

erwin Data Intelligence

# erwin Data Quality (DQLabs)

Software Installation Guide - Linux

# Contents

---

Purpose .....	3
System Requirements.....	3
<b>System Requirements</b> .....	<b>3</b>
<b>Prerequisites:</b> .....	<b>4</b>
<b>Software Requirements</b> .....	<b>4</b>
Installation Process .....	4
<b>DQLabs Installation</b> .....	<b>4</b>
<b>Verify Installation</b> .....	<b>8</b>
Completing the Application set up .....	10
<b>Login to the Application</b> .....	<b>10</b>
<b>Activate License</b> .....	<b>10</b>

## Purpose

The document provides the step-by-step process to install DQLabs application on a Linux environment. The audience of this document are personnel with a technical working knowledge on Linux Operating Systems.

## System Requirements

This section provides the minimum requirements that are needed to successfully install the DQLabs application in a Linux environment.

### System Requirements

<b>Operating system</b>	Linux Ubuntu 18.04LTS
<b>Processor</b>	64 Bit
<b>CPU cores</b>	16 or above
<b>RAM</b>	32 GB or above
<b>Disk Space</b>	Minimum 512 GB
<b>Operating system type</b>	Ubuntu 18.04.4 LTS
<b>Kernel</b>	Linux 5.3.0-28-generic

```
ubuntu@ip-172-31-24-121:~$ whoami
ubuntu
ubuntu@ip-172-31-24-121:~$ hostnamectl
  Static hostname: ip-172-31-24-121
        Icon name: computer-vm
        Chassis: vm
        Machine ID: ec217af7622a8996d708f11bbcdb7054
        Boot ID: 9cfaleb8ad5c4adca1c300319be2db42
        Virtualization: kvm
        Operating System: Ubuntu 18.04.6 LTS
        Kernel: Linux 5.4.0-1072-aws
        Architecture: x86_64
ubuntu@ip-172-31-24-121:~$ █
```

## Prerequisites:

- **Dedicated Server** - DQLabs needs a dedicated server for installation, and we recommend installation of Erwin DI and DQLabs on separate servers
- **Internet Access** - The server should have access to internet at the time of installation to successfully complete the installation and activate the application using a valid license key. Internet can be disabled upon successful set up of the DQLabs product.
- **License key** - A new DQ license key is required for activating the product upon installing the product

## Software Requirements

(Auto install by installation script)

PostgreSQL server 12	64 bit
Python3.7 with pip3.7	64 bit
Apache Http Server	2.4.51
Java 1.8 JDK	64 Bit
LIVY Server	N/A
Spark	Ubuntu 18.04.4 LTS
MSSQL, Oracle, PostgreSQL, MySQL ODBC/JDBC driver	64 bit

## Installation Process

### DQLabs Installation

#### Step 1:

Login using putty session

```

ubuntu@ITXLAP112: ~
login as: ubuntu
ubuntu@172.26.208.1's password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 4.4.0-19041-Microsoft x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Mon Nov 15 17:04:10 IST 2021

System load:  0.52   Processes:            11
Usage of /home: unknown  Users logged in:      0
Memory usage: 46%   IPv4 address for eth2: 172.26.208.1
Swap usage:   0%    IPv4 address for wifi0: 192.168.2.184

84 updates can be applied immediately.
23 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Last login: Mon Nov 15 13:41:06 2021 from 172.26.208.1
ubuntu@ITXLAP112:~$

```

**Step 2:**

If sudo command for user doesn't need/require password, you can skip below steps (i.e entire Step2)

- Enter the command **sudo visudo**

```
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/b

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
quest   ALL=(ALL) NOPASSWD:ALL
# Members of the admin group may gain root privileges
%admin  ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL) NOPASSWD:ALL
hadoop  ALL=(ALL:ALL) ALL
# See sudoers(5) for more information on "#include" directives:
jenkins ALL=(ALL)        NOPASSWD: ALL

#includedir /etc/sudoers.d
```

- As shown in the above screen with arrow indicators, please add the required permission in “user privilege specifications” and “members of group sudo to execute any command”
- Example: **quest ALL = (ALL) NOPASSWD:ALL** in user privilege specifications  
Example: **%sudo ALL = (ALL) NOPASSWD:ALL** in members of group sudo to execute any command
- To save the changes you can press “ ctrl+x” and type “Y ”

**Step 3:**

Create a new shell script use vi editor with extension of '.sh'

```
ubuntu@ip-172-31-86-182:~$ vi dqlabs.sh
```

**Go to insert mode by typing 'I'**

Example: vi dqlabs.sh

**Step 4:**

Save and exit the script on VI editor

- Extract file “erwin Quest\_DQlabs\_Linux\_Windows\_Installation” provided by erwin Quest team
- Go to location erwin Quest\_DQlabs\_Linux\_Windows\_Installation\Linux\Env-setup
- Open the .sh file named “**DQLabs\_Linux\_V1.9.sh**” in the notepad (or any other word processor)
- Copy and paste the script from the notepad (or any other word processor) in the VI editor

```

py2spark
systemctl enable livy.service
echo "Install Livy Server Completed" >> ./installation.txt

mkdir -p /usr/local/DQLABS/
./env.sh dqlabs-installation-setup us-east-1 checkout-client.7za /usr/local/DQLABS/checkout-client.7za
./env.sh dqlabs-installation-setup us-east-1 checkout-server.7za /usr/local/DQLABS/checkout-server.7za
7z x /usr/local/DQLABS/checkout-client.7za --o=/usr/local/DQLABS/
7z x /usr/local/DQLABS/checkout-server.7za --o=/usr/local/DQLABS/
rm -rf /usr/local/DQLABS/*.*.7za
cd /usr/local/DQLABS/checkout-server/
pip3.7 install ngram
pip3.7 install --requirements.txt
npm install --g pm2
python3.7 manage.py makemigrations
python3.7 manage.py migrate
python3.7 manage.py create_superuser
echo "from django.contrib.auth.models import User; User.objects.create_superuser('admin@dqlabs.ai', 'admin@dqlabs.ai', 'DQadmin') | python3.7 manage.py shell"
rm2 start /usr/local/DQLABS/checkout-server/start.json
curl https://packages.microsoft.com/keys/microsoft.asc | apt-key add -
curl https://packages.microsoft.com/config/ubuntu/18.04/prod.list > /etc/apt/sources.list.d/mssql-release.list
apt-get update
ACCEPT_EULA=Y apt-get install -y mssql-tools
ACCEPT_EULA=Y apt-get install -y mssql-tools
echo 'export PATH="$PATH:/opt/mssql-tools/bin"' >> ~/.bashrc
source ~/.bashrc

cd /usr/local/DQLABS/checkout-client/
npm install --unsafe
npm run build:dev -- --env '{print 22}' | sed -n 3p
hostip=$(ifconfig | grep -A 1 'eth0' | tail -n 1 | awk '{print $2}')
sed -i "s/192.168.1.22/$hostip/g" /usr/local/DQLABS/checkout-client/src/constants/appConstants.js
npm run build:prod
rm -rf /var/www/html/*
cp -rf build/* /var/www/html/
curl http://localhost:8000/api/setting/initialload

```

- Go to command mode by pressing the escape key
- Save and exit the editor using **':wq!'** and hit on Enter

#### Step 5:

Grant full permission for the script file

- Use the command **'chmod 777 script file name'**.  
For example, **'chmod 777 dqlabs.sh'**

```

ubuntu@ip-172-31-58-190:~$ chmod 777 dqlabs.sh
ubuntu@ip-172-31-58-190:~$

```

#### Step 6:

Run the installation file

- Use the command **'./filename.sh'**.  
Example it is as follows, **'./dqlabs.sh'**

```

ubuntu@ip-172-31-58-190:~$ ./dqlabs.sh
Please Provide Dqlabs version for deployment (1_3_8):

```

#### Step 7:

Select which code version to be installed (for example: **1\_3\_8**)

Please verify the version number to be installed from your erwin Quest team

```

ubuntu@ip-172-31-58-190:~$ ./dqlabs.sh
Please Provide Dqlabs version for deployment (1_3_8):

```

**Step 8:**

Enable or disable Load balancer service

- If you don't have a load balancer, enter '**no**' and then press enter to continue

```
ARE you using loadbalancer (yes/no): no
user input is no
```

- If you are using load balancer, type "**yes**" and hit enter
  - Then enter your **Load balancer DNS name** and hit enter
  - Re-enter the same DNS name to confirm and hit enter
  - If DNS name runs on SSL type, choose "**https**" Or "**http**" and hit enter

```
ubuntu@ip-172-31-58-190:~$ ./dqlabs.sh
ARE you using loadbalancer (yes/no): yes
user input is yes
Please provide your Loadbalancer DNS (foo.subdomain.com) :demo@dqlabs.com
Please re-enter your Loadbalancer DNS (foo.subdomain.com) :demo@dqlabs.com
enter DNS name is : demo@dqlabs.com
Is DNS name runs on ssl (http or https) :
```

**Step 9:**

Depending on your hosting server IP address type, please enter **public** or **private**

**Example**

- If the hosting server is on a Public IP please type **public** and hit enter
- Else if the hosting server is on a Private IP then please type **private** and hit enter

```
user input is no
Need to run Dqlabs in private IP or public IP (private public or localhost) :public
```

```
user input is no
Need to run Dqlabs in private IP or public IP (private public or localhost) :private
```

The spark server installation process will begin now.

**Step 10:**

When prompted, type '**quit()**' [as in below ref. image] and hit enter to complete spark server installation. Now continue with rest of the installation process.

```
SparkSession available as 'spark'.
>>> quit()
COMMAND  PID  USER  FD  TYPE  DEVICE  SIZE/OFF  NODE  NAME
java     12255  ubuntu  178u  IPv6  145340      0t0  TCP  *:8999 (LISTEN)
```

The installation process will continue, and the below output will be displayed

```
Use --update-env to update environment variables
[PM2] Applying action restartProcessId on app [all](ids: [ 0 ])
[PM2] [Dqlabs-server] (0) ✓
```

id	name	namespace	version	mode	pid	uptime	Q	status	cpu	mem	user	watching
0	Dqlabs-server	default	N/A	fork	19538	0s	2	online	0%	7.4mb	ubuntu	disabled

```
ubuntu@ip-172-31-61-251:~$
```

## Verify Installation

### Step 11:

To verify if the installation is successful, please follow the below steps

After installation is completed, check the port for proper functioning.

- Check the **port – 8999** by using the command **“lsof -i:8999”**
- You should be posted with the keyword **‘Listen’** (refer below screenshot).

```
ubuntu@ip-172-31-61-251:~$ lsof -i:8999
COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME
java 19400 ubuntu 178u IPv6 174620 0t0 TCP *:8999 (LISTEN)
ubuntu@ip-172-31-61-251:~$ sudo lsof -i:5432
COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME
postgres 10817 postgres 5u IPv4 92481 0t0 TCP *:postgresql (LISTEN)
postgres 10817 postgres 6u IPv6 92482 0t0 TCP *:postgresql (LISTEN)
python3.7 19541 ubuntu 13u IPv4 177195 0t0 TCP localhost:55668->localhost:postgresql (ESTABLISHED)
python3.7 19541 ubuntu 50u IPv4 176563 0t0 TCP localhost:55670->localhost:postgresql (ESTABLISHED)
python3.7 19541 ubuntu 52u IPv4 169392 0t0 TCP localhost:55672->localhost:postgresql (ESTABLISHED)
python3.7 19541 ubuntu 53u IPv4 174672 0t0 TCP localhost:55676->localhost:postgresql (ESTABLISHED)
postgres 19700 postgres 11u IPv4 176034 0t0 TCP localhost:postgresql->localhost:55668 (ESTABLISHED)
postgres 19713 postgres 11u IPv4 176035 0t0 TCP localhost:postgresql->localhost:55670 (ESTABLISHED)
postgres 19719 postgres 11u IPv4 176037 0t0 TCP localhost:postgresql->localhost:55672 (ESTABLISHED)
postgres 19722 postgres 11u IPv4 176038 0t0 TCP localhost:postgresql->localhost:55676 (ESTABLISHED)
ubuntu@ip-172-31-61-251:~$ sudo lsof -i:80
COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME
apache2 19010 root 4u IPv6 168532 0t0 TCP *:http (LISTEN)
apache2 19012 www-data 4u IPv6 168532 0t0 TCP *:http (LISTEN)
apache2 19013 www-data 4u IPv6 168532 0t0 TCP *:http (LISTEN)
ubuntu@ip-172-31-61-251:~$
```

- Check for **port – 5432** by using the command **“sudo lsof -i:5432”**
- You should be posted with the keywords **‘Listen’** and **‘Established’**. If they are displayed on screen, then the port is enabled correctly
- Check for **port – 80** by using the command **“sudo lsof -i:80”**
- Port Address be posted with the keywords **‘Listen’**. If it is posted, then the port is enabled correctly

### Step 12:

Check whether the **API ‘pm2’** is connected or not, using the command **‘sudo service pm2 status’**

- The status must be **‘Online’** for an established connection

```
[EM2] [WARN] Applications Dqlabs-server not running, starting...
[PM2] App [Dqlabs-server] launched (1 instances)
```

id	name	namespace	version	mode	pid	uptime	Q	status	cpu	mem	user	watching
0	Dqlabs-server	default	N/A	fork	3287	0s	0	online	0%	2.3mb	ubuntu	disabled

```
ubuntu@ip-172-31-26-119:~/usr/local/DQLABS/checkout-server$
```



**Step 14:**

Check for the functioning of the 'LIVY Server'.

- Open your web browser
- please enter the IP in the format, "**IP address: 8999**". In this example, we have used the following address, "192.168.86.1:8999"
- The page should open as in below reference.

Session Id	Application Id	Name	Owner	Proxy User	Session Kind	State	Logs
0					pyspark	starting	<a href="#">session</a>

- Click on '**Session**' hyperlink to navigate to the **Session 0** page

Session 0  
 Session Kind: pyspark  
 State: idle  
 Logs: [session](#)

Id	Execution Code	Execution State	Progress	Execution Status	Output	Started	Completed	Duration
0	<pre>from dqlabs import get_context, DQLabsOptions  options = DQLabsOptions() dqlabs_context = get_context(options)  execution_env = 'livy' dqlabs_context.load_all_dep endencies(execution_env)</pre>	available	100%	ok	<pre>/usr/local/lib/python3.7/dist-packages/fuzzywuzzy/fuzz.py:11: UserWarning: Using slow pure-python SequenceMatcher. Install python- Levenshtein to remove this warning warnings.warn('Using slow pure-python SequenceMatcher. Install python-Levenshtein to remove this warning')</pre>	2021-11-15 16:34:00	2021-11-15 16:34:04	5 s

- The Execution Status must be 'Ok' for a valid connection.

**If all the above steps are done successfully, this completes the installation and verification of the installation Status.**

The next step is to login to the application and activate the license.

# Completing the Application set up

## Login to the Application

- Go to address bar of any supported browser
- Input the IP address/DNS name where the application was installed (for example: (<https://Quest.erwinDQ.io/> or <http://192.168.96>) and hit enter
- The product will be launched on the browser and the login screen will be displayed
- Use the provided Admin Username and Password to login into the application in order to complete the set up.

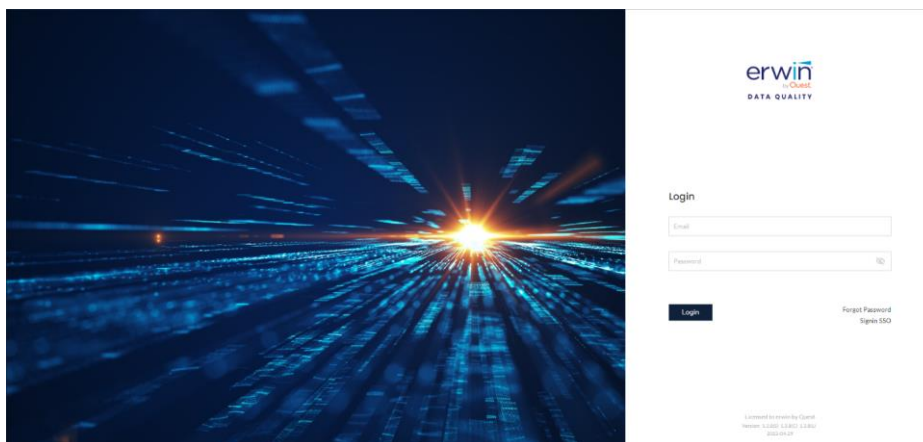


Figure 16 (DQLabs Login Screen)

**Note:** Reach out to your erwin by Quest contact to obtain the Admin login credentials (email/password) to login to the DQ application. Once the Admin logs in, they need to activate the instance using a valid license key (see next step).

## Activate License

- After logging in to the application, click on **settings** icon (left bottom corner) as show in below image

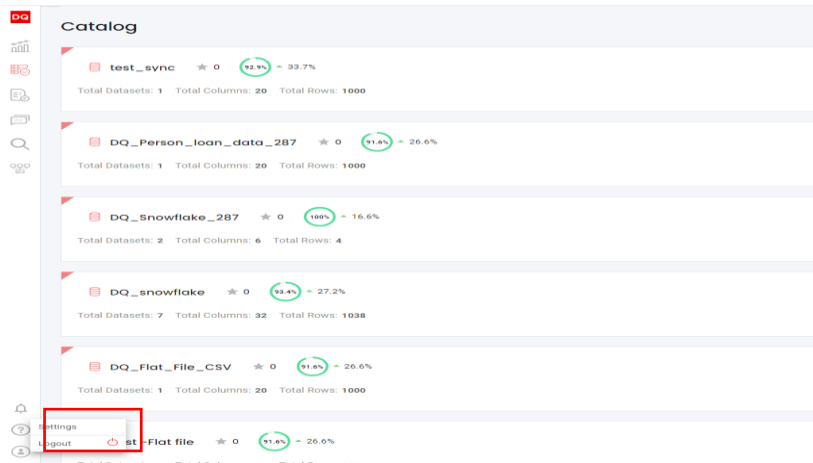


Figure 17.1 (License Activation)

- Click on "System" tab and then click on "License" dropdown
- Enter the license key {shared by the Erwin Quest License team} as show in the image below

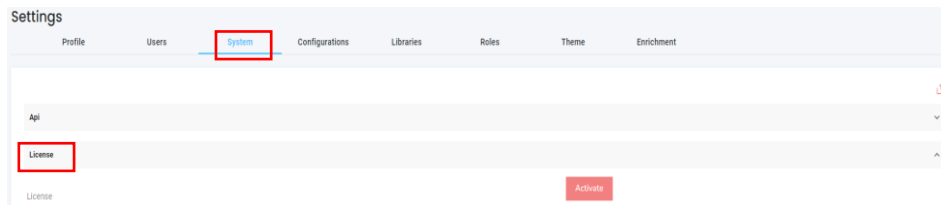


Figure 17.2 (License Activation)

- Click on "Activate" button
- Please verify the License status to be "Active" (as shown in below ref. image), to ensure that the License is successfully activated

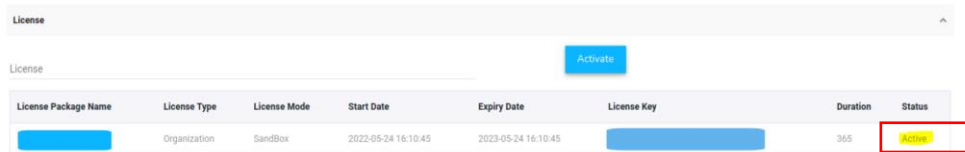


Figure 17.3 (License Activation)

**Important Note\*\*:** If any errors are encountered during the installation process, the install script will exit with an appropriate error message. If you encounter any issues, reach out to the Erwin support or professional services team so we can provide the necessary assistance.

End of document