QAD Adaptive Applications

Enterprise Edition

Installation Guide

Production Execution

70-3489-3.5
Production Execution 3.5.0
QAD Enterprise Edition
December 2021

Production_Execution_IG_v35.pdf/r9m/r9m

QAD Inc.
100 Innovation Place
Santa Barbara, California 93108
Phone (805) 566-6000
http://www.qad.com

Contents

Change Summary

Change Summary

The following table summarizes significant differences between this document and previous versions.

Date/Version	Description	Reference
December 2021/v3.5.0	First release	--

Chapter 1:

Introduction

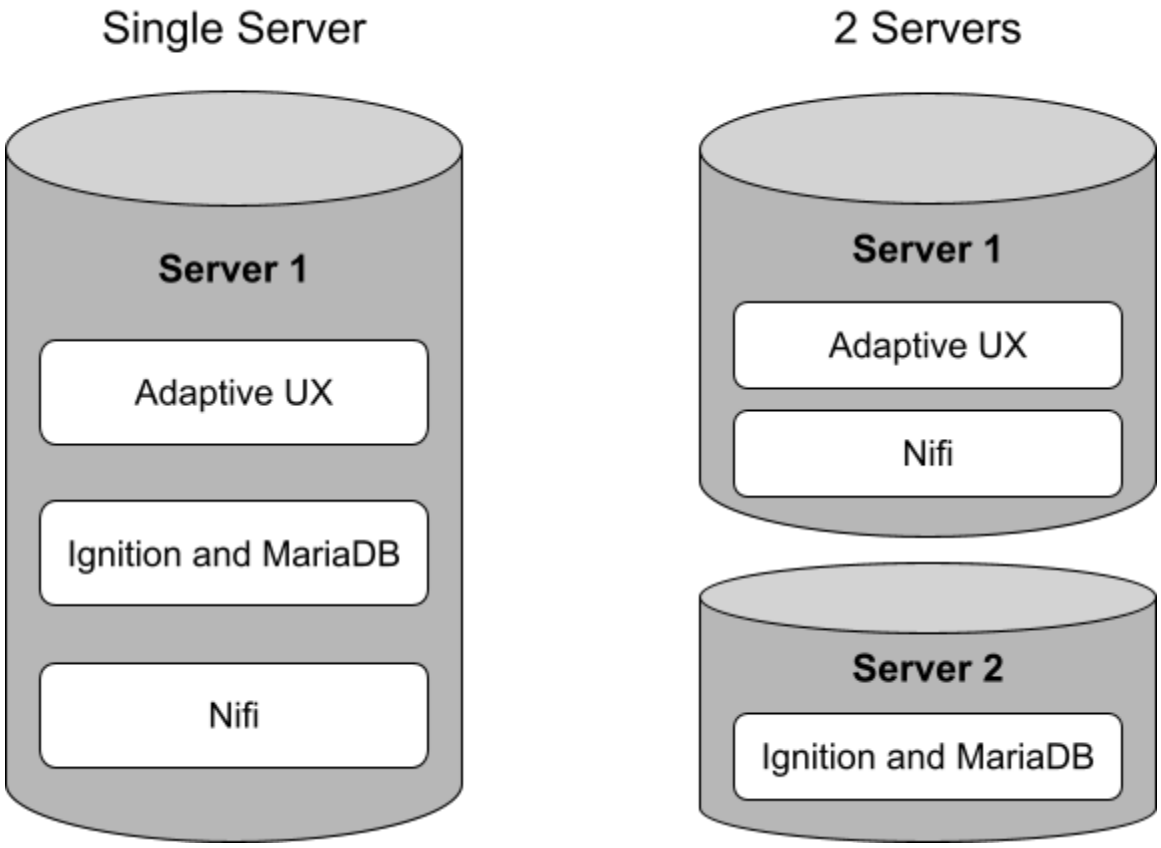
This chapter provides an overview of the Production Execution installation process and the software and environment requirements:

Overview

Installing Production Execution requires the following three sets of artifacts:

1. Adaptive UX-related components. Installed with the `peaux-app` YAB package.
2. Ignition and MariaDB related components. Installed with the manufacturer's installation package.
3. Nifi flow related-components. Installed with the `penifi` YAB package.

Depending on the implementation needs and goals, each of these pieces can be installed on a single server or on two servers. The following diagram shows the installation options.



The `peaux-app` and `penifi` packages are installed using yab. Ignition and MariaDB-related artifacts require manual installation.

Cloud Installation

QAD recommends installing Production Execution in the Cloud; therefore, the instructions in this installation guide are for Cloud installations only. For information about On-Premise installations, please contact QAD Services.

Software and Environment Requirements

The following sections describe the software and environment and other requirements for installing Production Execution on a single server or on multiple servers.

Single-Server Requirements

If you are installing the three sets of artifacts on a single server, the server must have the following software and environments to install software for Production Execution:

- Enterprise Edition 2016EE and above
- QAD Automation Solutions: Data Collection v3.2.2.10 or above
- QAD Automation Solutions: Label Printing Services v3.2.0.0 or above
- QAD Production Orders 3.2.4.0 or above
- Java Runtime Environment (JRE) 1.8 or above
- Progress 11.7 or above
- YAB 1.14.1 or above (yab-ee-app 1.14.1.0 or above)
- qad-enterprise-platform v3.14.10.0 and above
- If installing on a UNIX Server, the zip and unzip packages must be installed
- Production Execution supports Ignition v8.0.15 or above. For the version of Ignition being installed, see the Ignition website for software and hardware requirements: https://inductiveautomation.com/downloads/
- Production Execution supports MariaDB v10.5.x. See the MariaDB website for software and hardware requirements: https://mariadb.com/

Multiple Server Requirements

If you are installing the three sets of artifacts on separate servers, you must have the following software and environment to install software for Production Execution:

Adaptive UX Components and Nifi Flow Components

The server where the Adaptive UX components are installed must have the following software and environment to install software for Production Execution:

- Enterprise Edition 2016EE and above
- QAD Automation Solutions: Data Collection v3.2.2.10 or above
- QAD Automation Solutions: Label Printing Services v3.2.0.0 or above
- QAD Production Orders 3.2.4.0 or above

- Java Runtime Environment (JRE) 1.8 or above
- Progress 11.7 or above
- YAB 1.14.1 or above (yab-ee-app 1.14.1.0 or above)
- qad-enterprise-platform v3.14.10.0 or above

Ignition and MariaDB Components

The server where the Ignition and MariaDB components are installed must have the following software and environment to install software for Production Execution:

- Production Execution supports Ignition v8.0.15 or above. For the version of Ignition being installed, see the Ignition website for software and hardware requirements: https://inductiveautomation.com/downloads/
- Production Execution supports MariaDB v10.5.x. See the MariaDB website for software and hardware requirements: https://mariadb.com/
- PEIgnition package (downloaded from the QAD store during the installation process). This package contains artifacts required for Ignition and MariaDB configuration.

Chapter 2:

Installing Production Execution

This chapter provides detailed instructions on installing Production Execution:

Installation Overview

This chapter describes how to install the three sets of artifacts for Production Execution:

- Ignition
- Maria DB
- Adaptive UX

The installation instructions in this install guide are for Cloud installations. The installation packages for each set of artifacts do not need to be downloaded and are installed from the Cloud catalog.

Note: For information about On-Premise installations, please contact QAD Services.

Installing Production Execution

Pre-Installation Tasks

Back Up Databases

Before installing the three sets of artifacts for Production Execution, back up your databases and supporting environments.

To back up all databases, execute the command:

```
> yab database-backup
```

Alternatively, to back up a specific database, execute the command:

```
> yab database-[INSTANCE]-backup
```

Download PE Ignition Package

Download the PE Ignition package from the QAD Store onto the server where Ignition and Maria DB will be installed. The PE Ignition package contains artifacts for MariaDB and Ignition configuration.

Installing Ignition and MariaDB Artifacts

Stop Firewall

Before installing Ignition and the Maria DB artifacts, stop the firewall with the following command. Make sure ports are open to red-hat linux repositories.

```
service firewalld stop
```

Installing MariaDB

1. Create the following `MariaDB.repo` file if it does not exist. If this file exists, skip to the next step to update the file.

   ```
   /etc/yum.repos.d/MariaDB.repo
   ```

2. Specify the version of MariaDB to be installed by adding the following lines to the `MariaDB.repo` file:

   ```
   [mariadb]

   name = MariaDB-10.5

   baseurl=https://yum.mariadb.org/10.5/centos7-amd64

   # alternative:
   baseurl=http://archive.mariadb.org/mariadb-10.5/yum/centos7-amd64

   gpgkey=https://yum.mariadb.org/RPM-GPG-KEY-MariaDB

   gpgcheck=1
   ```

3. Create a directory to store MariaDB data and configuration files; for example, /dr01/mariadb/mysql.

   ```
   sudo mkdir -p /dr01/mariadb/mysql
   ```

4. Add user mysql if it does not exist.

   ```
   sudo useradd mysql
   ```

5. Update ownership of the folder /dr01/mariadb to mysql user.

   ```
   sudo chown -R mysql: /dr01/mariadb
   ```

6. Run the following command to install MariaDB Server and MariaDB Client:

   ```
   sudo yum install MariaDB-server MariaDB-client
   ```

7. The following installation information is displayed. When prompted, enter “y” to confirm.

   ```
   Install 3 Packages (+8 Dependent packages)

   Total download size: 53 M

   Is this ok [y/d/N]:  y

   Retrieving key from
   https://yum.mariadb.org/RPM-GPG-KEY-MariaDB
   ```

```
Importing GPG key 0x1BB943DB:

Userid: "MariaDB Package Signing Key
<package-signing-key@mariadb.org>"

Fingerprint: 1993 69e5 404b d5fc 7d2f e43b cbcb 082a 1bb9
43db

From: https://yum.mariadb.org/RPM-GPG-KEY-MariaDB

Is this ok [y/N]: y
```

8. Remove all content from `/etc/my.cnf` and update it with the following lines. Root or Sudo permission might be required for updating the file.

```
#

# This group is read both by the client and the server

# use it for options that affect everything

#

[client-server]

#

# include *.cnf from the config directory

#

!includedir /etc/my.cnf.d
```

9. Update `/etc/my.cnf.d/server.cnf` to specify the datadir and socket path. The following four lines under the [mysqld] tag need to be entered. The other tags should not be changed. Root or Sudo permission might be required for updating the file.

```
[mysqld]

datadir=/dr01/mariadb/mysql

socket=/dr01/mariadb/mysql.sock

character_set_server=utf8

collation_server=utf8_general_ci
```

10. Update `/etc/my.cnf.d/mysql-clients.cnf` to specify the path to the `mysql.sock` file. The following line under the [mysql] tag needs to be updated. The other tags should not be changed. Root or Sudo permission might be required for updating the file.

```
[mysql]

socket=/dr01/mariadb/mysql.sock
```

11. Install the system table by entering:

```
sudo mysql_install_db
```

12. Change the ownership of the file under `/dr01/mariadb/mysql` by entering:

```
sudo chown -R mysql: /dr01/mariadb/mysql
```

13. Enable MariaDB by entering:

```
sudo systemctl enable mariadb
```

14. Start MariaDB by entering:

```
sudo systemctl start mariadb
```

15. Secure the MariaDB installation by entering the following command:

```
sudo mariadb-secure-installation -S
/dr01/mariadb/mysql.sock
```

    a. When prompted for root user password, press Enter/leave blank.

    ```
    Enter current password for root (enter for none):
    ```

    b. Enter the following when prompted:

    ```
    Switch to unix_socket authentication [Y/n] n

    Change the root password? [Y/n] Y

    Enter new password that you want to set for root user

    New password: <enter new root user password>

    Re-enter new password: <re-enter new root user
    password>

    Remove anonymous users? [Y/n] Y

    Disallow root login remotely? [Y/n] n

    Remove test database and access to it? [Y/n] Y

    Reload privilege tables now? [Y/n] Y
    ```

16. Create “database qad” and then load schema and data:
    a. Create Database qad by entering:

```
mysql -S /dr01/mariadb/mysql.sock -u root -p
```

b. Enter Password: <enter root password>
c. Enter the following:

```
MariaDB [(none)]> create database qad;

MariaDB [(none)]> exit;
```

d. Switch to the folder where the `PEdbanddata.sql` file is available or edit the directory location in the command itself:

```
mysql -S /dr01/mariadb/mysql.sock -u root -p qad <
<directoryLocation>/PEdbanddata.sql
```

Note: The PE Ignition package that was downloaded contains the `PEdbanddata.sql` file.

e. Enter Password: <enter root password>

17. Provide remote access for root user by entering the following:

```
sudo mysql -S /dr01/mariadb/mysql.sock -u root -p

Password:  <enter root password>

MariaDB [(none)]> CREATE USER 'root'@'%' IDENTIFIED BY '
<enter root password>';

MariaDB [(none)]> GRANT ALL PRIVILEGES ON *.* TO
'root'@'%' WITH GRANT OPTION;

MariaDB [(none)]> GRANT ALL PRIVILEGES ON *.* TO
'root'@'localhost' WITH GRANT OPTION;

MariaDB [(none)]> FLUSH PRIVILEGES;

MariaDB [(none)]> exit;
```

18. The installation of MariaDB is complete. Proceed to the next section and install Ignition.

Installing Ignition

1. Download Ignition installer package v8.0.15:

   https://inductiveautomation.com/downloads/archive/8.0.15

2. Select the Ignition package based on the OS.

   Note: If installing on a Linux server, make sure the downloaded file has execute permissions.

3. Execute the install file. You can log in as root or use the sudo command to install:

   ```
   sudo ./ignition-8.0.15-linux-x64-installer.run
   ```

   a. Specify the username for installing Ignition.

      ```
      Username: <username to install ignition>
      ```

   b. Enter the Ignition Package installation location; for example, [/usr/local/bin/ignition]: /dr01/ignition:

      ```
      [/usr/local/bin/ignition]: <directory path of
      ignition install>
      ```

   c. Specify if the installation type is typical or custom. It is recommended that you select the "1 (Typical)" option:
      - [1] Typical - Includes Ignition with SQL Bridge, Perspective, Vision, OPC-UA, and driver modules for Allen-Bradley, Siemens, and MODBUS devices.
      - [2] Custom - Install additional modules and adjust the default modules to install.

   d. Setup is now ready to begin installing Ignition on your computer. Enter "Y" to begin installation:

      ```
      Ready to Install

      Do you want to continue? [Y/n]: Y
      ```

   e. A confirmation message is displayed when the installation is complete:

      ```
      Ignition Successfully Installed
      ```

   f. Enter "Y" to start Ignition:

      ```
      Start Ignition Now [Y/n]: Y
      ```

   g. Enter "Y" to install service:

      ```
      Install Service [Y/n]: Y
      ```

4. Copy the following three Kafka scripts, which are available in the package, into `/usr/bin` or any other directory that is in the system PATH:
   - `rdkafka_simple_producer.sh`
   - `rdkafka_simple_producer_ssl`
   - `rdkafka_simple_producer`
5. Set execute permissions to the scripts by entering:

```
sudo chmod +x
<scripts-location>/rdkafka_simple_producer.sh
rdkafka_simple_producer_ssl rdkafka_simple_producer
```

6. The installation of Ignition is complete. Restart the firewall and then proceed to the next section to install the Adaptive UX components and Nifi.

Restart Firewall

After installing MariaDB and Ignition, restart the firewall. Make sure ports are open to red-hat linux repositories.

```
service firewalld start
```

Installing Adaptive UX Components and Nifi Flow Components

The Adaptive UX and Nifi Flow components are installed using YAB.

YAB is a configuration management tool you can use to start, stop, reconfigure, and maintain the consistency of your Enterprise Edition environments. Your environment is defined in terms of discrete units, or packages, which you can replace and add as needed.

See *QAD Enterprise Edition Installation Guide* for more information about using YAB.

The peaux-app-.x.x.x.x package contains the Adaptive UX components. The penifi-x.x.x.x package contains the Nifi Flow Components.

Note: If you are only updating to a new version of the Nifi Flow components, see Updating Nifi Flow Components.

Follow these steps to install the Adaptive UX and the Nifi Flow components:

1. Add the following properties to the `configuration.properties` file and customize its value as per your configuration needs.

   ```
   pe.mariadb.name=<Name of the MariaDB database>

   pe.mariadb.host=<Hostname where MariaDB database is
   installed>

   pe.mariadb.port=<Port configured for MariaDB>

   pe.mariadb.user=<MariaDB login credentials for user with
   write access to Production Execution tables>

   pe.mariadb.password=<MariaDB login credentials for user
   with write access to Production Execution tables>
   ```

```
pe.integration.user=<AUX login credentials for user that
is used by Nifi to login to AUX and invoke APIs>

pe.integration.password=<AUX login credentials for user
that is used by Nifi to login to AUX and invoke APIs>

pe.qms.images.targetdirectory=<Target Directory for
storing QMS artifacts>

pe.qms.images.sourcedirectory=<Source Directory for
getting QMS artifacts>
```

2. To install both packages via the Cloud catalog, enter the following yab command:

   ```
   yab install peaux-app-x.x.x.x penifi-x.x.x.x
   ```

Configure Ignition Gateway

This section describes the steps required to configure the Ignition Gateway after it has been installed. During this configuration procedure you will:

- Create the Admin user and set the password.
- Update the port numbers used to access Ignition Gateway.
- Import project files, which contain PE functionality.
- Create a database connection for MariaDB.
- Create a new User Source, which connects the user roles provided with MariaDB with the Ignition project.
- Create a new Realtime Tag Provider, which provides additional functionality for PE.
- Link project files with User Source and Tags.

Follow these steps to configure the Ignition Gateway:

1. Check if the Ignition Gateway is started by entering:

   ```
   sudo <Ignition installation directory>/ignition.sh status
   ```

   Note: If the Ignition Gateway is stopped, enter the following command to start it:

   ```
   sudo <Ignition installation directory>/ignition.sh start
   ```

2. Ignition gateway is configured from a web browser. You will need the ignition gateway port, which is located in this file:

   ```
   <Ignition installation directory>/data/gateway.xml.
   ```

3. For new installs, the default port is 8088:

   ```
   <entry key="gateway.port">8088</entry>
   ```

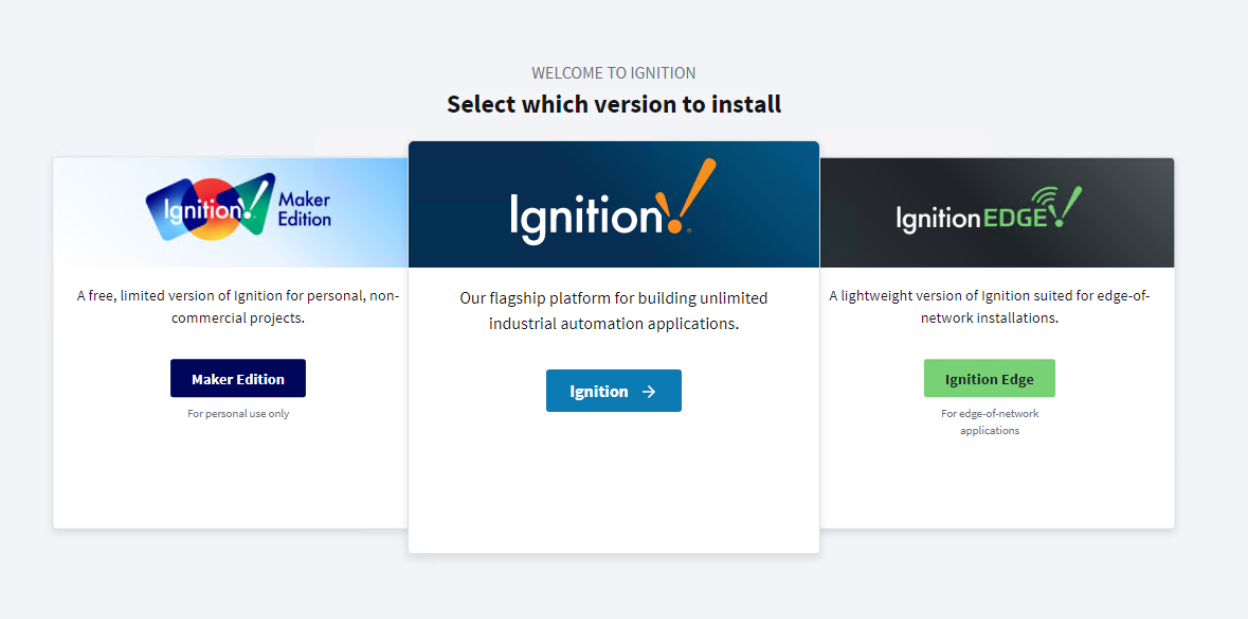
4. Access the Ignition Gateway from a web browser by entering:

   `http://<hostname>:<port>`

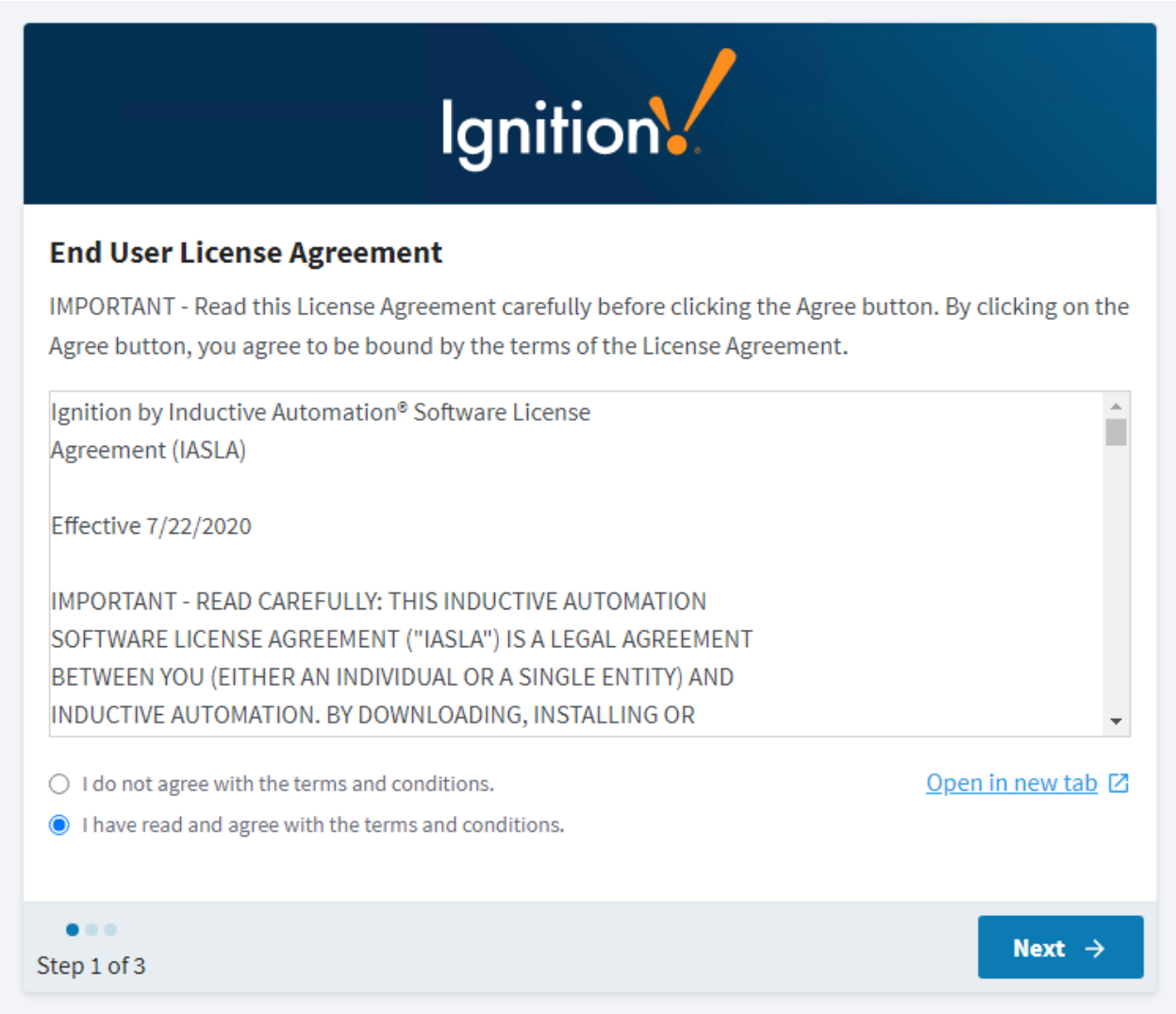
   Note: If it not accessible, the port needs to be updated by entering:

   `sudo <Ignition installation directory>/gwcmd.sh -k <new port>`

5. On the welcome page, select Ignition.

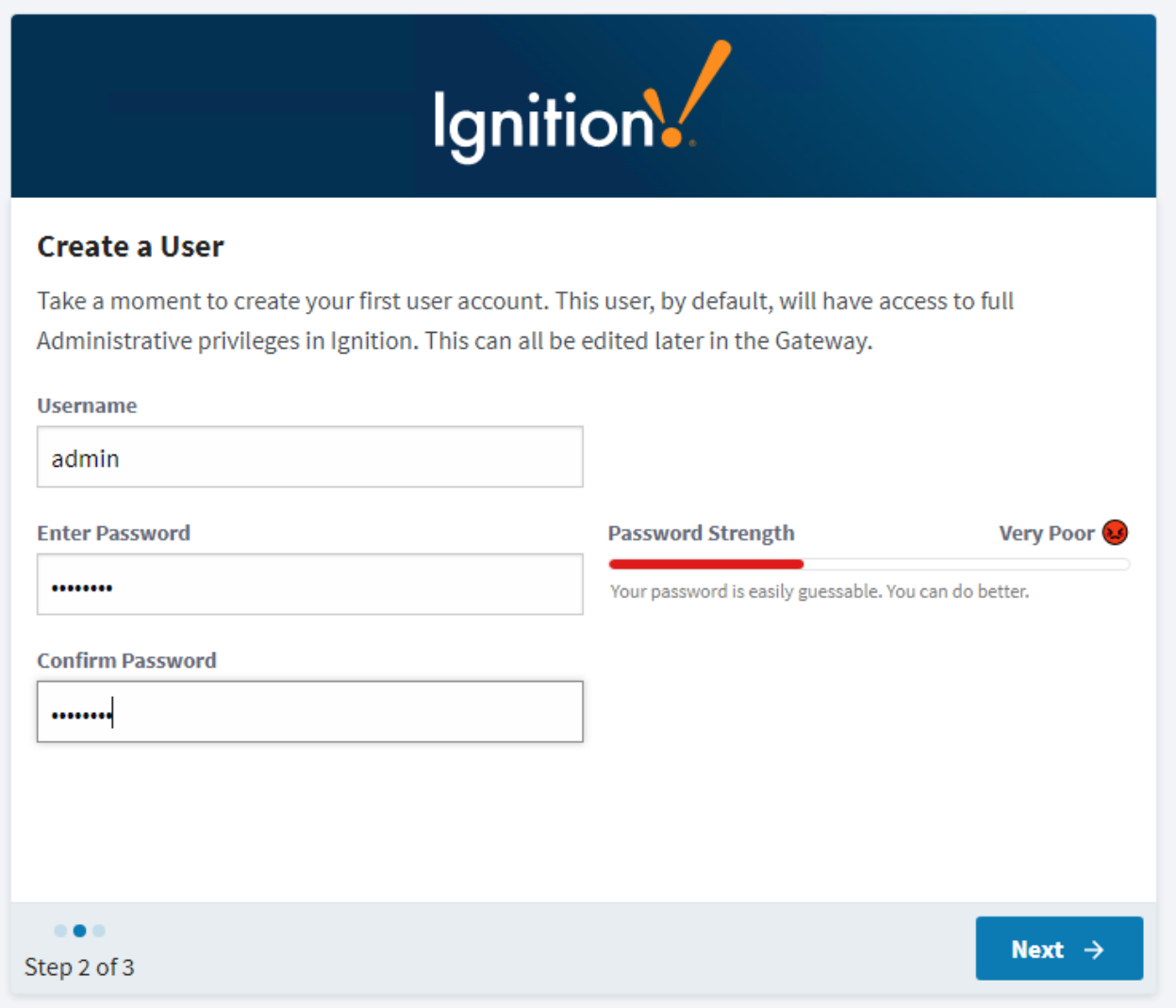


6. Navigate through the terms and conditions page.

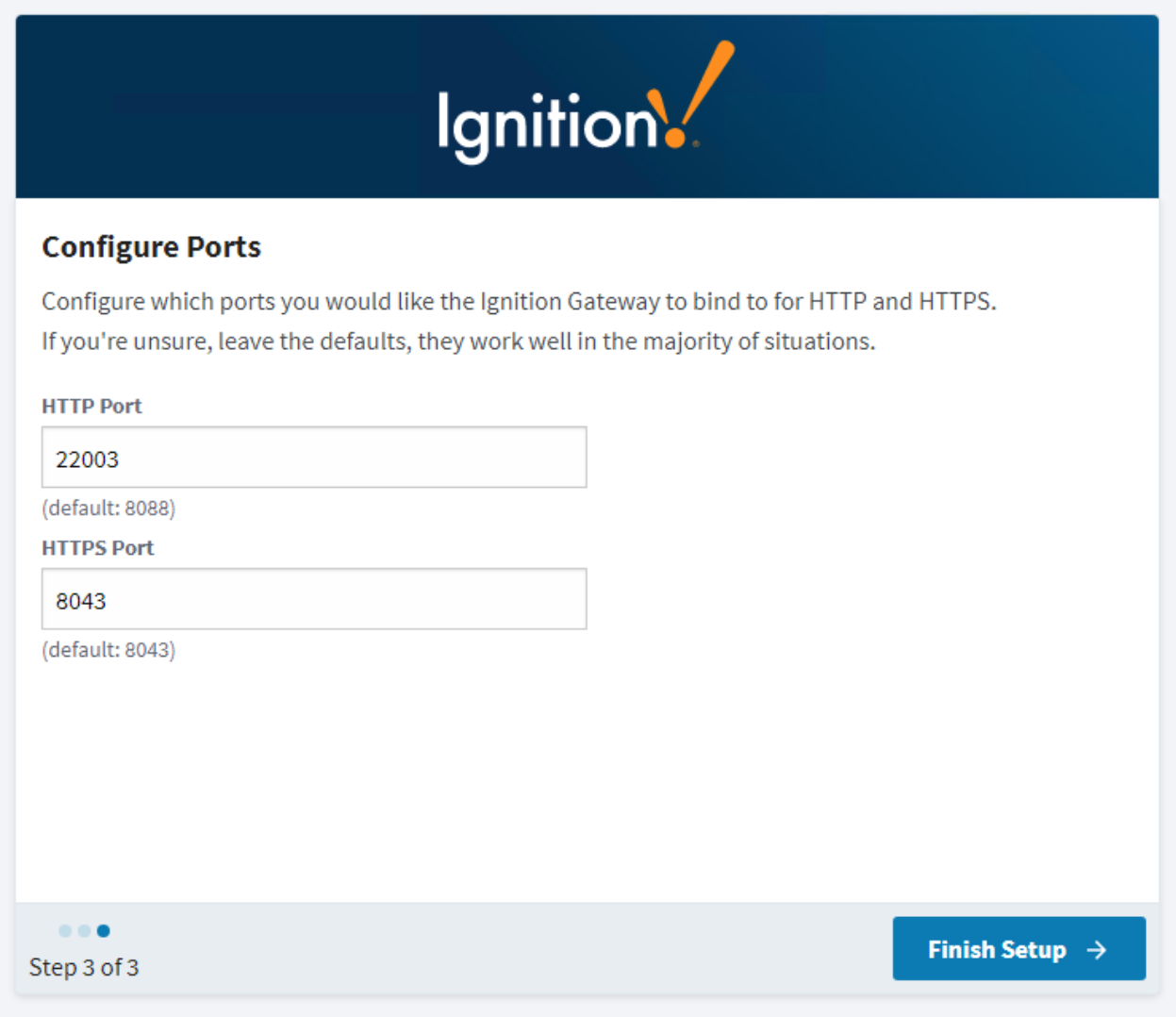


7. Create a user in Ignition for the Admin role and then set the password.

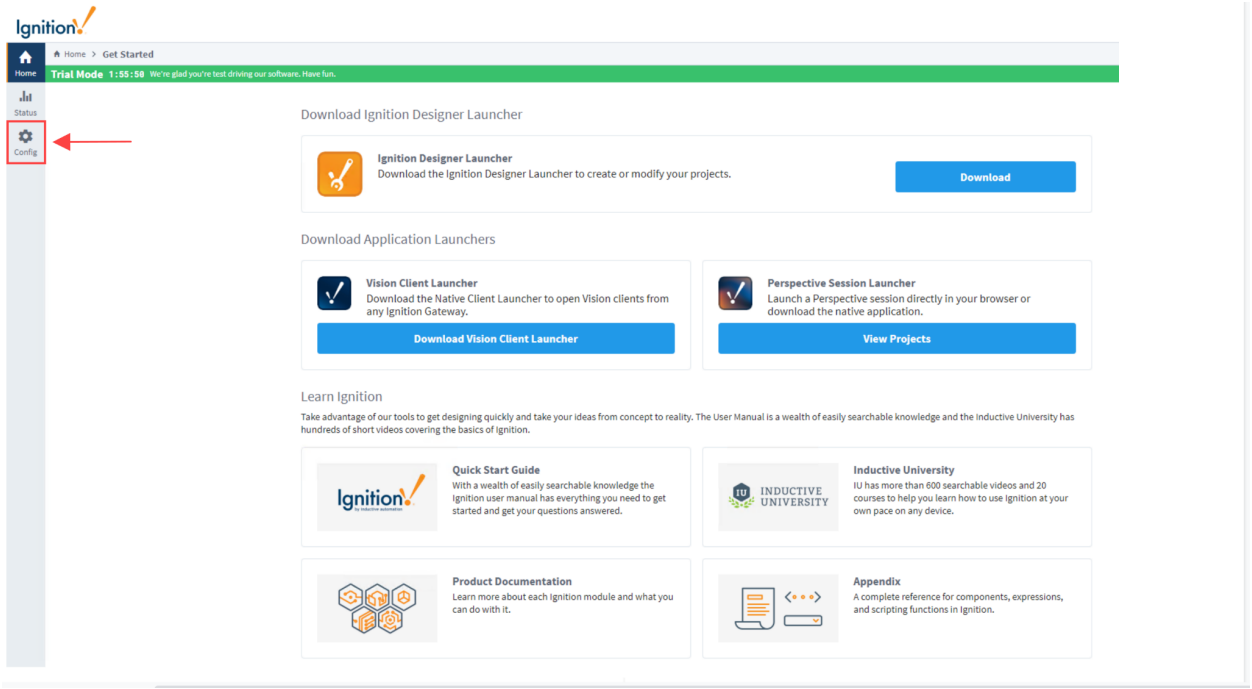
   Note: Make a record of the username and password because these credentials are used each time you access the Ignition Gateway.



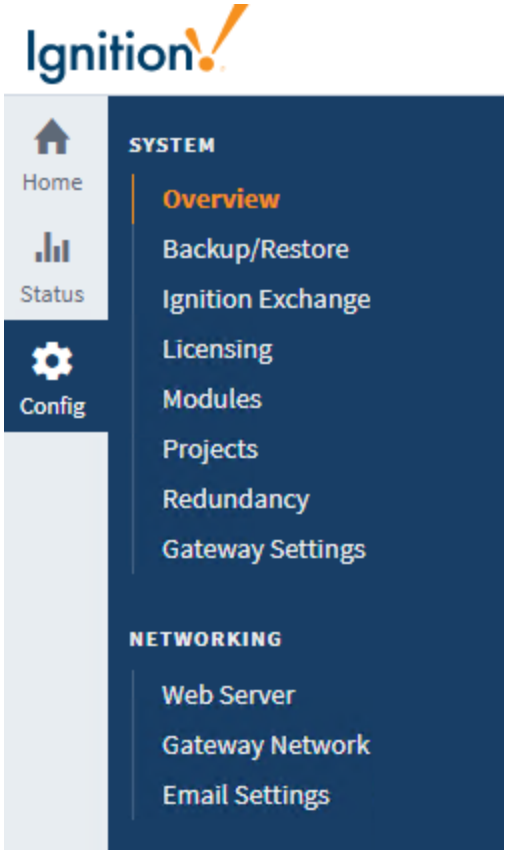
8. Update the port numbers for HTTP Port and HTTPS Port access. The port numbers that are entered are used to access the Ignition Gateway using the URL.

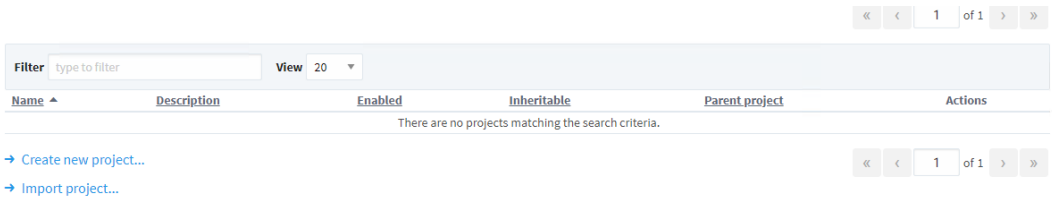


9. Next, import the Project .zip file and connect MariaDB.
10. Select the Config menu icon on the Ignition Gateway Home Page.

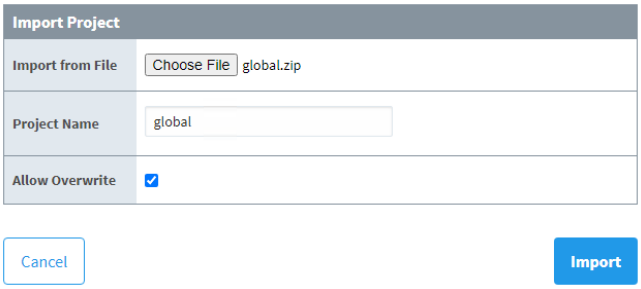


11. Select Projects from the Config menu. The Projects option allows users to import projects into the environment.

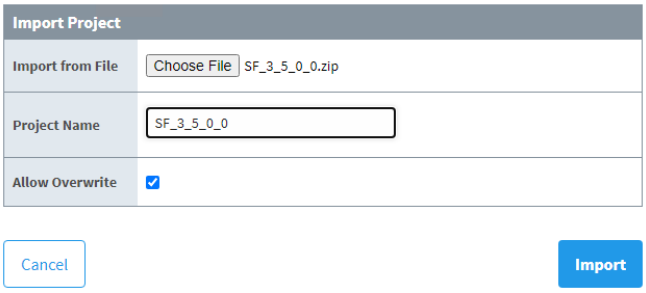




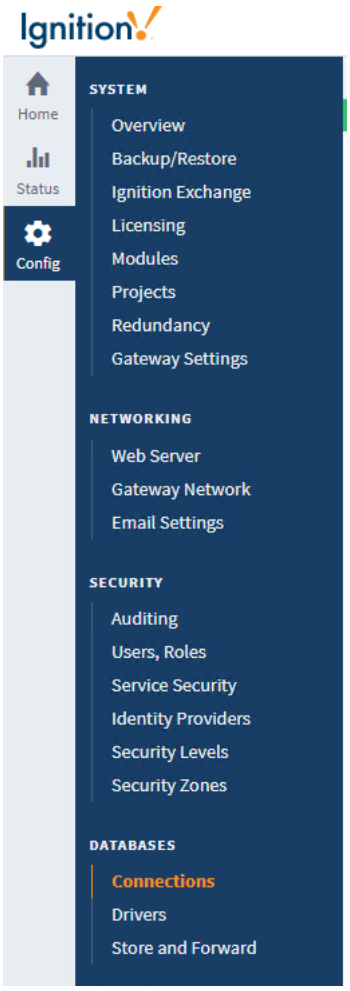
12. Import the `global.zip` Project file by selecting Choose File and then locating the file. The Project file is provided in the PE Ignition installation package.
13. In the Project Name field, enter “global” and then select the Allow Overwrite check box.
14. Select Import.



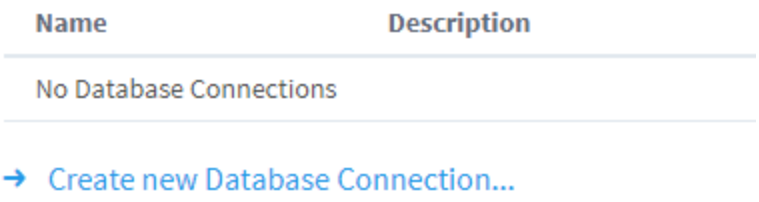
15. Import the `SF_3_5_0_0.zip` Project file, which is also provided in the PE Ignition installation package.
16. In the Project Name field, enter “SF_3_5_0_0” and select the Allow Overwrite check box.
17. Select Import.

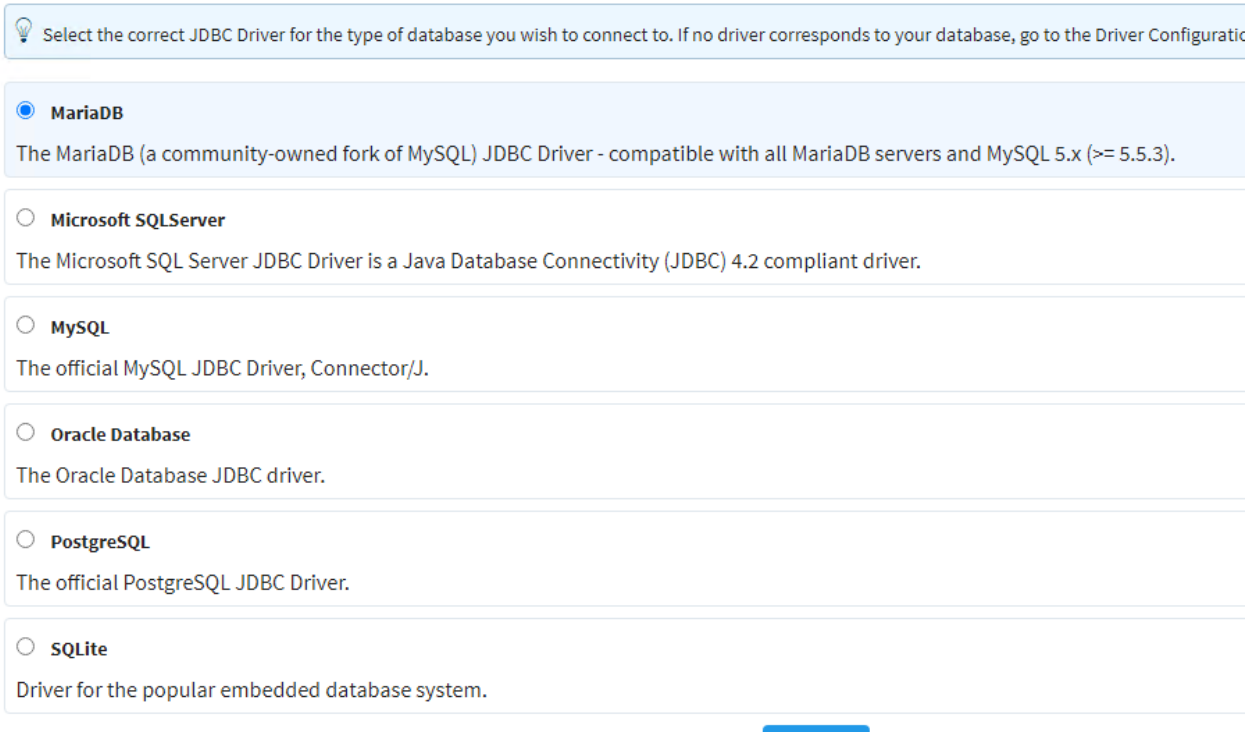


18. Create a new database connection for MariaDB. From the Config menu, select Databases > Connections.



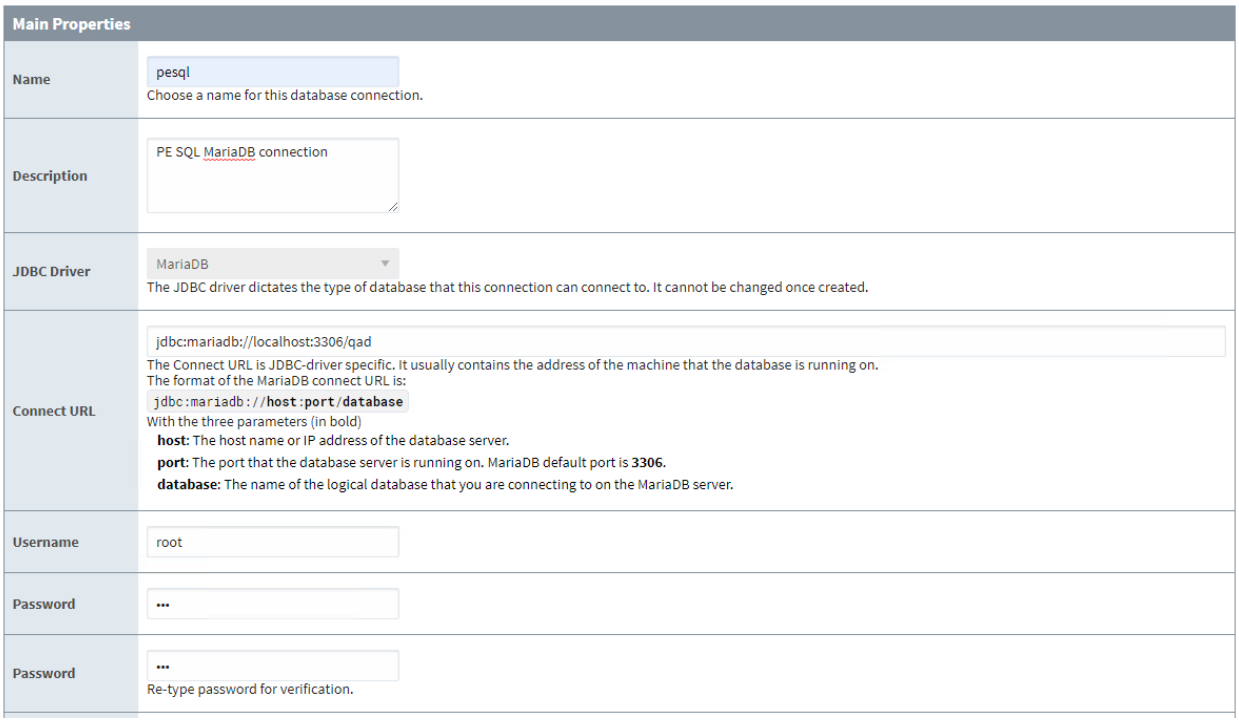
19. Select Create new Database Connection and then select the MariaDB database connection.



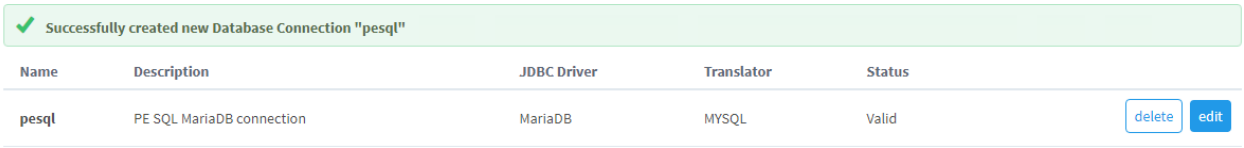


20. Enter the Connection details for MariaDB:

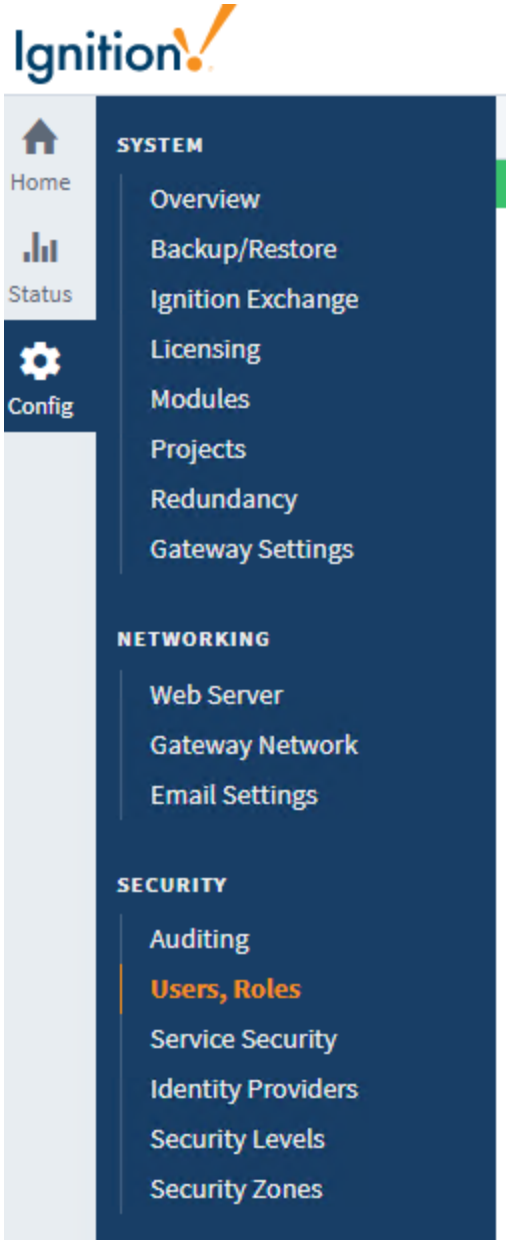
Field	Setting
Name	pesql
Description	PE SQL MariaDB Connection
JDBC Driver	MariaDB
Connect URL	Make sure the port and database name are specified correctly.
Username	<username for mariaDB database>
Password	<mariaDB password for username>



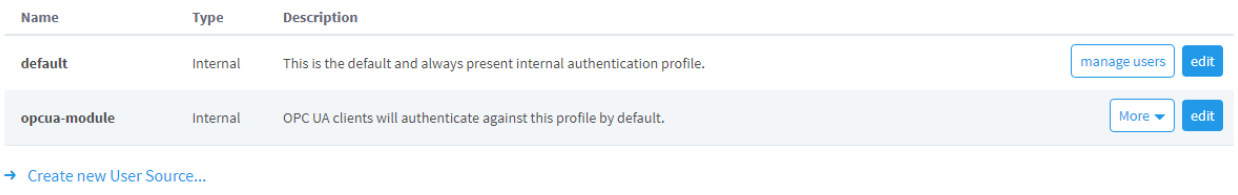
21. After the connection details are saved, the connection Status should display as Valid.



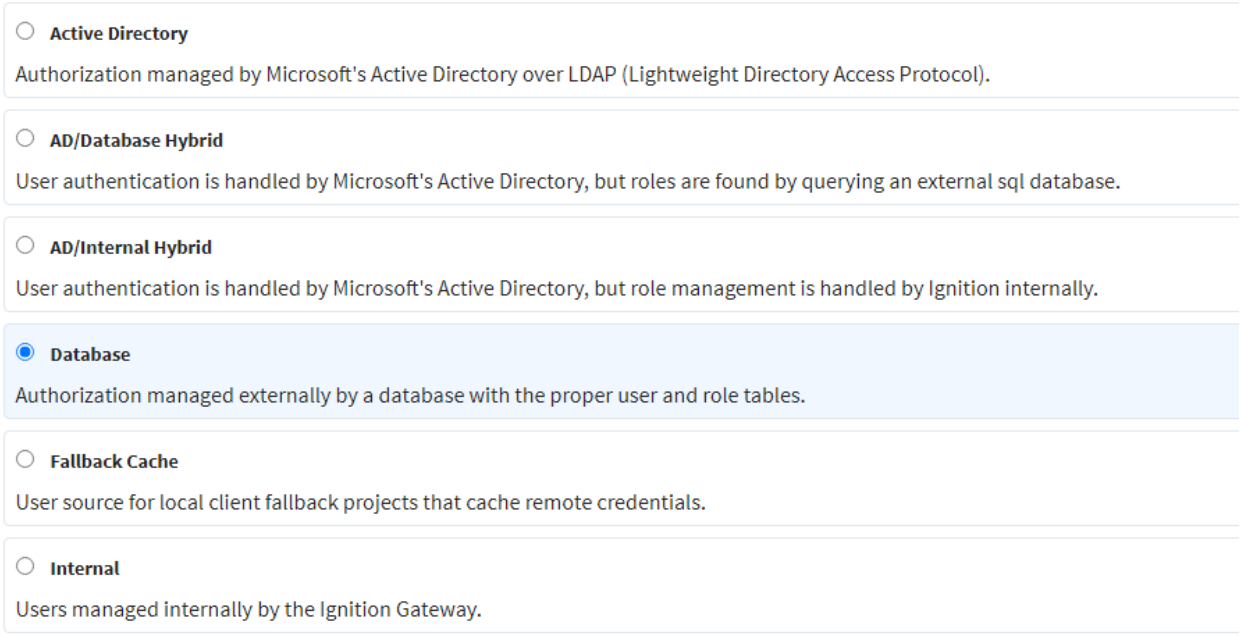
22. Next, create a New User Source by selecting Security > Users, Roles in the Config menu.



23. Select Create new User Source.



24. Select Database as the new User Source.



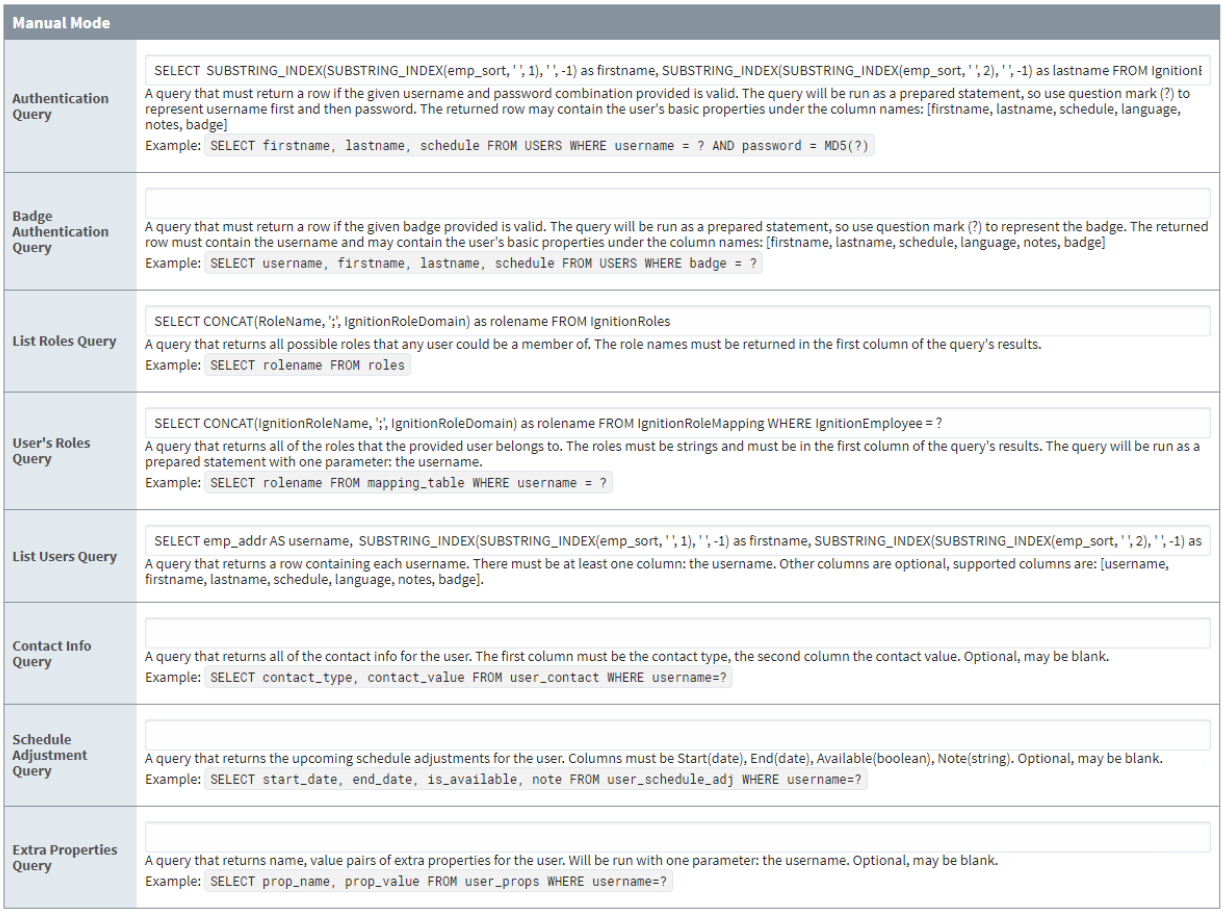
25. Enter Details for the new User Source:

Field	Setting
Authentication Query	SELECT SUBSTRING_INDEX(SUBSTRING_INDEX(emp_sort, ' ', 1), ' ', -1) as firstname, SUBSTRING_INDEX(SUBSTRING_INDEX(emp_sort, ' ', 2), ' ', -1) as lastname FROM IgnitionEmployee WHERE emp_addr = ? AND IgnitionPassword = MD5(?)
List Role Query	SELECT CONCAT(RoleName, ';', IgnitionRoleDomain) as rolename FROM IgnitionRoles
User's Roles Query	SELECT CONCAT(IgnitionRoleName, ';', IgnitionRoleDomain) as rolename FROM IgnitionRoleMapping WHERE IgnitionEmployee = ?
List Users Query	SELECT emp_addr AS username, SUBSTRING_INDEX(SUBSTRING_INDEX(emp_sort, ' ', 1), ' ', -1) as firstname, SUBSTRING_INDEX(SUBSTRING_INDEX(emp_sort, ' ', 2), ' ', -1) as lastname FROM IgnitionEmployee

Main	
Name	dev
Description	
Schedule Restricted	☐ Users are only able to log in when their assigned schedule is active. (default: false)
Failover Source	- none - If this source is unreachable for authentication, this failover source will be used instead.
Failover Mode	Hard The failover mode to use if a failover source is set. **Hard:** Failover only if this source is un-reachable. **Soft:** Try the failover source when a user fails to authenticate with this source. (default: HARD)
Cache Validation Timeout	60000 The amount of time in milliseconds between cache updates of the user source. Values less than zero disables the cache. (default: 60,000)
Lockout Enabled	☐ Lock out a user's account after more than the maximum allowed number of failed authentication attempts occur within the lockout window (default: true)
Lockout Attempts	5 Maximum number of failed authentication attempts allowed within the lockout window before locking the user out. Values less than one disables lockout. (default: 5)
Lockout Window	15 The duration of the lockout window in minutes. Values less than one disables lockout. (default: 15)

Main Properties	
Database	pesql Choose the database connection this user source will use.
Mode	Manual In Automatic mode, tables will be created for you and all interaction with the database is handled automatically. In Manual mode, you write queries by hand against tables that you've created. Management of users (adding, removing etc) is not supported in manual mode. (default: Automatic)

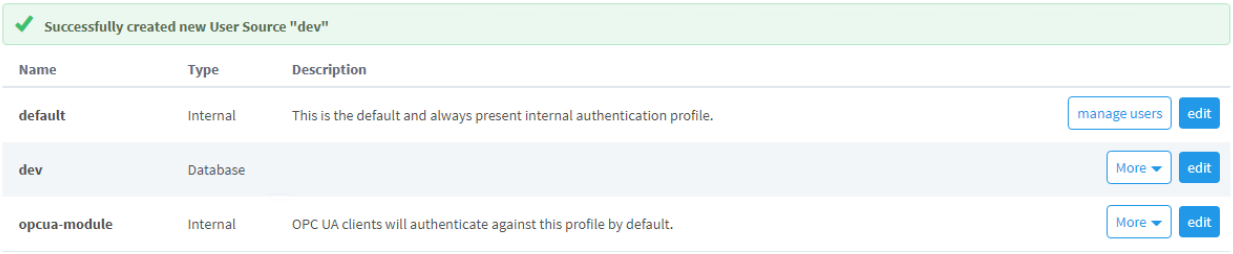
Automatic Mode	
Tablename Prefix	scada_ When in automatic mode, tables will be created to store the users and roles. They will have this prefix. (default: scada_)



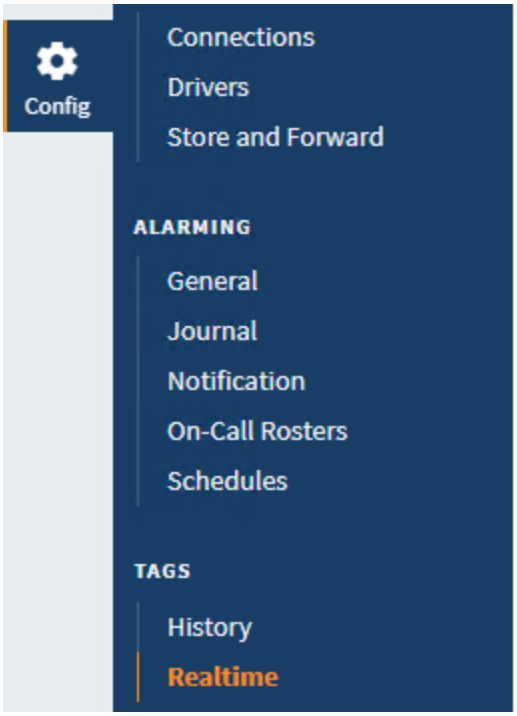
Manual Mode

Authentication Query	SELECT SUBSTRING_INDEX(SUBSTRING_INDEX(emp_sort, ' ', 1), ' ', -1) as firstname, SUBSTRING_INDEX(SUBSTRING_INDEX(emp_sort, ' ', 2), ' ', -1) as lastname FROM IgnitionE A query that must return a row if the given username and password combination provided is valid. The query will be run as a prepared statement, so use question mark (?) to represent username first and then password. The returned row may contain the user's basic properties under the column names: [firstname, lastname, schedule, language, notes, badge] Example: `SELECT firstname, lastname, schedule FROM USERS WHERE username = ? AND password = MD5(?)`
Badge Authentication Query	A query that must return a row if the given badge provided is valid. The query will be run as a prepared statement, so use question mark (?) to represent the badge. The returned row must contain the username and may contain the user's basic properties under the column names: [firstname, lastname, schedule, language, notes, badge] Example: `SELECT username, firstname, lastname, schedule FROM USERS WHERE badge = ?`
List Roles Query	SELECT CONCAT(RoleName, ';', IgnitionRoleDomain) as rolename FROM IgnitionRoles A query that returns all possible roles that any user could be a member of. The role names must be returned in the first column of the query's results. Example: `SELECT rolename FROM roles`
User's Roles Query	SELECT CONCAT(IgnitionRoleName, ';', IgnitionRoleDomain) as rolename FROM IgnitionRoleMapping WHERE IgnitionEmployee = ? A query that returns all of the roles that the provided user belongs to. The roles must be strings and must be in the first column of the query's results. The query will be run as a prepared statement with one parameter: the username. Example: `SELECT rolename FROM mapping_table WHERE username = ?`
List Users Query	SELECT emp_addr AS username, SUBSTRING_INDEX(SUBSTRING_INDEX(emp_sort, ' ', 1), ' ', -1) as firstname, SUBSTRING_INDEX(SUBSTRING_INDEX(emp_sort, ' ', 2), ' ', -1) as A query that returns a row containing each username. There must be at least one column: the username. Other columns are optional, supported columns are: [username, firstname, lastname, schedule, language, notes, badge].
Contact Info Query	A query that returns all of the contact info for the user. The first column must be the contact type, the second column the contact value. Optional, may be blank. Example: `SELECT contact_type, contact_value FROM user_contact WHERE username=?`
Schedule Adjustment Query	A query that returns the upcoming schedule adjustments for the user. Columns must be Start(date), End(date), Available(boolean), Note(string). Optional, may be blank. Example: `SELECT start_date, end_date, is_available, note FROM user_schedule_adj WHERE username=?`
Extra Properties Query	A query that returns name, value pairs of extra properties for the user. Will be run with one parameter: the username. Optional, may be blank. Example: `SELECT prop_name, prop_value FROM user_props WHERE username=?`

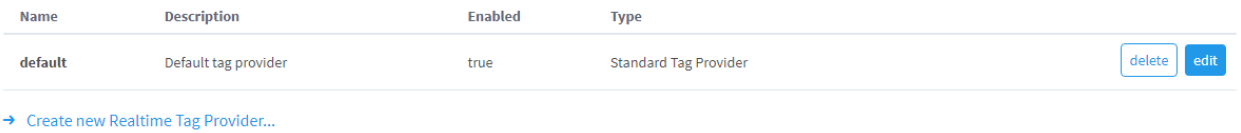
26. After the User Source is saved, a confirmation message is displayed.



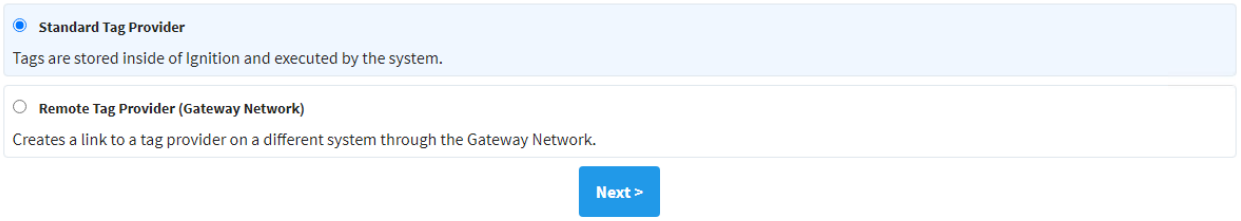
27. Create a new Realtime Tag Provider by selecting Tags > Realtime from the Config menu.



28. Select Create new Realtime Tag Provider.

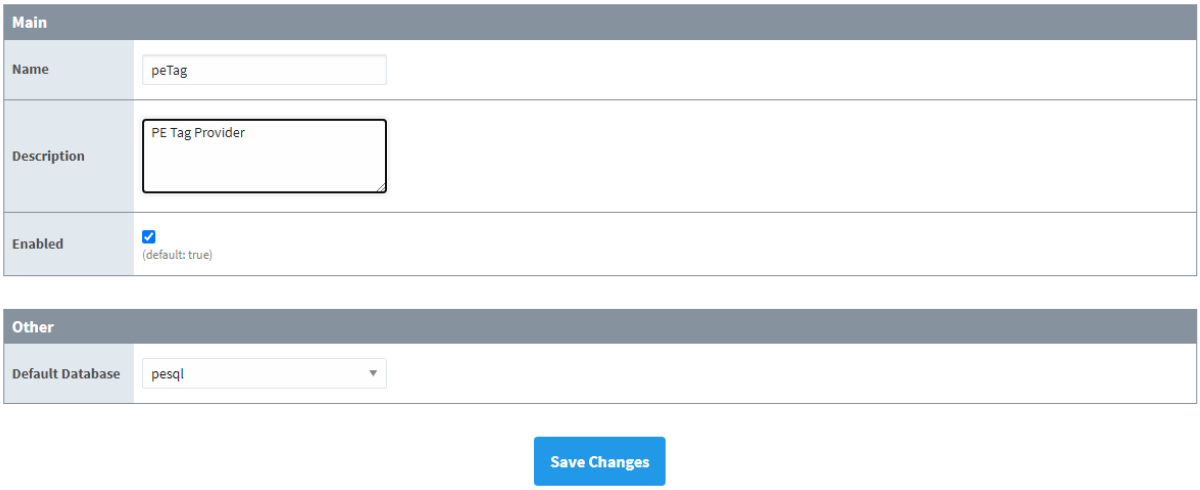


29. Select Standard Tag provider.

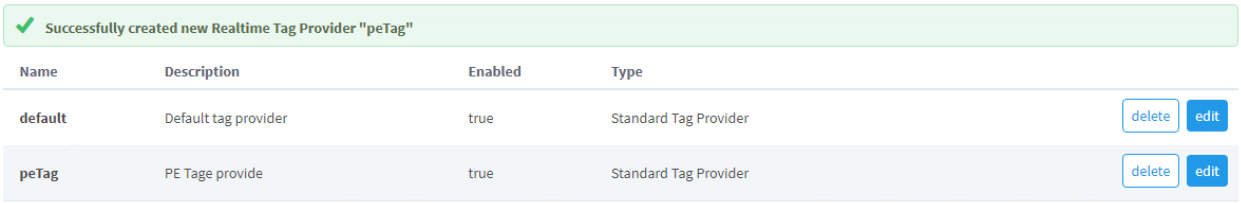


30. Enter Realtime Tag Provider information:

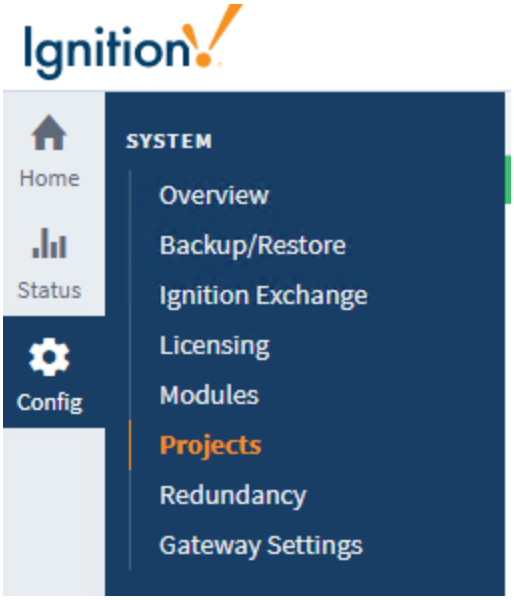
Field	Setting
Name	peTag
Description	PE Tag Provide
Enabled	Yes/Check box selected
Default Database	pesql



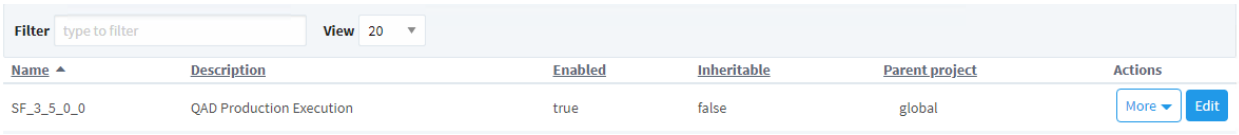
31. After the Realtime Tag Provider is saved, a confirmation message is displayed.



32. Next, edit the Project so that it can use the pesql database and peTag. Select Projects from the Config menu.

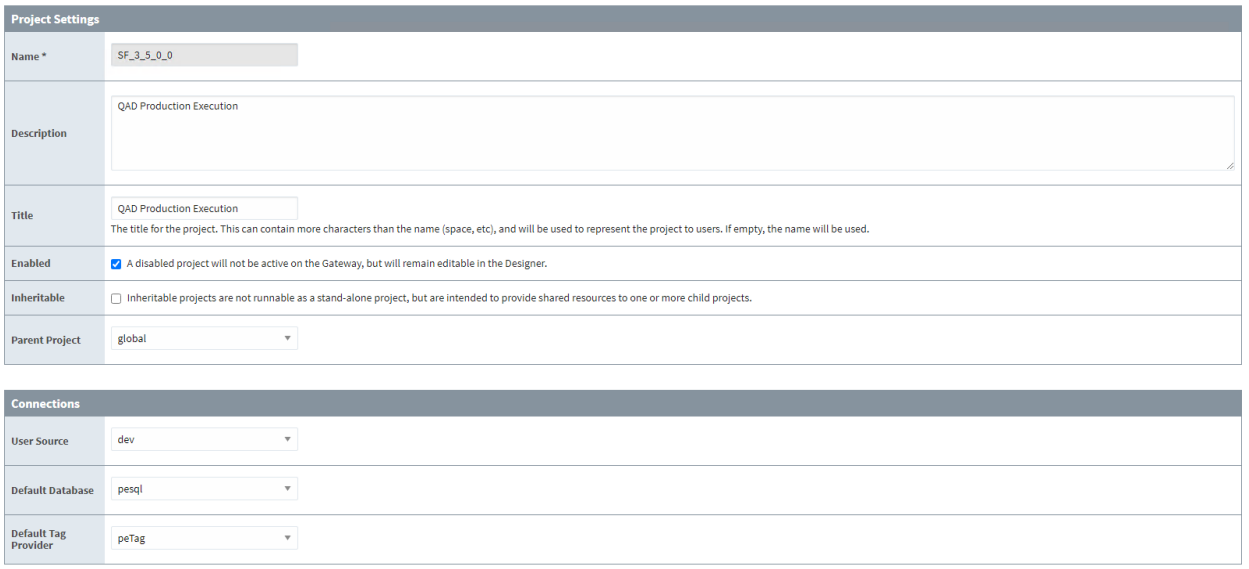


33. For the SF_3_5_0_ Project, select Edit.



34. Enter the following information:

Field	Setting
User Source	dev
Default Database	pesql
Default Tag Provider	peTag



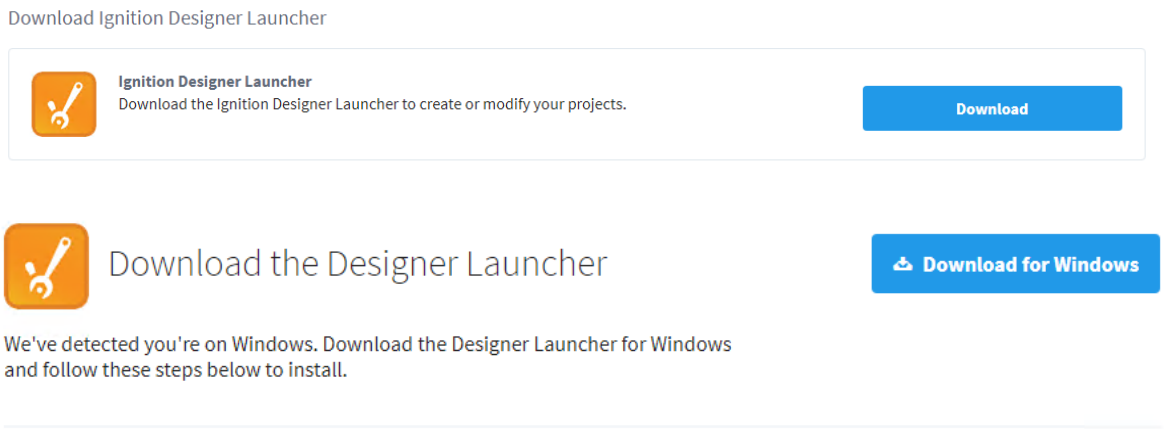
Install and Configure the Ignition Designer

This section contains instructions to configure the Ignition Designer.

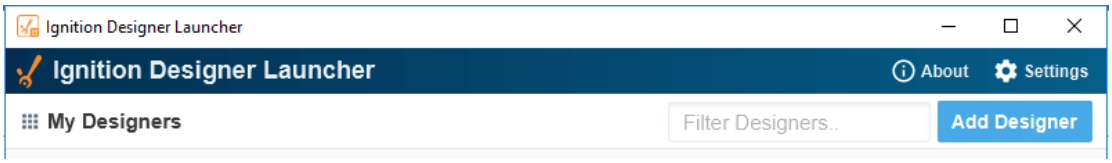
In this procedure, you will,

- Download and install Ignition Designer.
- Import Tags and Tag Groups, which provide additional functionality for PE.
- Import the images/icons that are displayed on the Production Execution screens.

1. Download the `DesignerLauncherSetup.exe` file from the Ignition Gateway homepage. Open the .exe file and follow the install instructions on the screens.



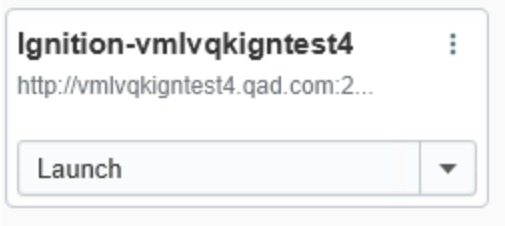
2. Once installation is complete, open the Ignition Designer and select Add Designer.



3. Select Manually Add Gateway.

4. Enter the Gateway URL: `http://<hostname>:<gateway port>` and then select Add Gateway.

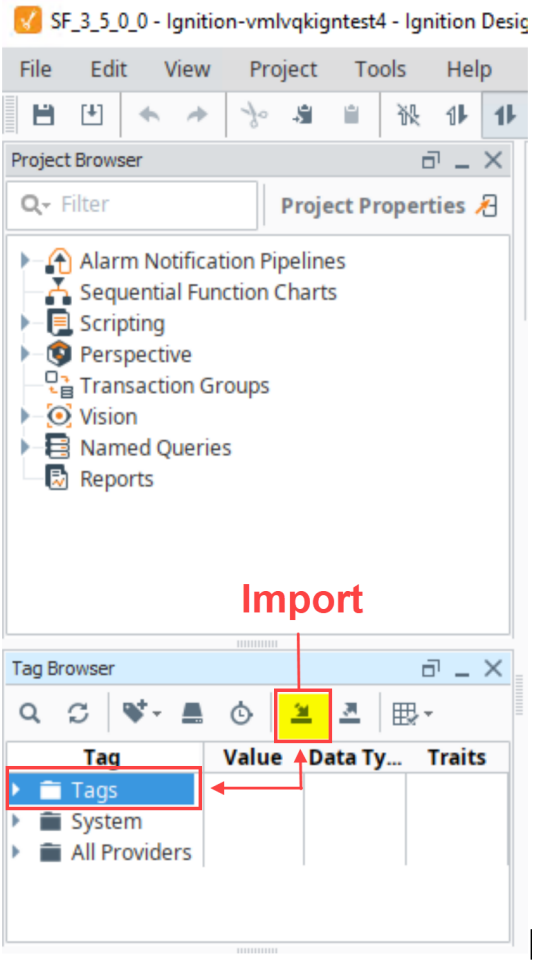
5. Select Launch.



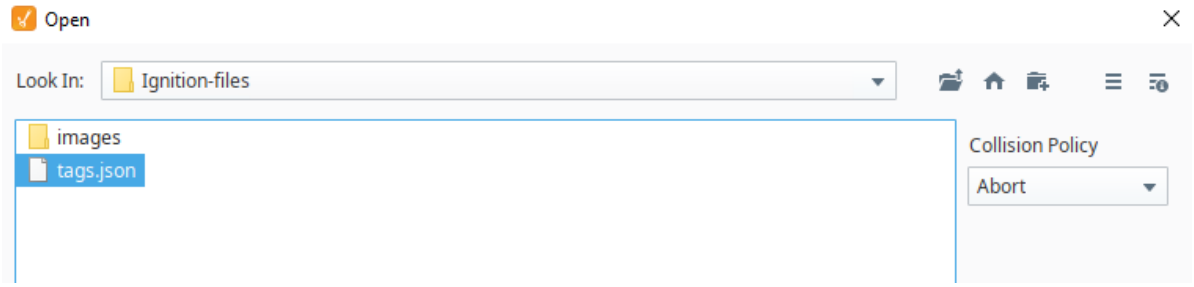
6. Enter the Ignition gateway User credentials.

7. Select the `SF_3_5_0_0` Project and then select Open.

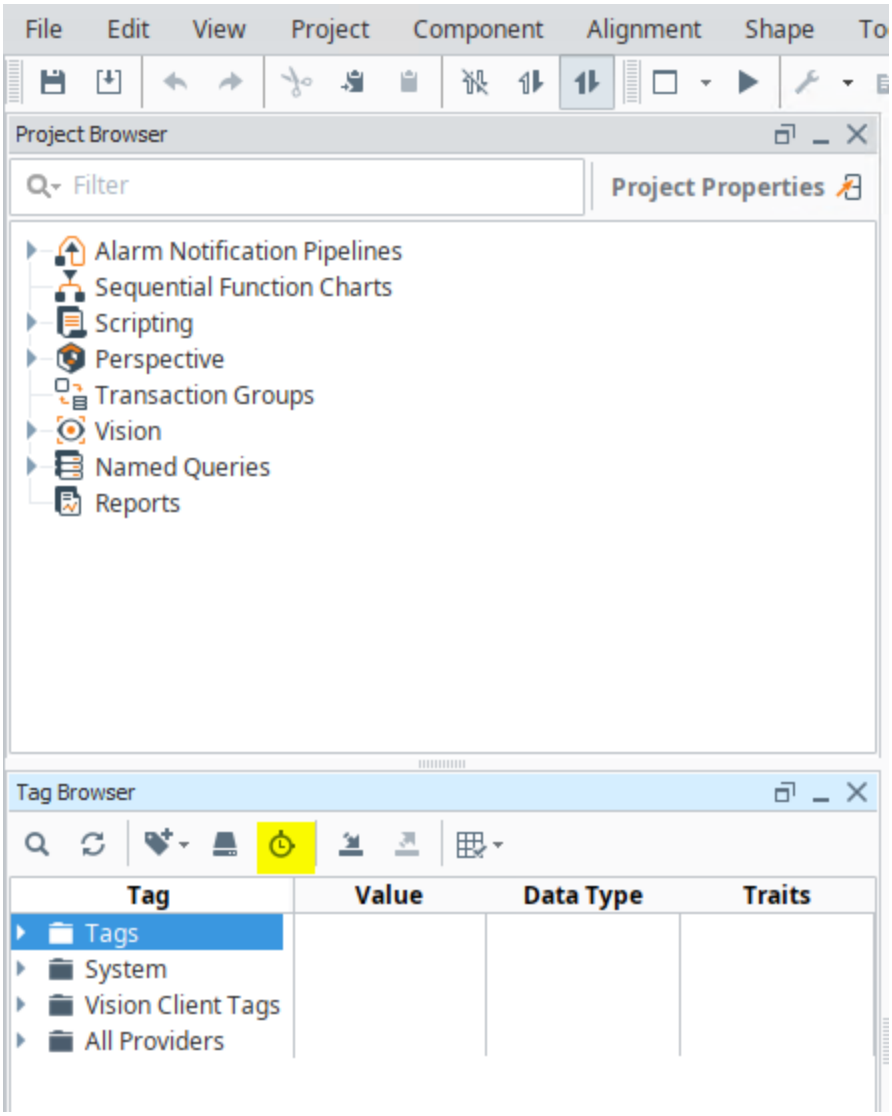
8. Importing Tags by highlighting Tag and selecting Import.



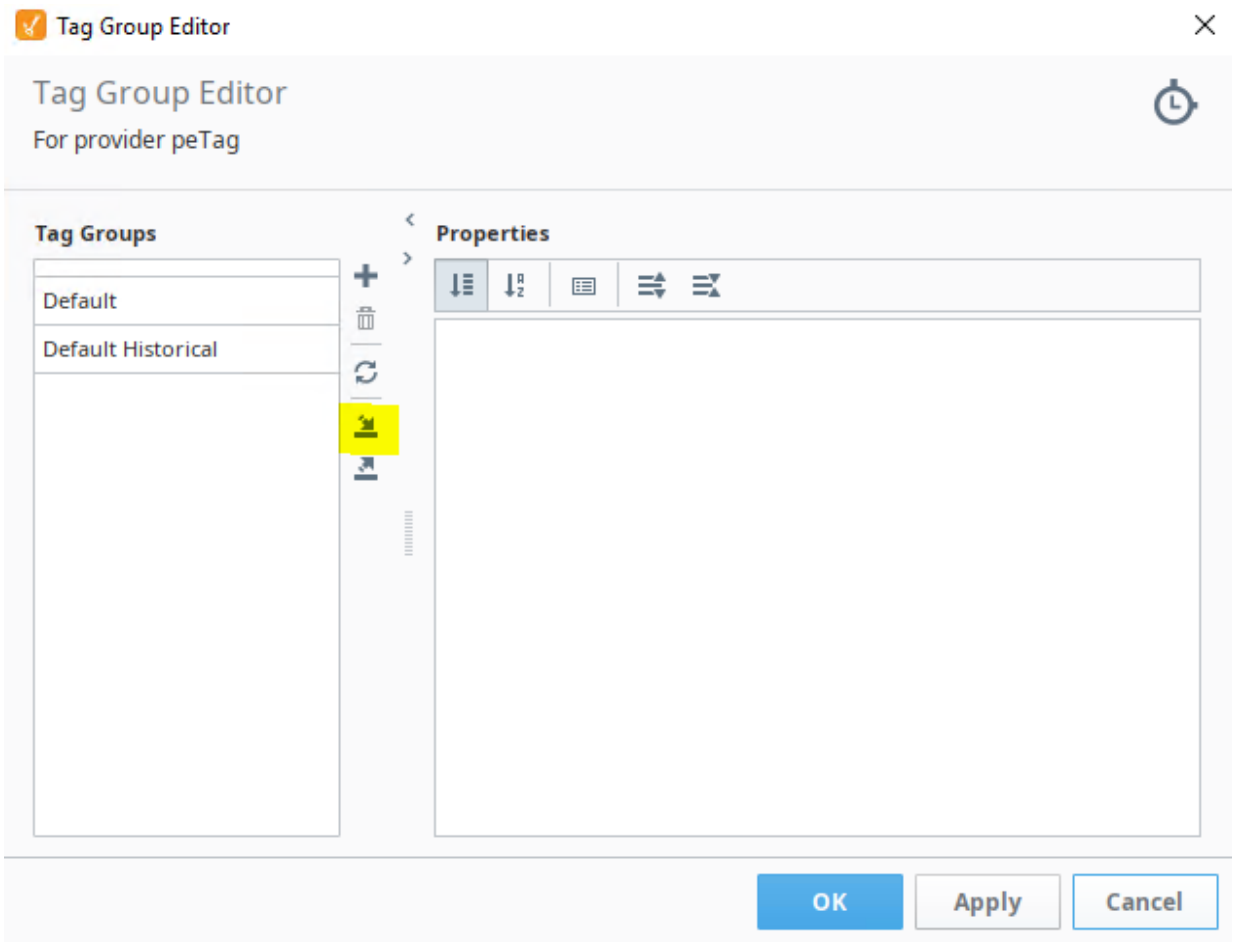
9. Select the `tags.json` file to import. This file is available in the PE Ignition package.



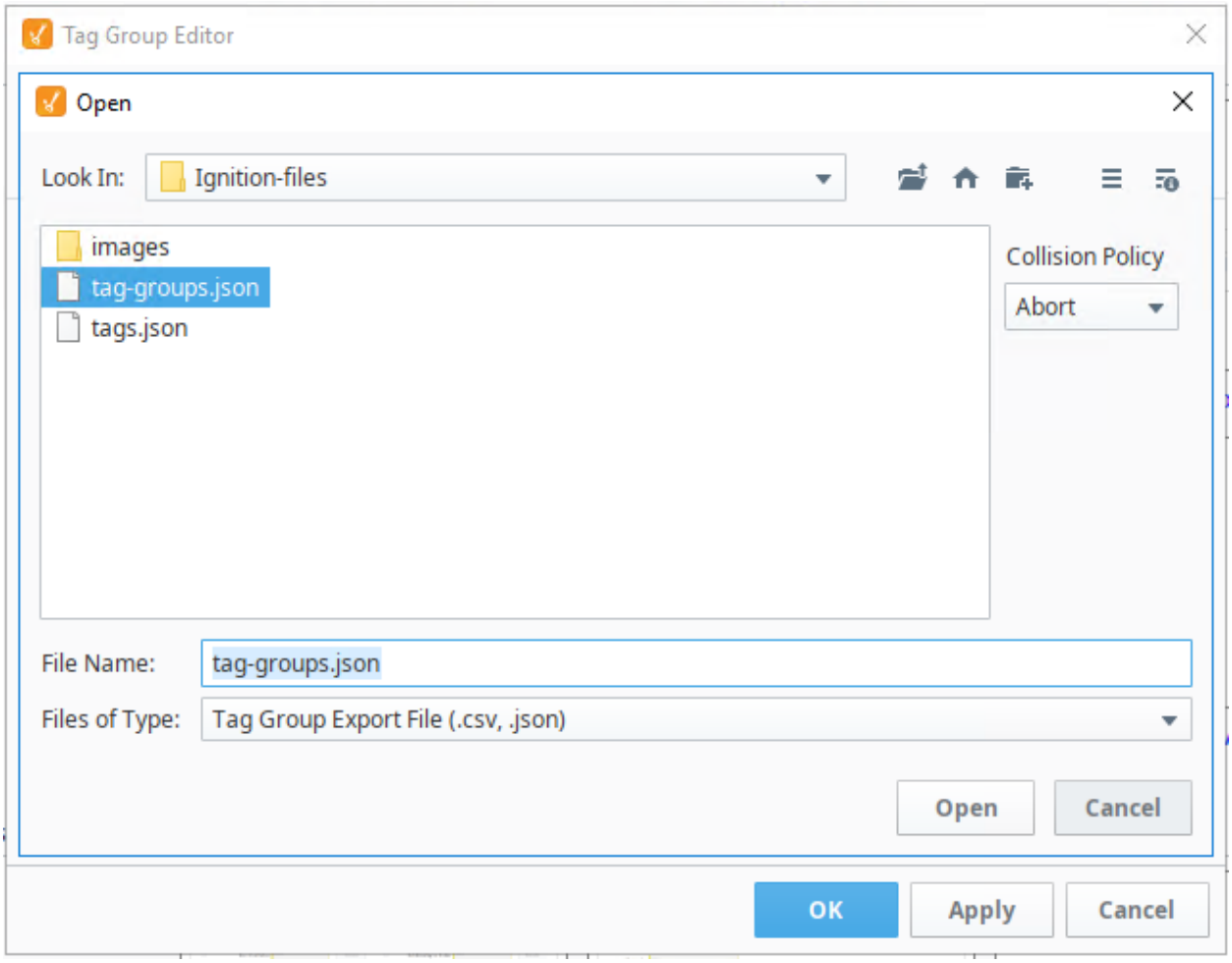
10. Select Tags and then the Edit Tag Group.



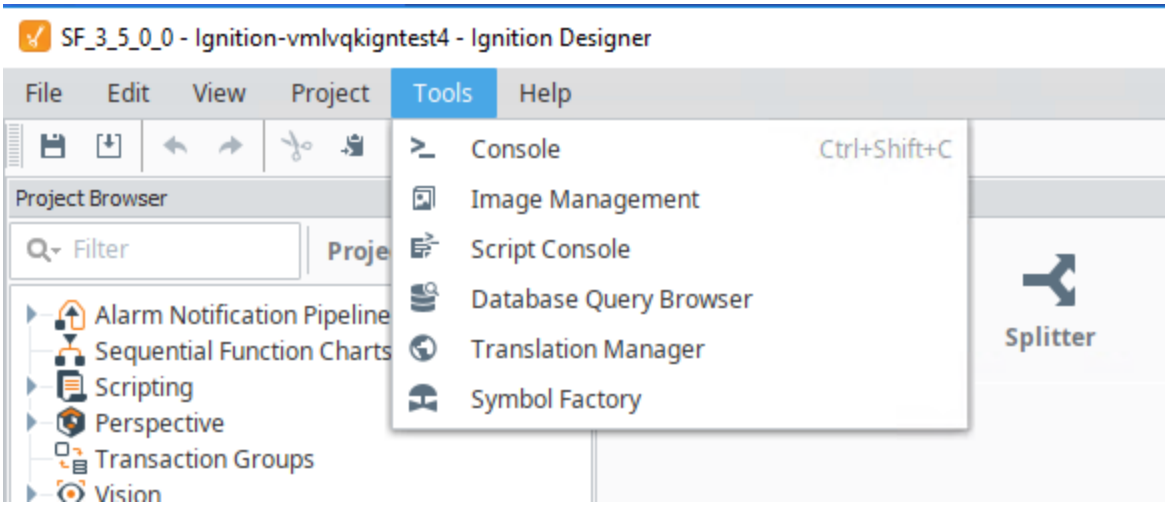
11. From the Tag Group Editor Dialog box, select Import.



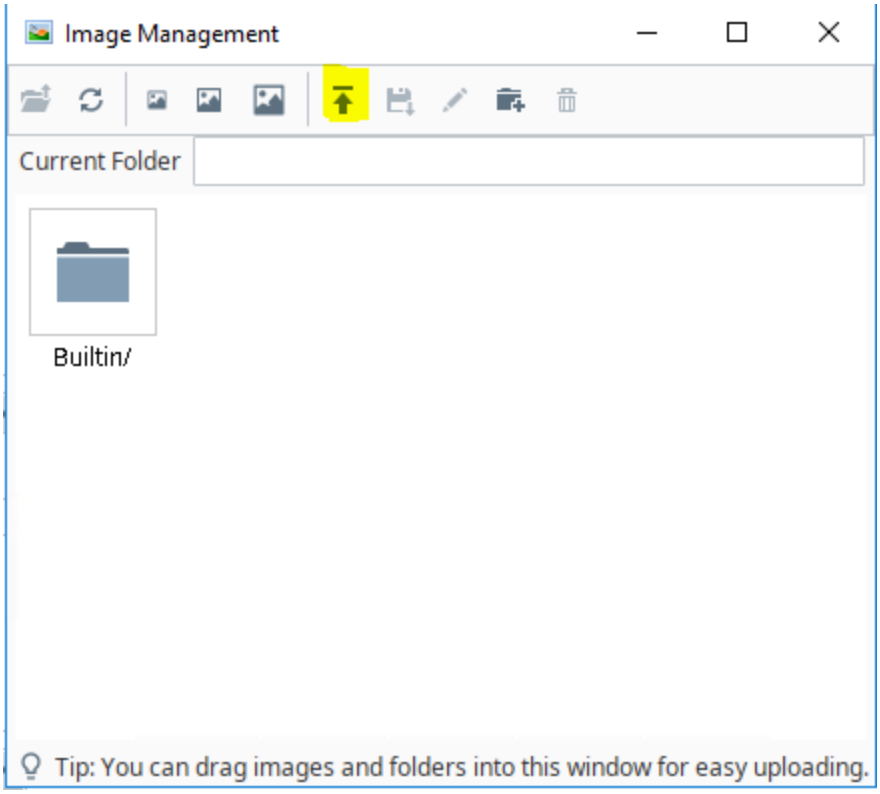
12. Select the location where the `tag-groups.json` file is saved. This file is available in the PE Ignition package.



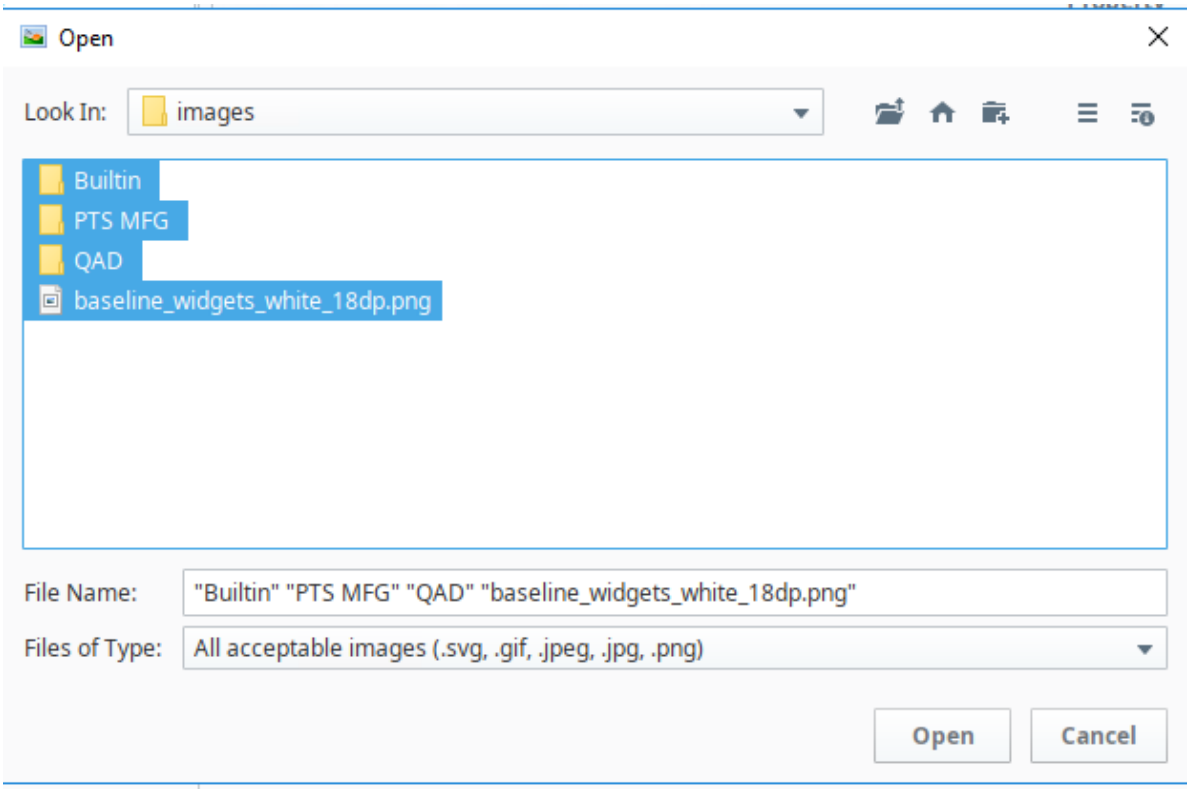
13. Next, import images/icons by extracting the `image.zip` file from the package into a temporary location. Select Tools from the menu and then select Image Management.



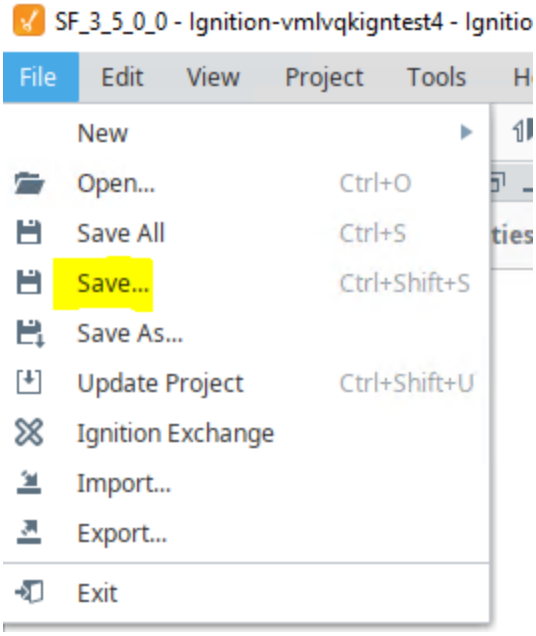
14. In the Image Management window, select Upload new image.



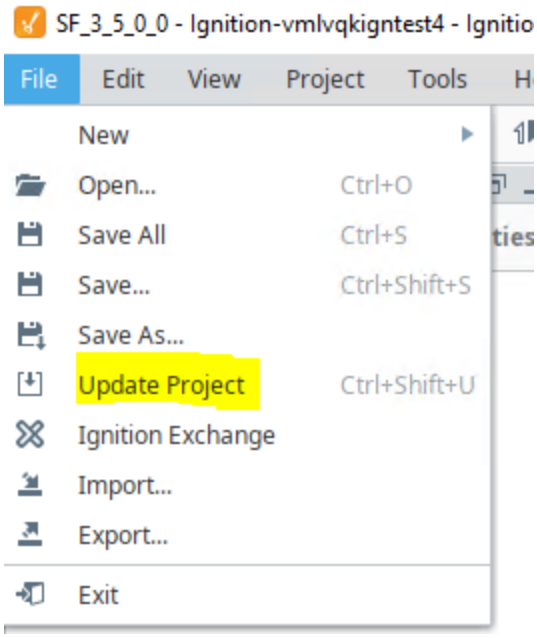
15. From the temporary location where the unzipped images are located, select all files and folders and then select Open.The Images.zip file is provided with the PE Ignition package.



16. It may take a few minutes for Ignition to import the images. Once the import is completed, close the Image Management window.
17. Save the Project changes by selecting Save.



18. Then select Update Project.

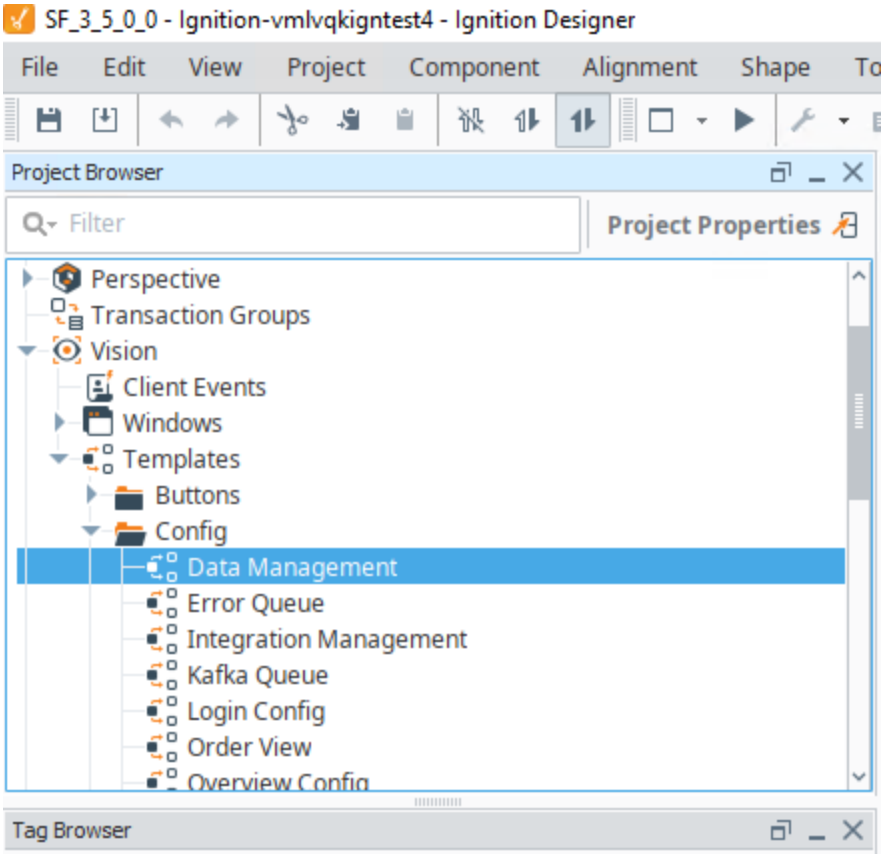


Troubleshooting

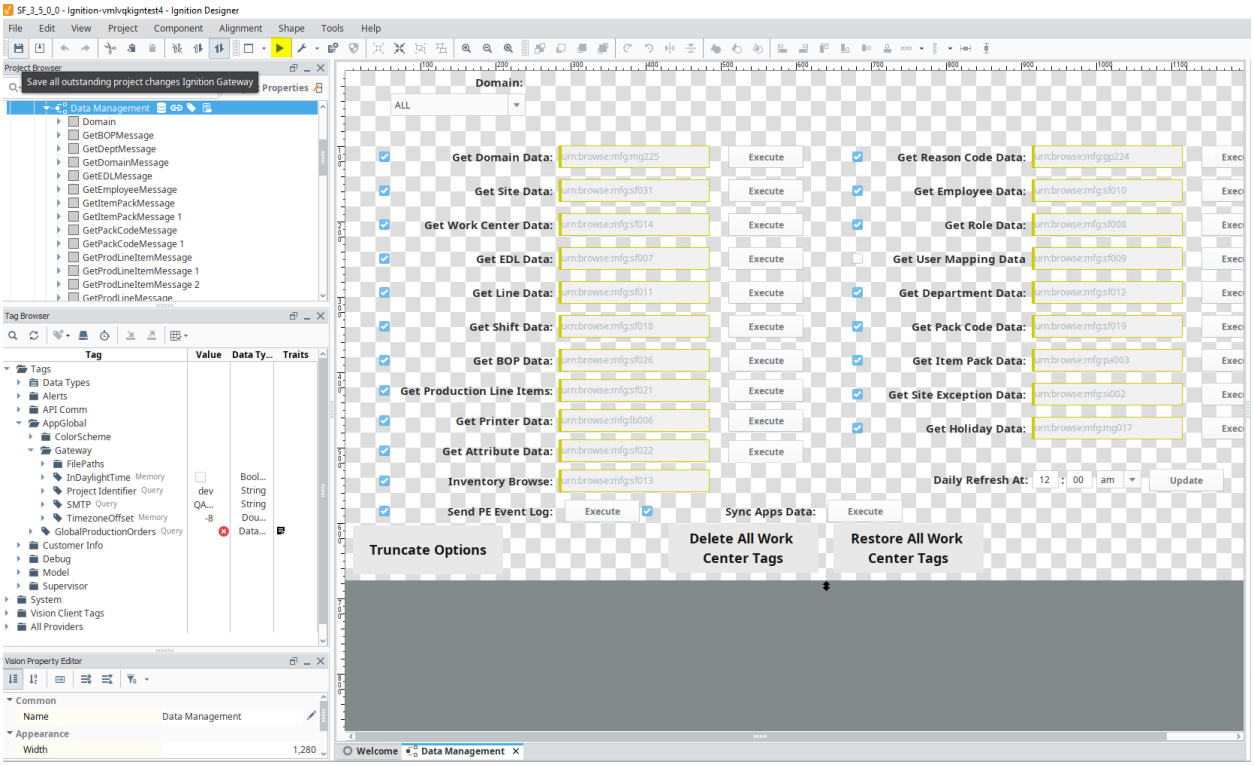
Login Problems - Ignition Vision Client

Follow these steps if you cannot login to the Ignition Vision client:

1. In Designer, select Vision > Templates >Config > Data Management.



2. In the Data Management screen, select Play from the top menu bar.



3. Then select Delete All Work Center Tags. Wait a few seconds while Ignition completes the deletion process.
4. Select Restore All Work Center Tags.
5. Close the Data Management tab without saving.

EQMS PE Integration Configuration

This section only applies if EQMS is installed.

1. Download and install the Web Service module in Ignition. Make sure to install the same version as the Ignition version that is installed:

   https://inductiveautomation.com/downloads/third-party-modules

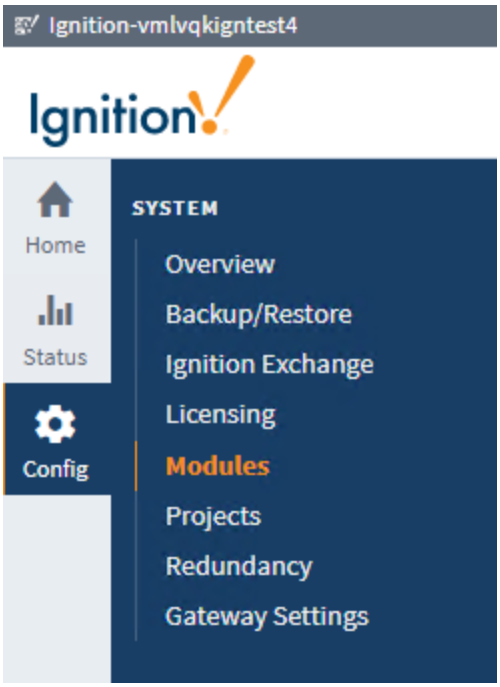
2. Download the Web Services Module file (`Web Service-module.modl`) for Sepasoft, Inc.

Sepasoft, Inc. MES Modules for Ignition	Version	Checksum
Web Services Module (4.7 MB)	2.80.0.2006162356 STABLE	sha-256

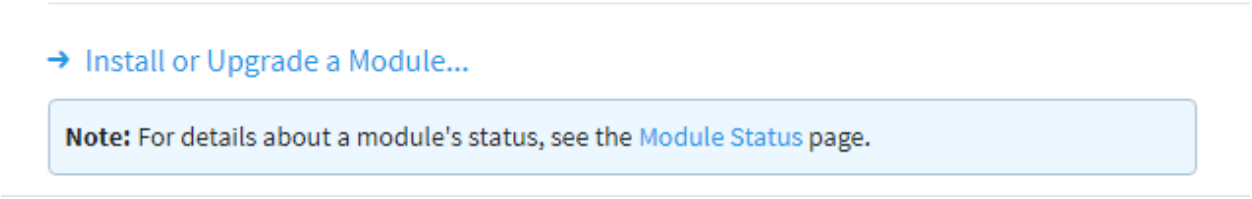
3. Open Ignition Gateway in the web browser.

   `http://<ignitionhostname>:<port>`

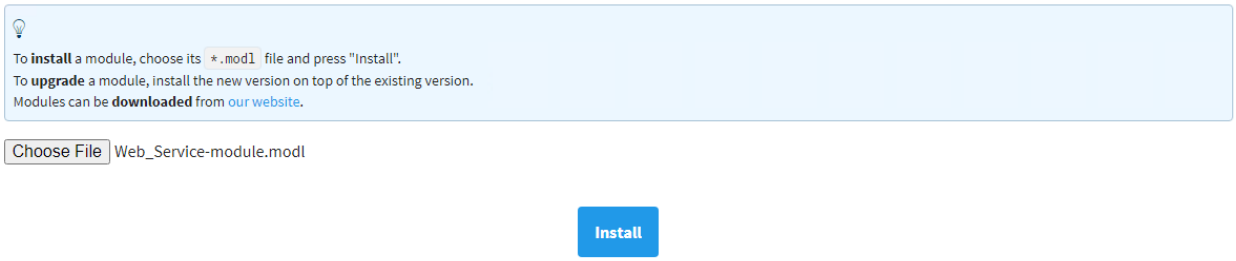
4. Select Modules from the Config menu.



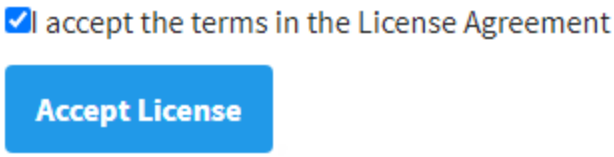
5. Scroll to the bottom of the page and select Install or Upgrade a Module.



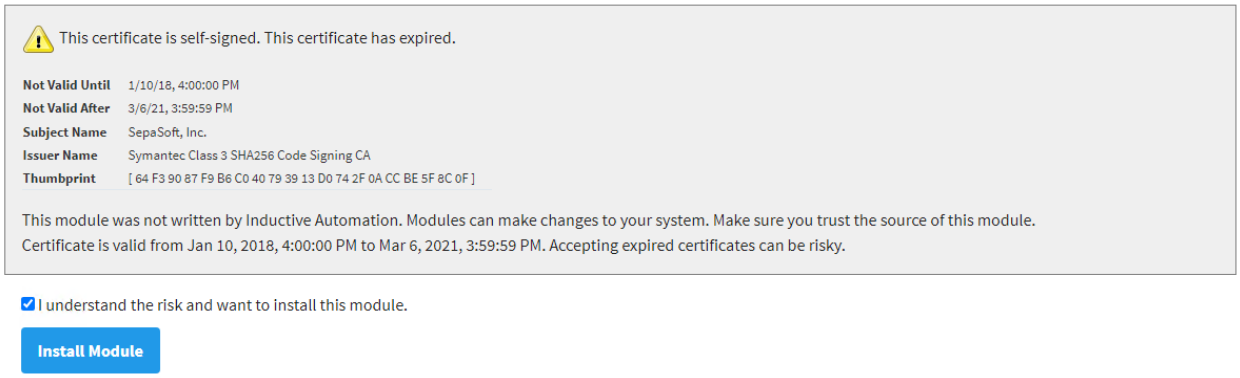
6. Select Choose File and then select the downloaded `Web_Service-module.modl`. Select Install to install the module.



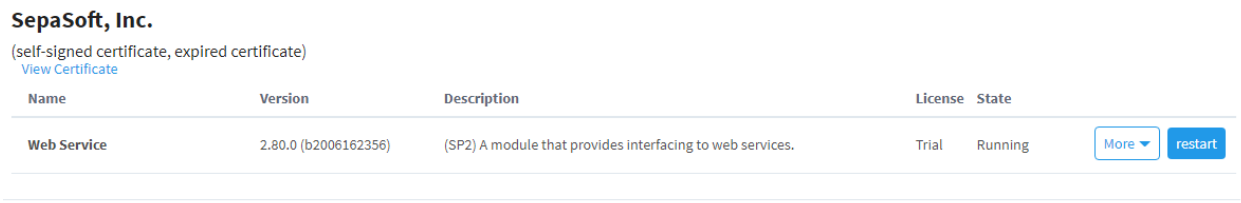
7. Read and accept the terms in the License Agreement and then select Accept License.



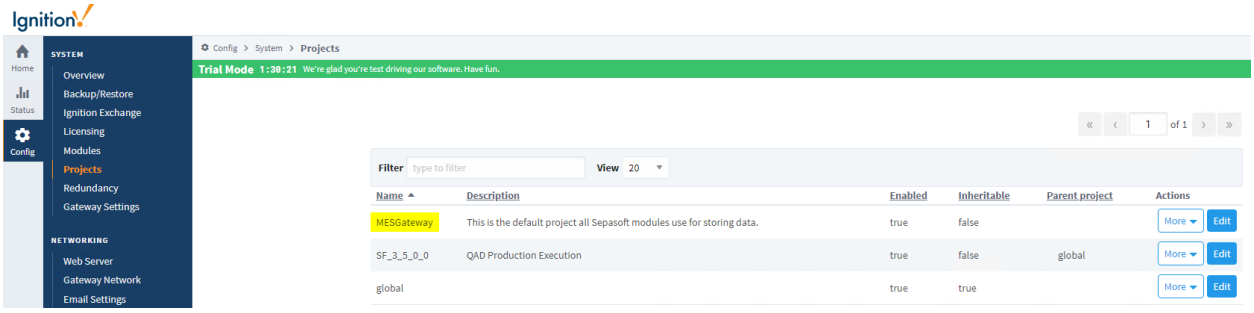
8. Accept the self-signed certificate and then select Install Module.



9. After the Module is installed, the Config > Module screen will display the SepaSoft, Inc. module for web service.

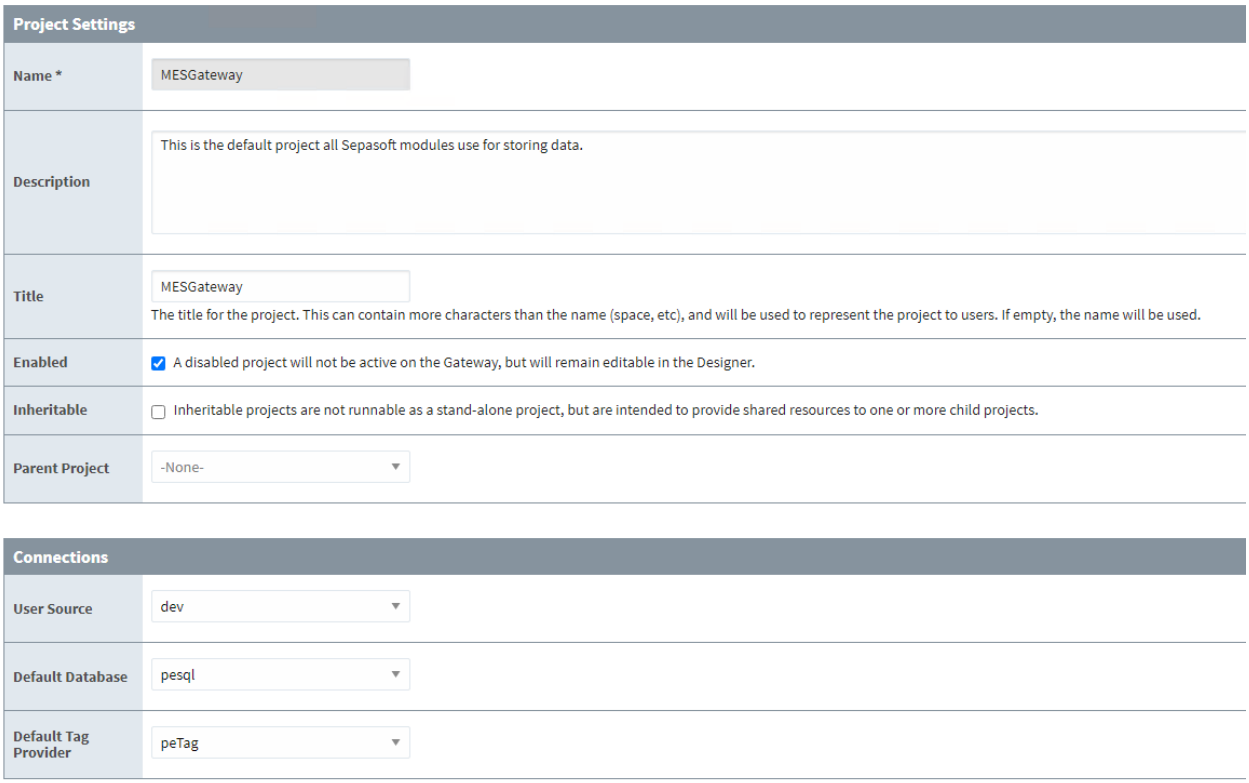


10. The new MESGateway project, which is now available in the Config >Project Web screen, needs to be updated to assign the correct Database and Tags.
11. Select Edit for the MESGateway project.

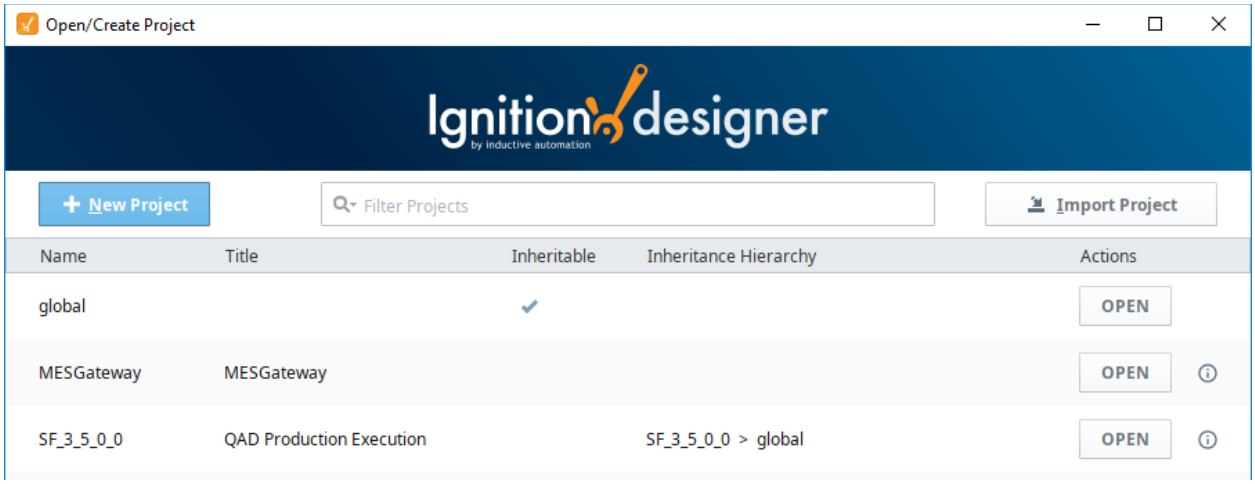


12. Assign the Database and Tags to the MESGateway project and then select Save.

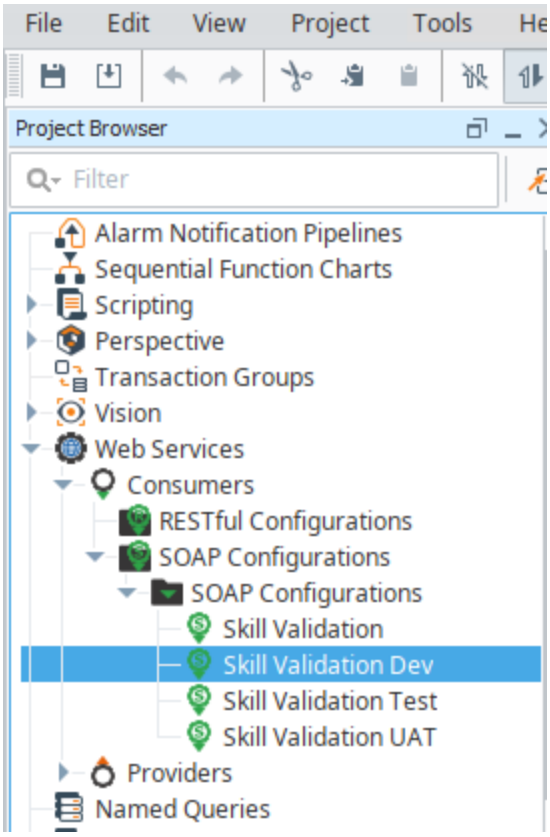
Field	Setting
User Source	dev
Default Database	pesql
Default Tag Provider	peTag



13. Next, configure the Web Services module in Ignition Designer to point to the EQMS host. Open the Ignition Designer and select the MESGateway Project.

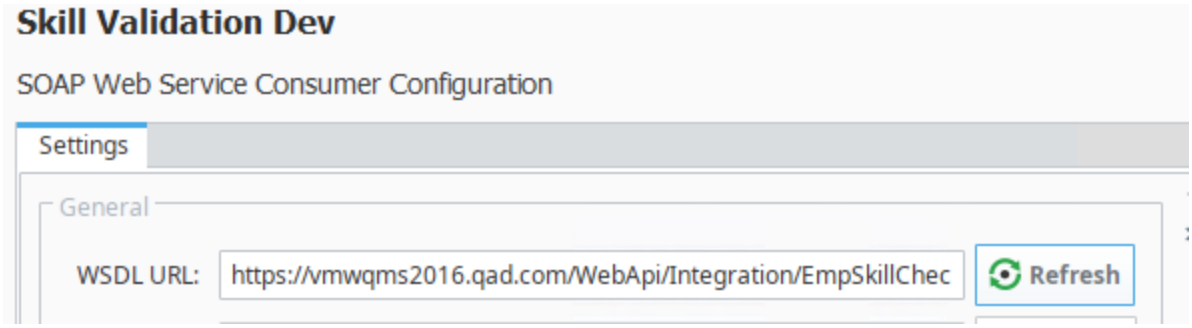


14. From the Project Browser, select Web Services > Consumers > SOAP Configurations > SOAP Configurations > Skill Validation Dev.

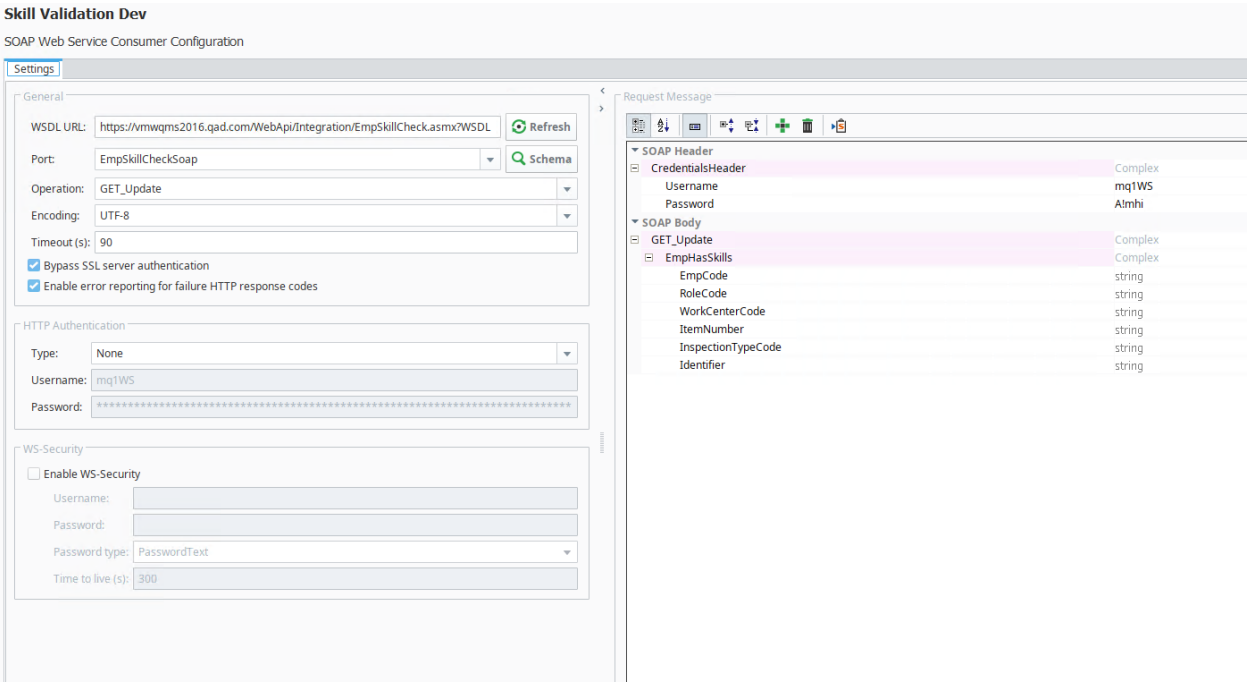


15. In the Skill Validation Dev window, enter the following WSDL link for the QMS host that needs to be integrated:

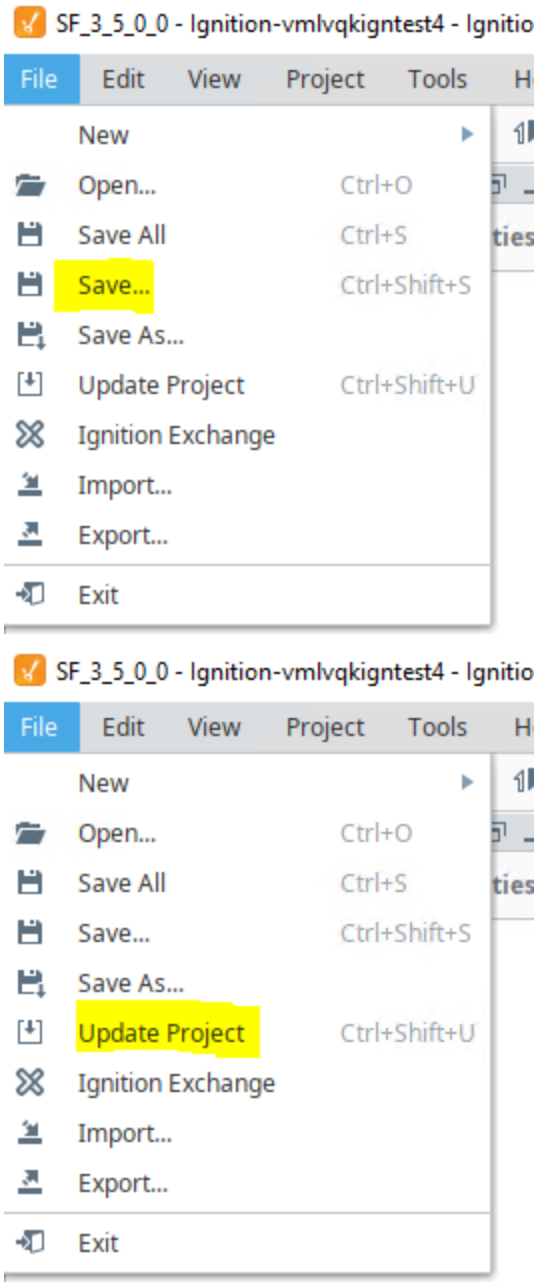
```
https://<EQMS hostname>/WebApi/Integration/EmpSkillCheck.asmx?WSDL
```



16. Select Refresh to refresh all the values on the screen.
17. Enter the values as shown. The WSDL URL must contain the EQMS host name.



18. Save the Project changes and then Update Project.



Updating Nifi Flow Components

Follow these steps if you are currently using Production Execution and are updating the Nifi Flow Components:

1. Deactivate or remove the previous version of the Nifi Flow Components. It is recommended that you deactivate the previous version of the package rather than delete it

because it allows you to easily re-activate the previous version if necessary. See Deactivate Previous Version of Nifi Flow.

2. Install the new version of the Nifi Flow Components using yab. See Installing a New Version of the Nifi Flow Components.

Deactivate Previous Version of Nifi Flow

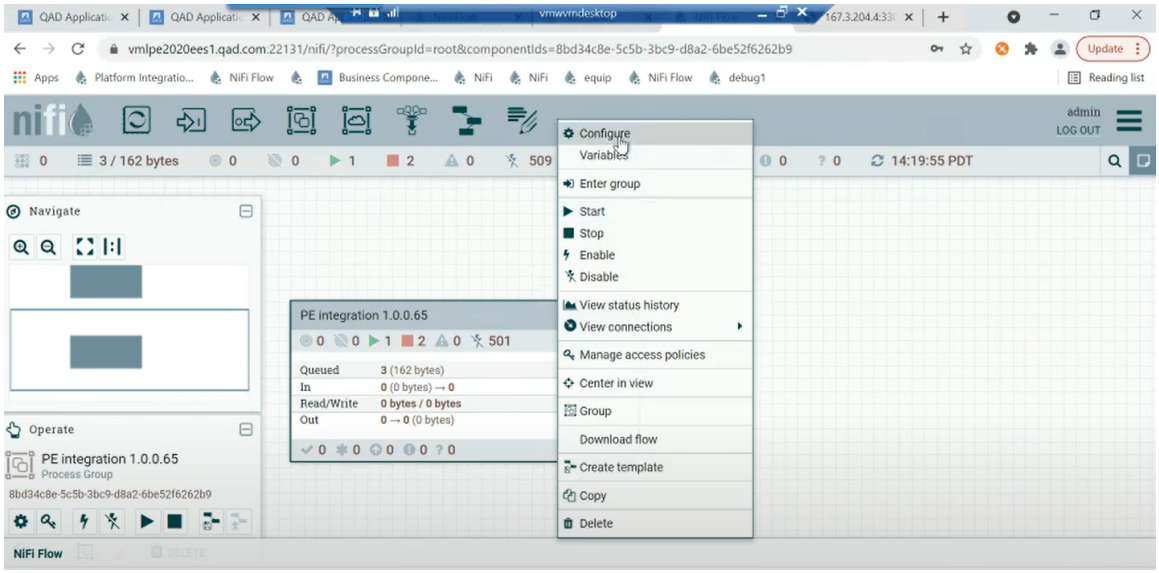
Note: This task only applies if you are upgrading the Nifi flow and there is a prior version of the `penifi` package deployed.

If there is a prior version of the `penifi` package deployed then the Nifi flow needs to be deactivated or removed before installing the new version. This is required because the updated Nifi flow uses the same port as the previous flow.

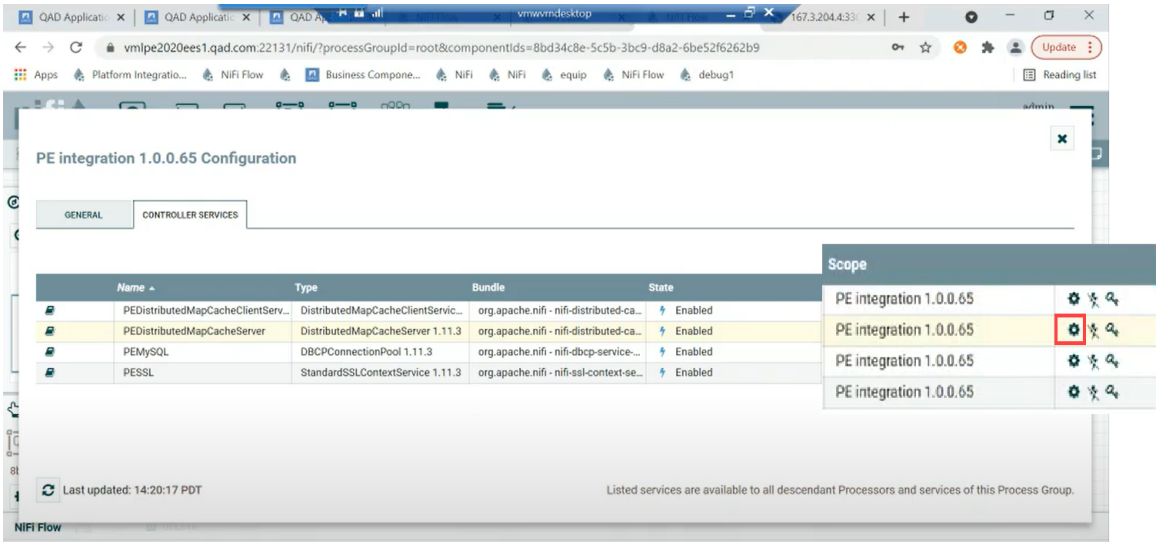
Note: It is recommended that you deactivate the previous version of the `penifi` package rather than delete it because it allows you to easily re-activate the previous version if necessary.

Follow these steps to deactivate the previous version of the `penifi` package:

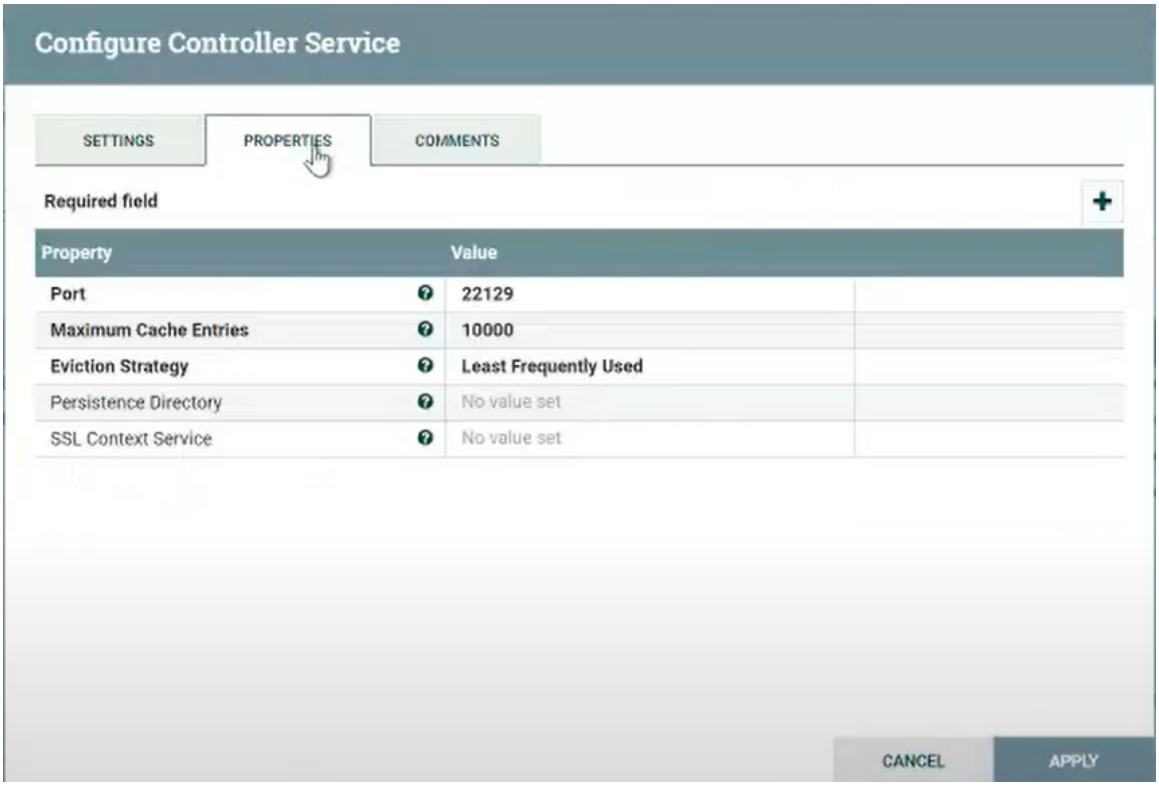
1. In Nifi, right-click on the flow and select Configure.



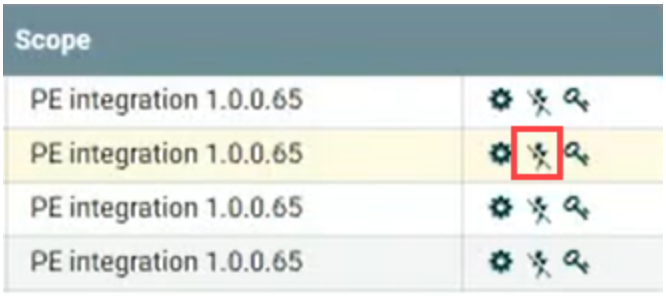
2. In the Configuration window > Controller Services tab, select the Settings button for the `PEDistributedMapCacheServer` piece of the flow.



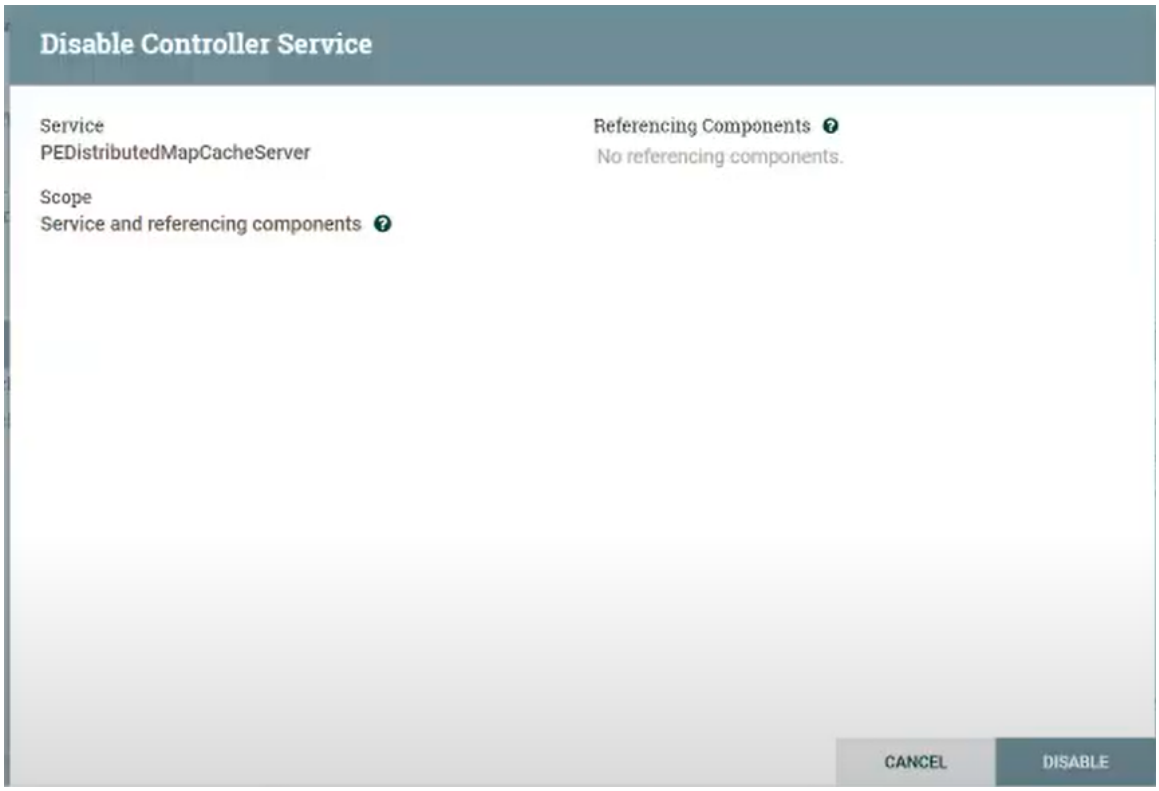
3. A Configure Controller Service window appears. Write down the Port number. If you needed to re-create this piece of the flow, you could create a new piece with the same name (PEDistributedMapCacheServer) and the port number you just recorded. Close the Configure Controller Service window.



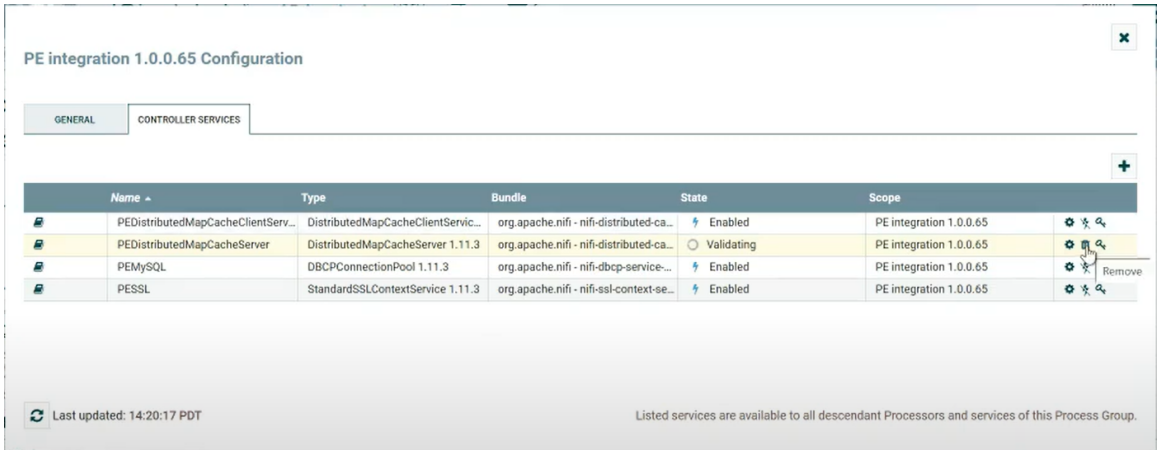
4. Click the Disable button for PEDistributedMapCacheServer.



5. A Disable Controller Service window opens. Select Disable and then close the window.



6. Once you disable PEDistributedMapCacheServer, a Delete button appears, allowing you to delete PEDistributedMapCacheServer. Select the Delete/Remove button.



7. A Delete Controller Service window opens. Select Yes to delete.

8. Because you deleted this piece of the PE integration x.x.x.xx flow, this version of the flow is now disabled and not functional. You can now import the new version of the PE integration x.x.x.xx flow.

   Note: If the new version of the PE integration x.x.x.xx flow does not function properly, you can re-enable the old version by re-creating the piece you just deleted. To re-create the piece, you need the information you recorded in the Configure Controller Service settings window.

Deleting a Nifi Flow Using YAB

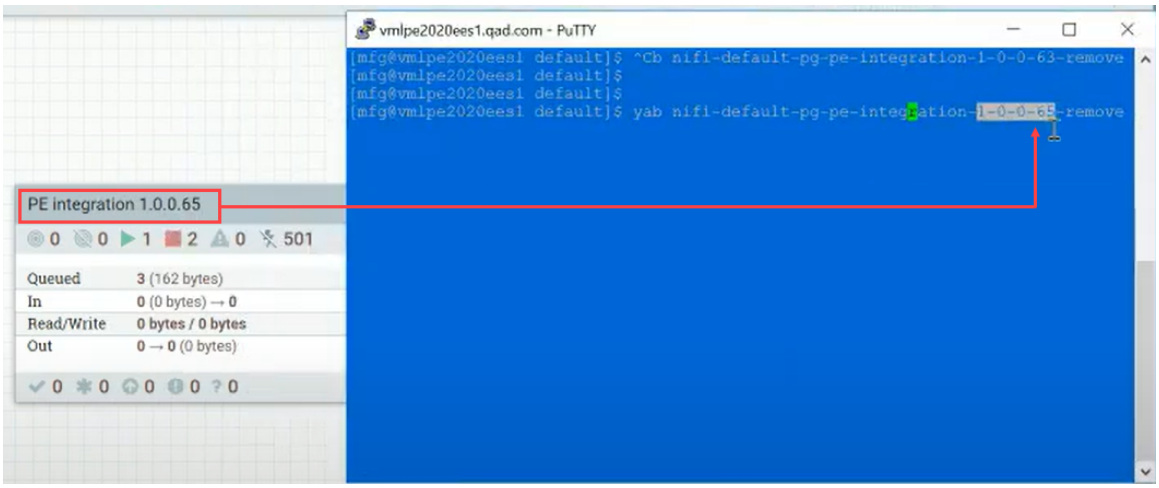
Follow these steps to delete a Nifi flow using yab:

1. Enter the following yab command to view a list of Nifi processes:

```
yab nifi-status
```

2. Then enter the following command to remove the version of the Nifi flow you want to remove:

```
yab nifi-default-pg-pe-integration-x-x-x-x-remove
```



Deleting a Nifi Flow Using Nifi

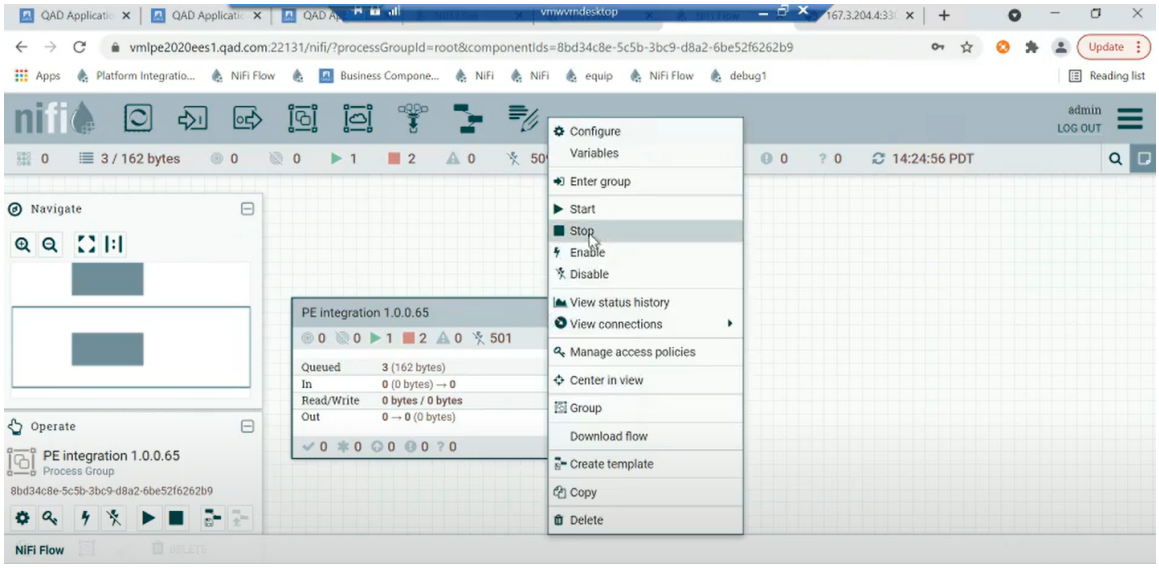
Note: It is recommended that you delete Nifi flows using yab as the process is easier and quicker.

Before deleting a Nifi flow you must do the following; otherwise, you will receive an error:

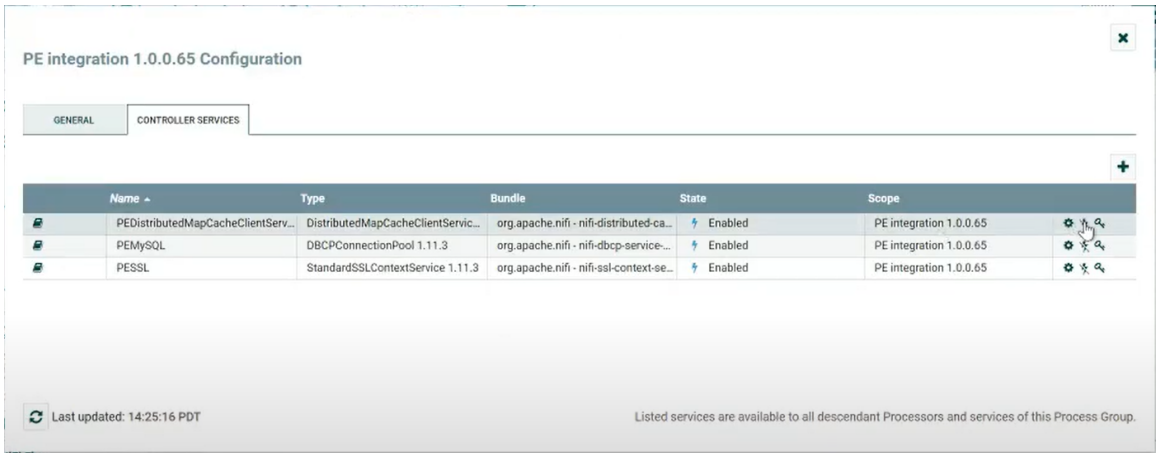
- Disable all associated Controller Services.
- Clear any associated Queues.

Disable Controller Services

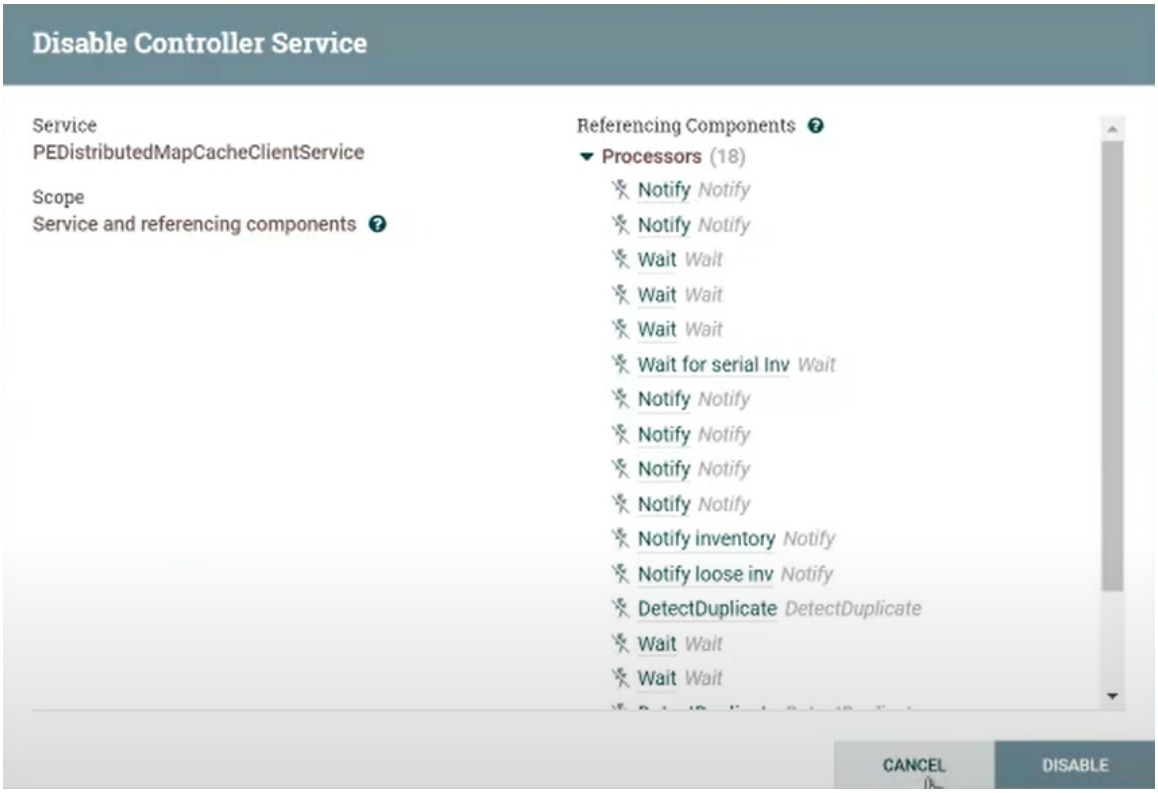
1. Right-click on the Nifi flow and select Stop.



2. Right-click on the Nifi flow again and select Disable.
3. Right-click on the Nifi flow again and select Configure. The Nifi flow Configuration window opens.



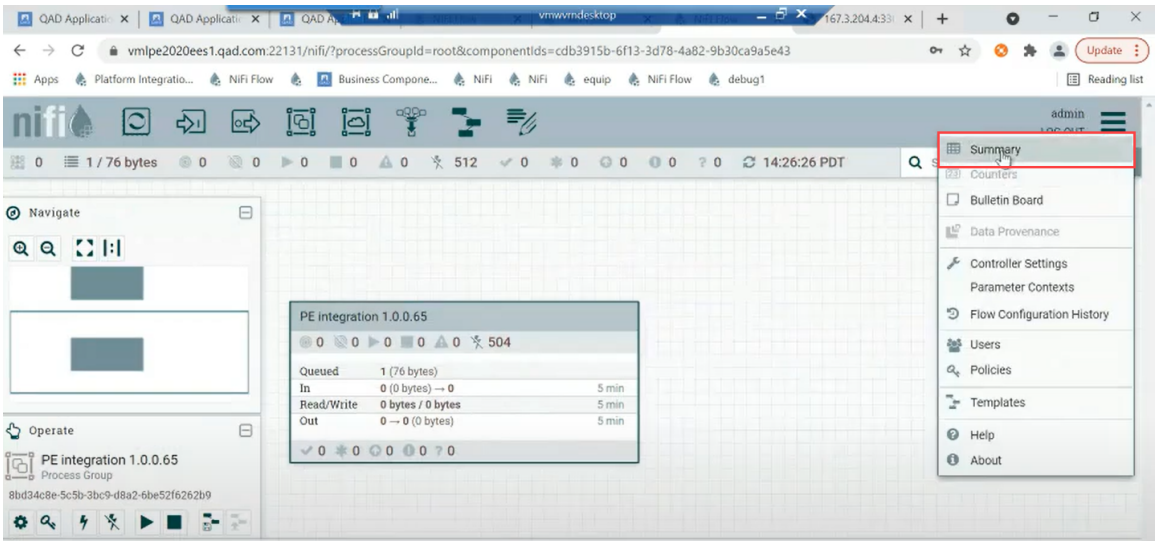
4. Select the Disable button for the first Controller Service. A Disable Controller Service window opens. Select Disable and then close the window.



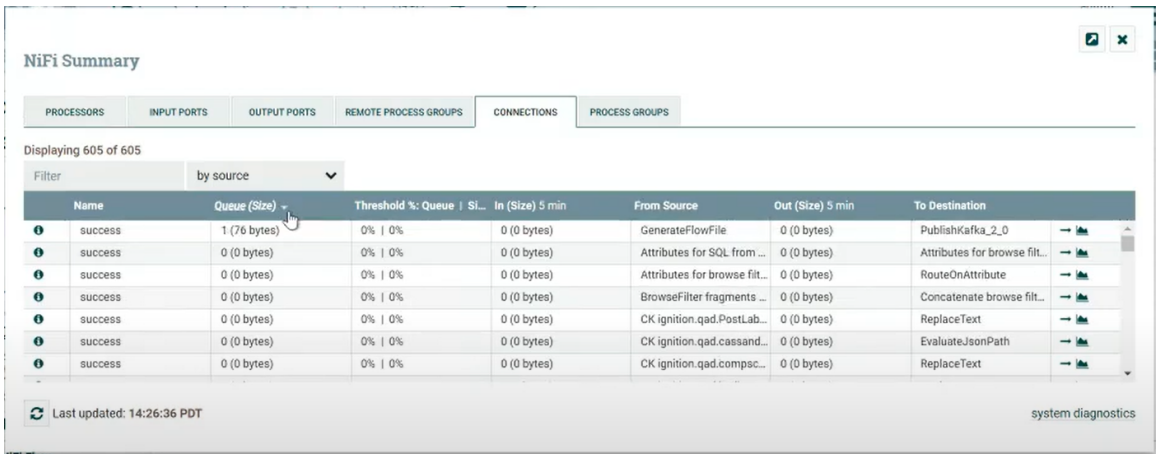
5. Repeat this step for the other Controller Services until all are disabled.

Clear Nifi Flow Queues

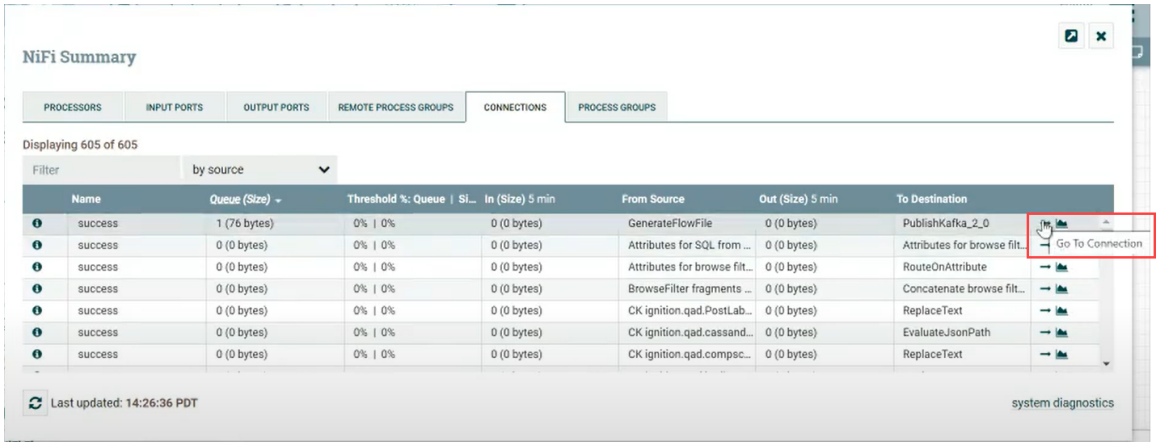
6. Select Summary from the Hamburger menu in the top right of the screen. The Nifi Summary window opens.



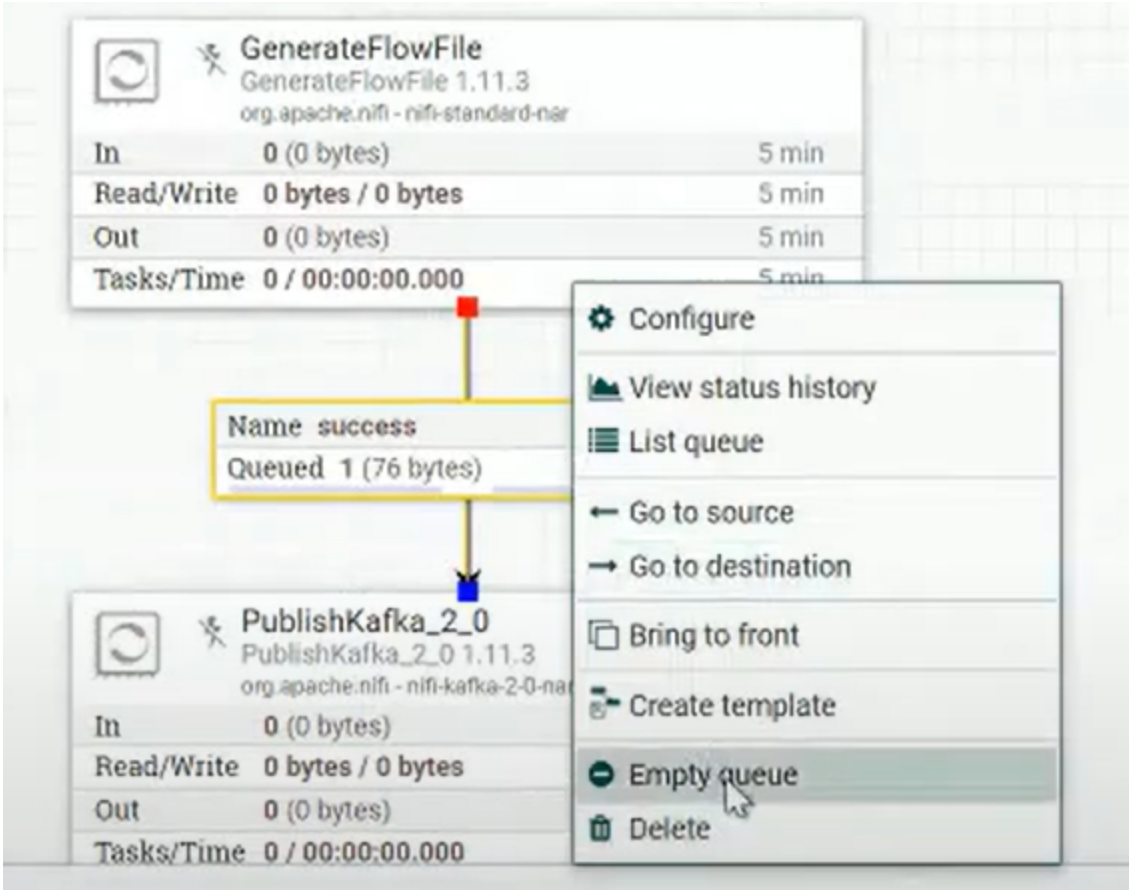
7. On the Connections tab, sort the Queue (Size) column in descending order so that any Queues that are non-zero are displayed first.



8. Select the Go To Connection button for the first Queue.



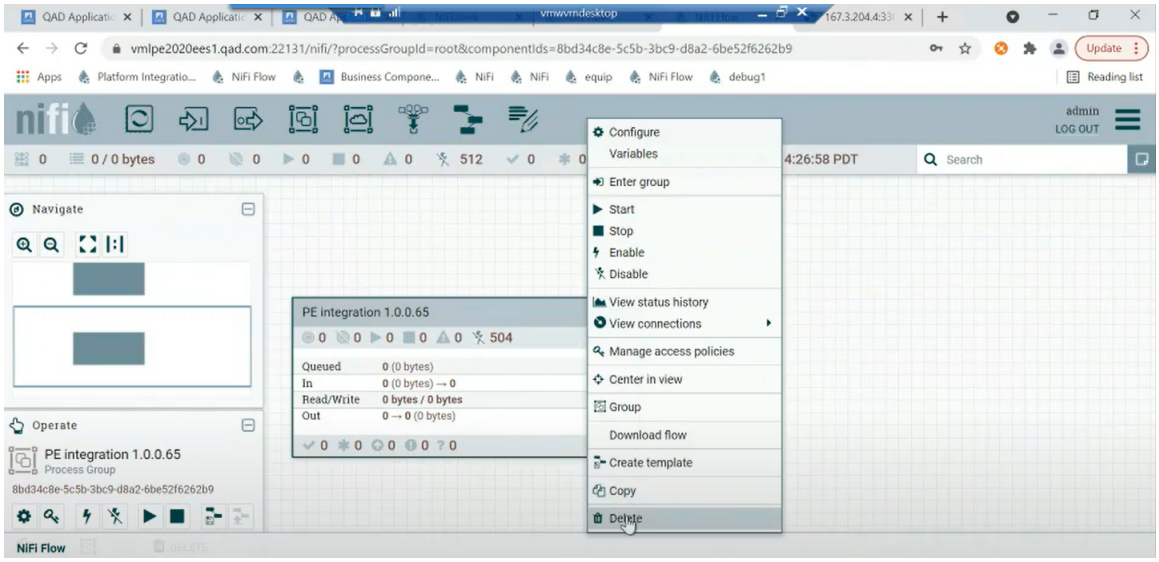
9. Right-click on the Queue and select Empty Queue.



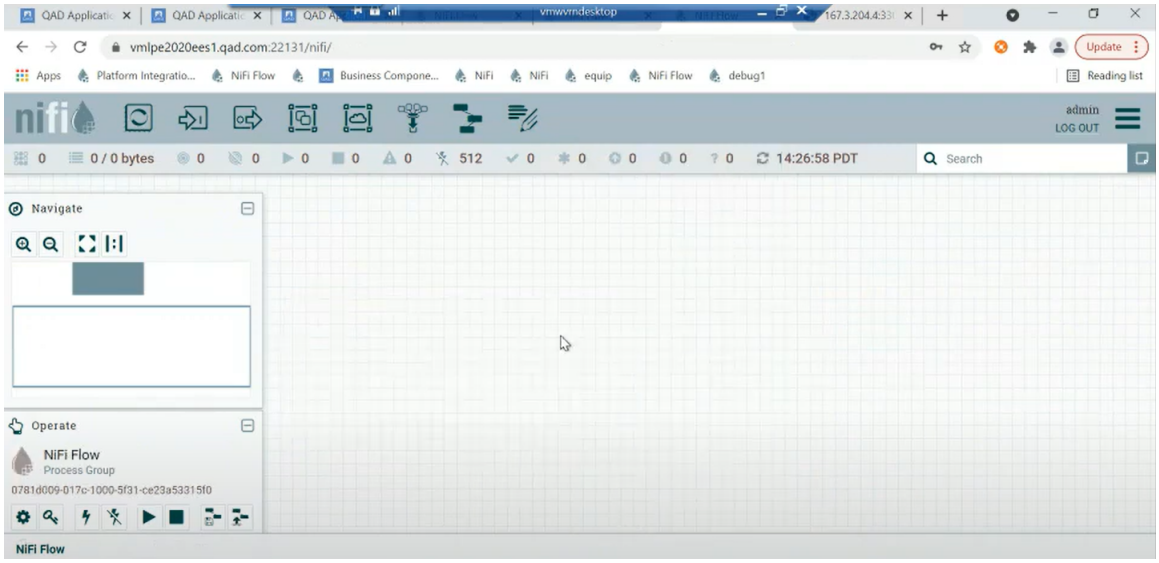
10. Navigate back to the Nifi Summary window and repeat these steps for any remaining Queues.

Deleting a Nifi Flow

11. To delete a Nifi flow, right-click on the flow and select Delete.



12. The flow is deleted.



Installing a New Version of the Nifi Flow Components

To install a new version of the Nifi Flow Components, enter the following yab command:

```
yab install penifi-x.x.x.x
```