needs to be empty

Administrator Information

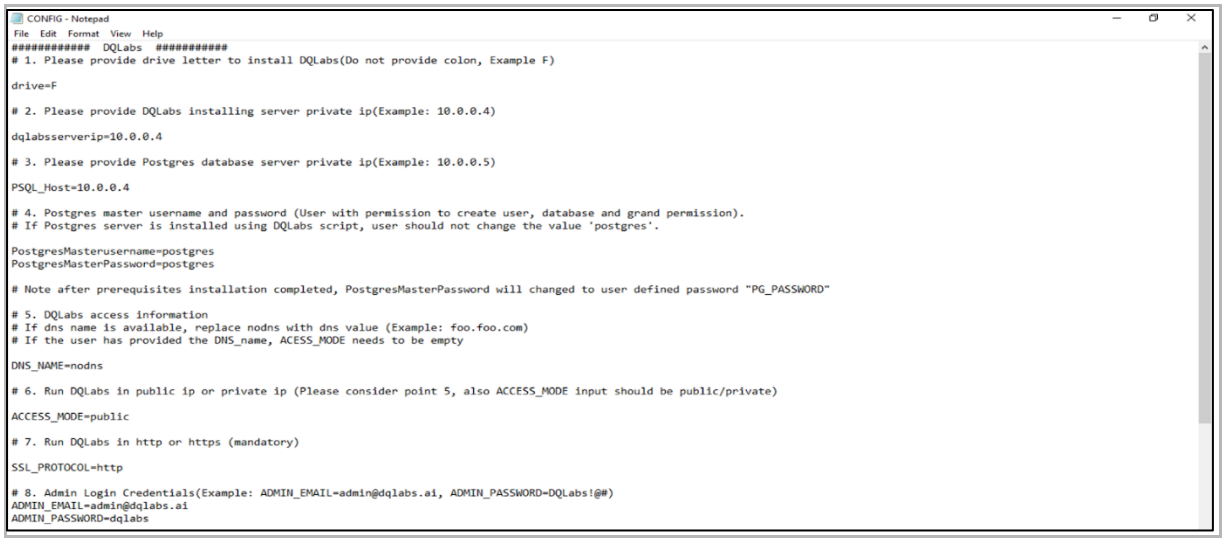
I. **ADMIN_EMAIL:** Administrator's email address (Ensure valid email format)
Example - (admin@dqlabs.ai)

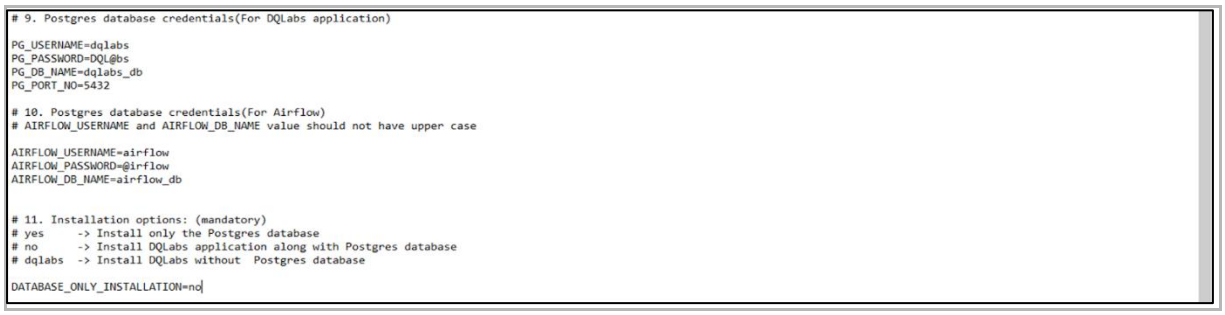
J. **ADMIN_PASSWORD**: Administrator's password
Example - (Dql@b$)

Postgres database credentials

(Do not use special characters and spaces in your username, dbname, and passwords**)**

K. **PG_USERNAME**=<Postgres_Username>
L. **PG_PASSWORD**=<Postgres_Password>
M. **PG_DB_NAME**=<Database_name>
N. **PG_PORT_NO**=5432
O. **AIRFLOW_USERNAME**=airflow_user
P. **AIRFLOW_PASSWORD**=airflow
Q. **AIRFLOW_DB_NAME**=airflow_db
R. **DATABASE_ONLY_INSTALLATION**=
yes -> Install only the Postgres database
no -> Install DQLabs application along with Postgres database
dqlabs -> Install DQLabs without Postgres database





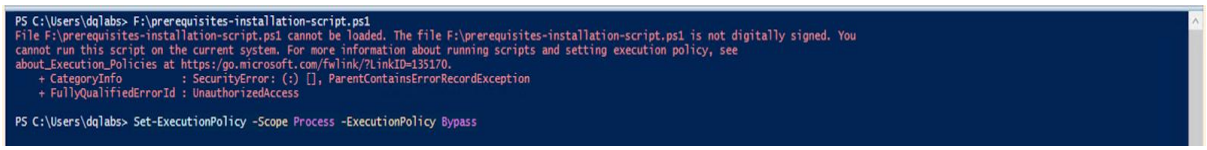
Prerequisites Deployment

Step 1: Open PowerShell ISE as an administrator. (Press the Win key and search for PowerShell ISE > Right Click on Windows PowerShell ISE and click on “Run as administrator”).

Step 2: In the PowerShell ISE window, click on File > Open, locate the prerequisites installation script.

Step 3: Click the “Run Script” button in PowerShell (as shown in the reference image below). If the script throws a digitally not signed error, run the following command and select yes to all.

```
Set-ExecutionPolicy -Scope Process -ExecutionPolicy Bypass
```



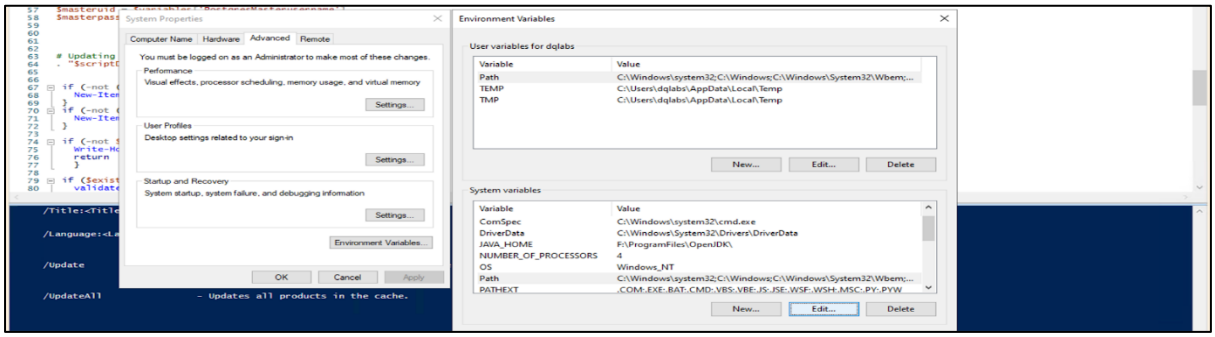
Step 4: Run the script. The user gets a prompt to confirm the installation type, confirm the type, and provide the input as yes.

Step 5: Once the script execution is successful, the following status message will appear.

Prerequisite Deployment – Validation

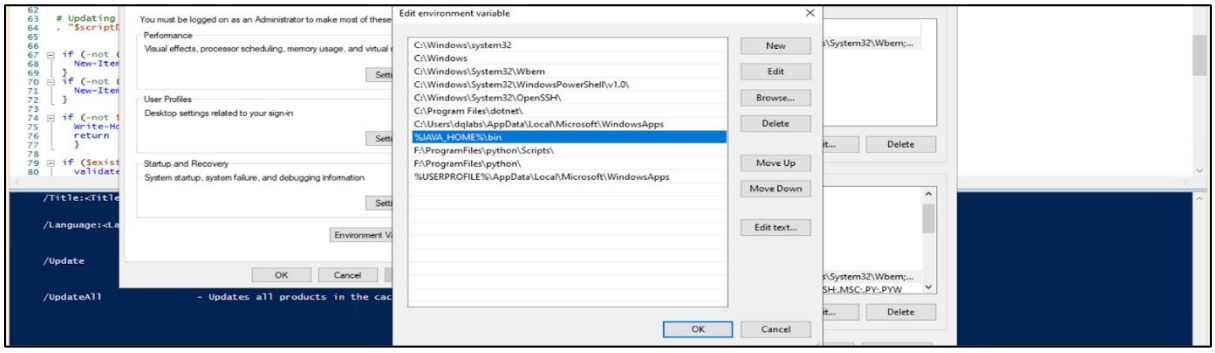
Step 1: Environment Variables

- Click on Start > Search for Advanced System settings.
- On the Advanced tab, click Environment Variables. On the Environment Variables window, select Path under the System Variables section and click on Edit



Step 2: Java Home Check

- Ensure you see “JAVA_HOME” as shown in Fig, then click on ‘OK’ and exit.



Step 3: Java and Python Version check

- Open a new Command Prompt window with Administrator privileges.
- Type java --version and hit enter. You should see Java 11 as the major version number.

```
C:\Users\dqlabs>java --version
openjdk 11.0.18 2023-01-17
OpenJDK Runtime Environment OpenLogic-OpenJDK (build 11.0.18+10-adhoc..jdk11u)
OpenJDK 64-Bit Server VM OpenLogic-OpenJDK (build 11.0.18+10-adhoc..jdk11u, mixed mode)
```

- Then type python -V and hit enter to check the Python version. You should see 3.10 as the major version number.

```
C:\Users\dqlabs>python -V
Python 3.11.9
```

Please note that if you do not see the correct Java major version and Python major version, do not proceed with the installation.

Step 4: PostgreSQL Login

- Click on the Start and search “pgadmin”.
- Open pgAdmin and enter the following credentials to connect:
    - General authentication password: postgres
      (or)
    - Localhost PostgreSQL server authentication password: <user-defined password>
- Verify if the databases are created and listed under the databases section, as in the image below:

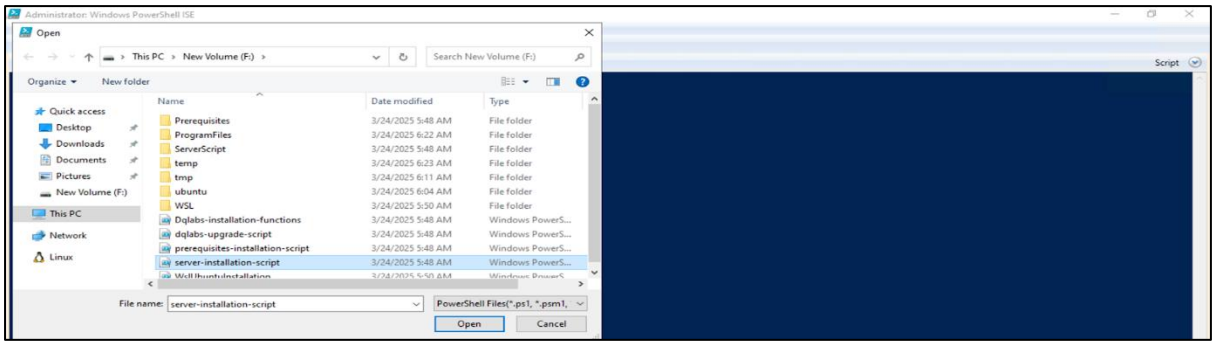
- Close the pgAdmin application.

If the dqlabs/airflow database is not created, stop the installation process and reach out to the support or professional services team for further assistance.

Server Deployment

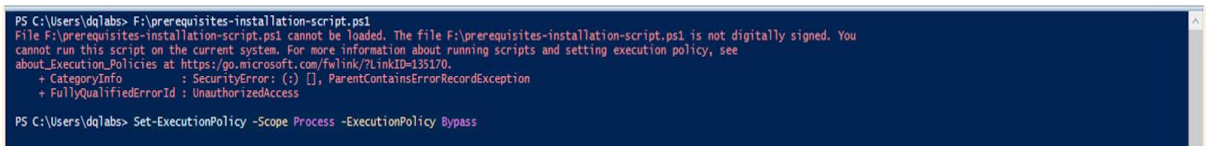
Step 1: Open “PowerShell ISE” as administrator (Press Win key and search for “PowerShell ISE”).

Step 2: In the PowerShell ISE window, click on File > Open to open a PowerShell script file. Locate the Server deployment installation script “server-installation-script”.

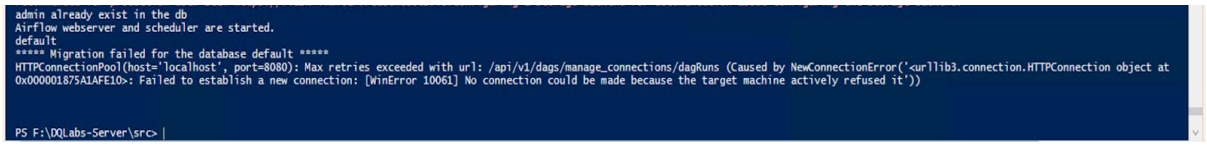


Step 3: Click the “Run Script” button in PowerShell (as shown in the reference image below). If the script throws a digitally not signed error, run the following command and select yes to all.

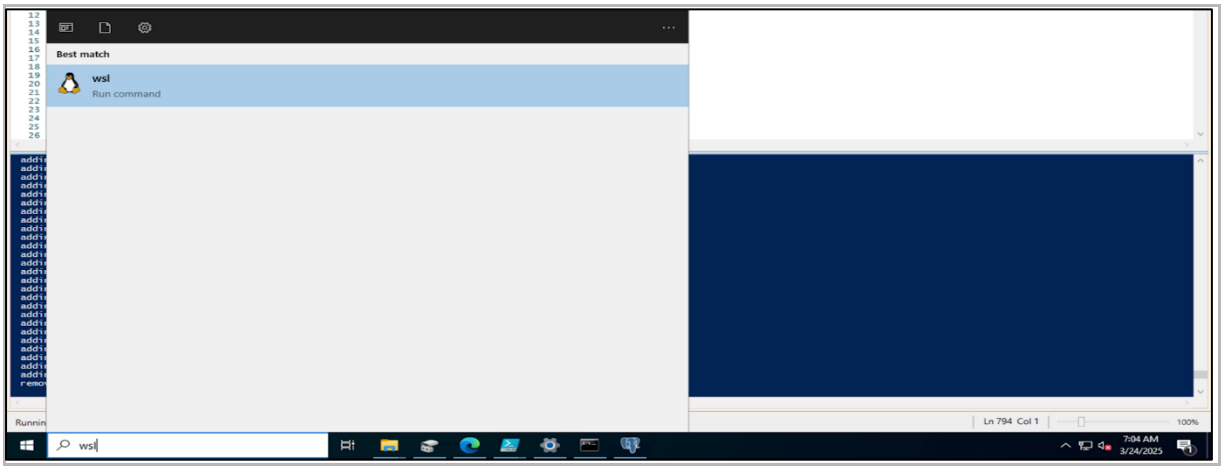
```
Set-ExecutionPolicy -Scope Process -ExecutionPolicy Bypass
```



Step 4: Run the script. Once the script execution is successful, the following status message will appear.



Step 5: Search and open wsl window.



Step 6: Open PowerShell ISE with administrator access and run the following commands (Ensure to replace the <DQ_Drive> Ex: cd “F:\DQLabs-Server\src”).

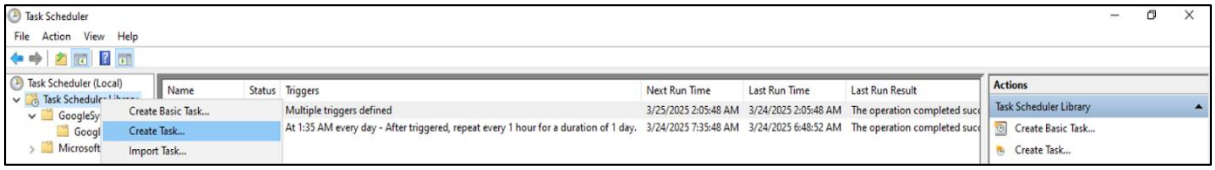
```
cd “<DQ_Drive>:\DQLabs-Server\src"

python manage.py migrate_airflow_connection
```

Post-Installation Procedure

Airflow Auto Start Setup Instructions (Task Scheduler):

Follow the instructions provided below to set up a task in the Windows Scheduler:

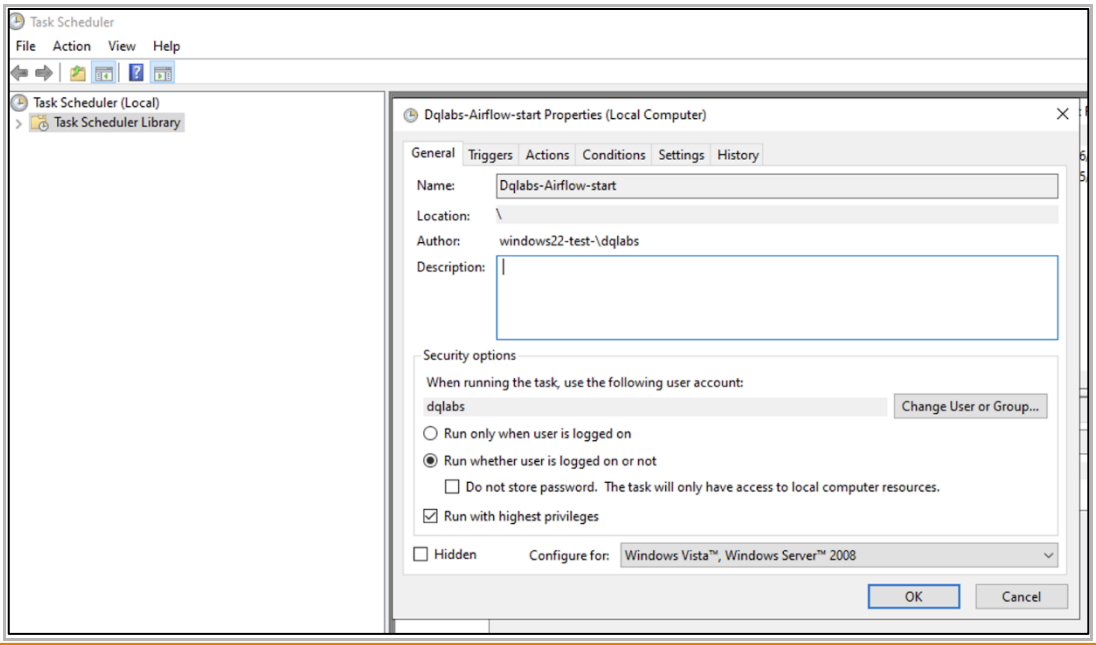
Step 1: Search for “Task Scheduler” in your Windows Server and open the application.



Step 2: Right-click on the Task Scheduler Library on the left pane and click Create Task.

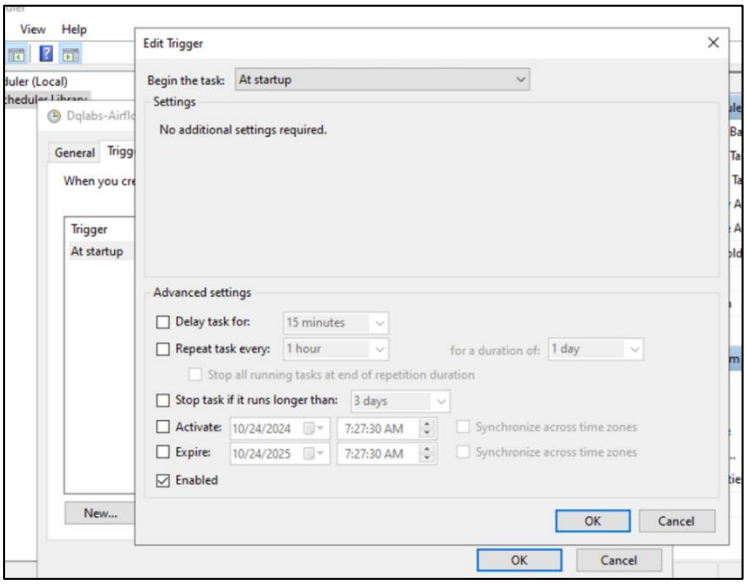
Step 3: On the pop-up window, carry out the following actions:

- Provide the Task a name
- Choose Run whether the user is logged on or not
- Check the box for Run with the highest privileges
- Click OK



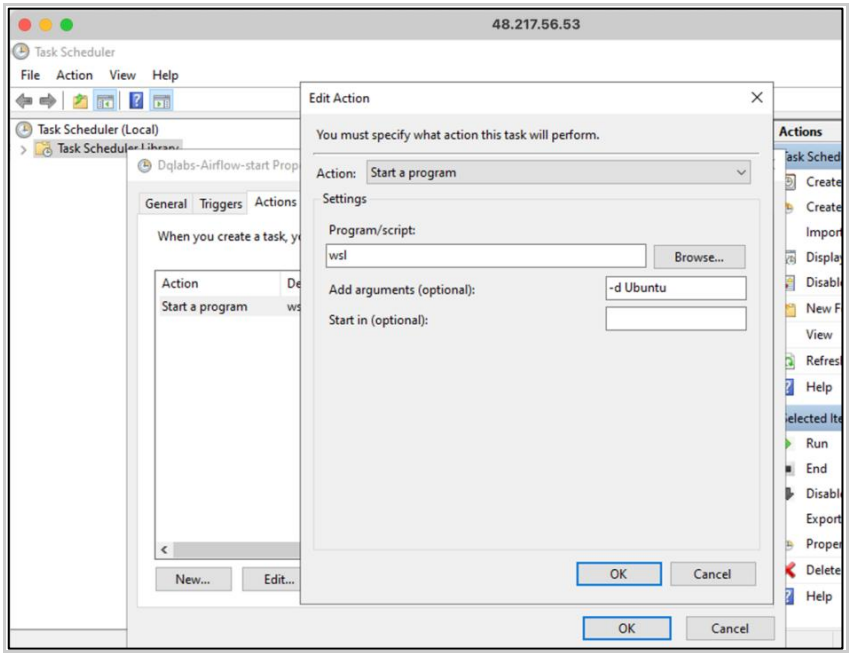
Step 4: Switch to the Triggers tab and perform the following actions:

- Click New
- Begin the task: At startup
- Click **Ok**



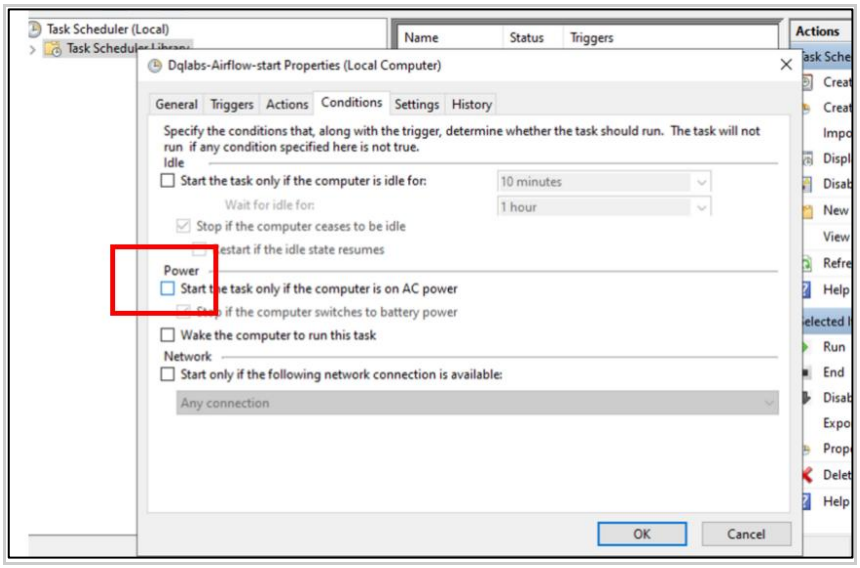
Step 5: Switch to the Actions tab and perform the following actions:

- Click New
- Under Program/script, enter wsl
- Under Add arguments, enter -d Ubuntu
- Click Ok



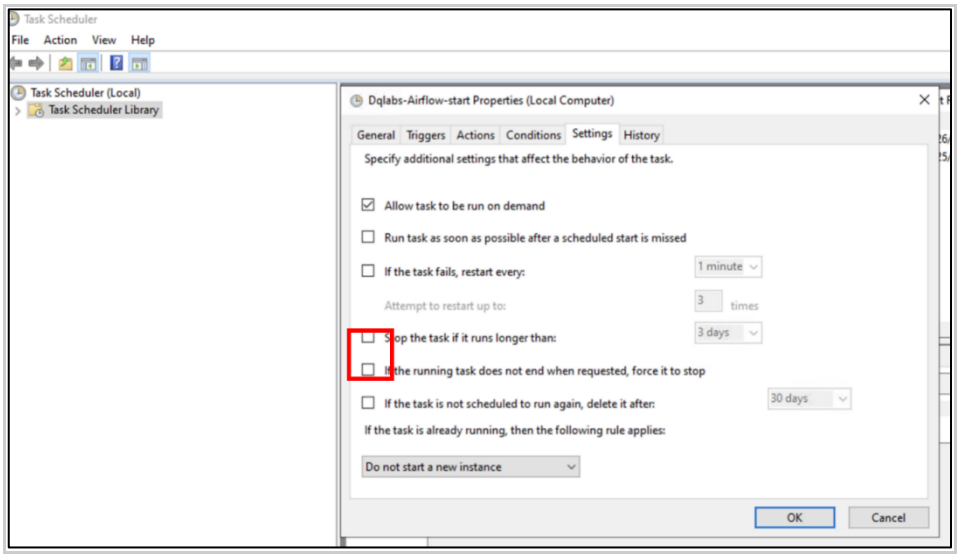
Step 6: Switch to the Conditions tab and perform the following actions:

- Click New
- Uncheck the box - Start the task only if the computer is on AC power
- Click Ok



Step 7: Switch to the Settings tab and perform the following actions:

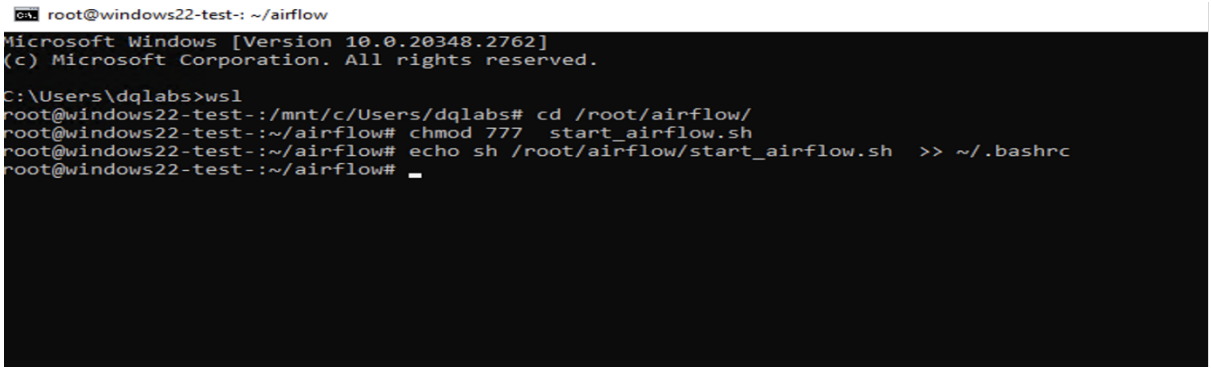
- Uncheck the box - Stop the task if it runs longer than
- Uncheck the box - if the running task does not end when requested, force it to stop
- Click Ok



Step 8: Follow the steps below to autostart Airflow through WSL:

1. Open Command Prompt as an administrator and run the following commands one after the other:

```
wsl


cd /root/airflow/

chmod 777  start_airflow.sh

echo sh /root/airflow/start_airflow.sh  >> ~/.bashrc
```



2. Now, reboot the Windows server and test if airflow starts automatically with a time delay of 2 minutes. To test if airflow has started:

a. After reboot, run localhost:8080 on the server browser to check if the Aiflow UI appears
b. Execute jobs from the Application UI. Running jobs/tasks implies that Airflow has started.

Patch Process

Step 1: Open the File Explorer and navigate to the path <DQ_Drive>:\DQLabs-Server\src\default_data\queries.

Step 2: Take a backup of the synapse.py file and replace it with synapse.py

Step 3: Open the command prompt and run the following commands:

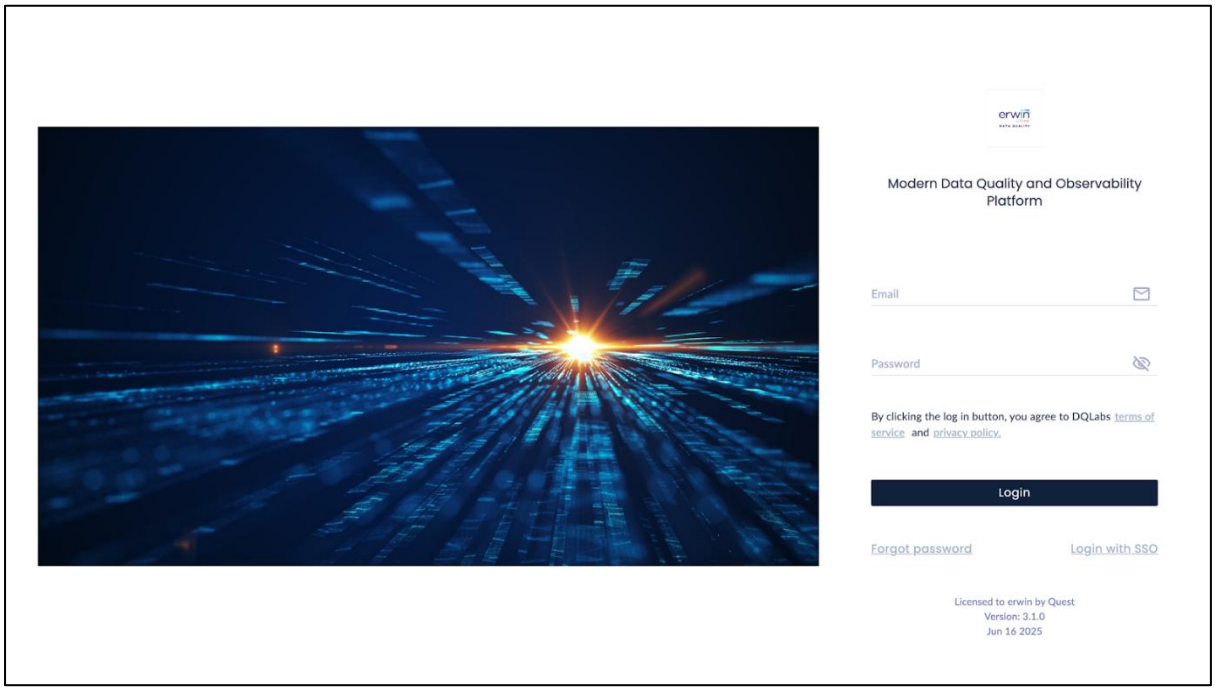
```
# Replace the drive letter with <DQ_Drive>
cd /d <DQ_Drive>:\DQLabs-Server\src


python manage.py init_default_scripts --name queries
```

UI Validation

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.