MD: Configuration

MQTT Distributor provides a configuration section to the Ignition Gateway. These can be seen in the Configure section of the Ignition Gateway web UI in the left-hand navigation panel - Configure MQTT Distributor Settings. Once in the configuration section there are two tabs. Each has a number of configuration options as described below.

General

These are the global MQTT Server configuration parameters. For more details on enabling TLS for the MQTT Server see this tutorial: TLS Enable MQTT Distributor.

Main

- **Enabled**
  - This denotes whether or not to enable or disable the MQTT Server functionality of MQTT Distributor

Non-TLS Settings

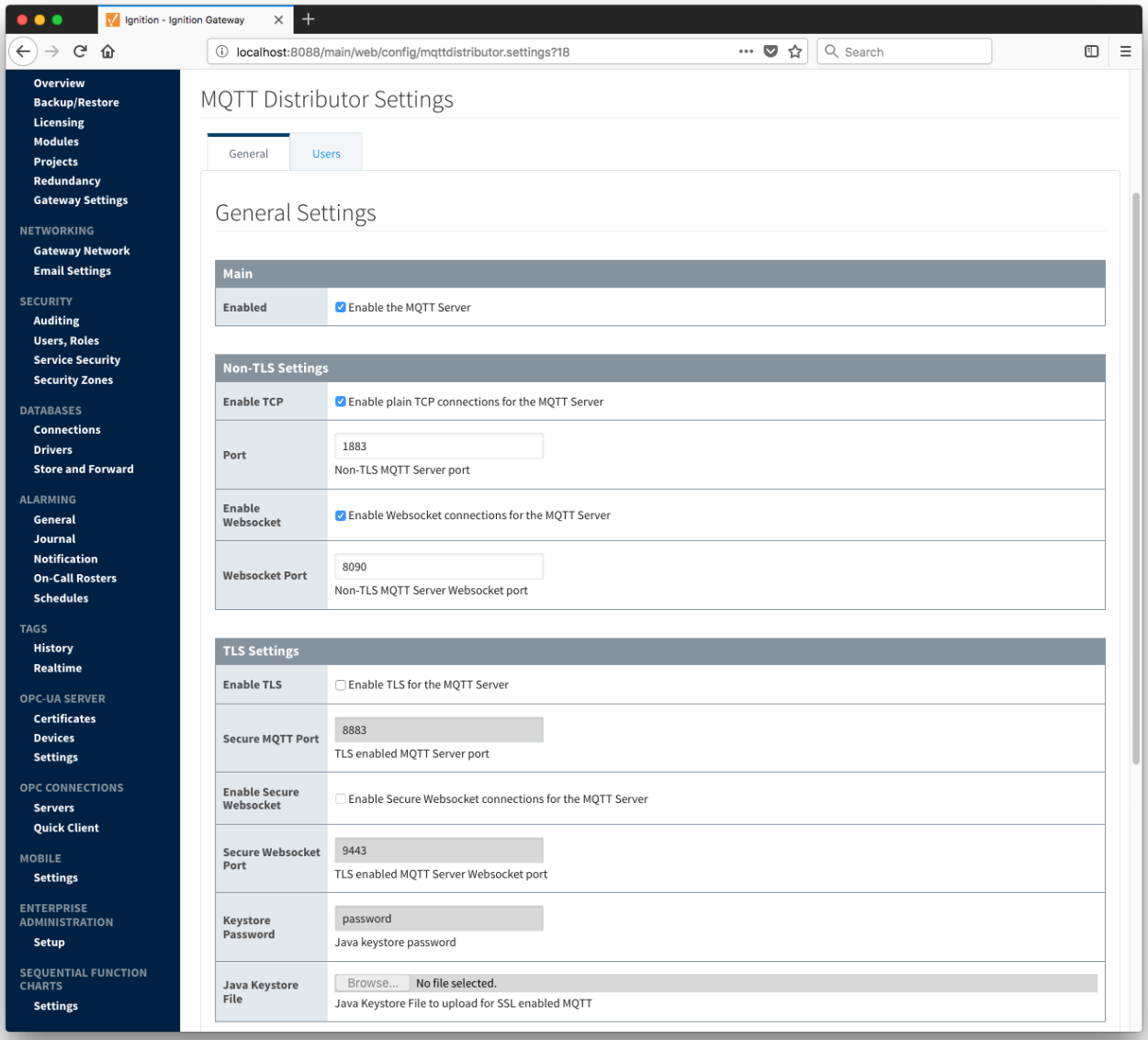
- **Enable TCP**
  - This denotes whether or not to enable plain TCP connections. This is enabled by default.
- **Port**
  - This is the standard TCP MQTT Server listening port. By default it is port 1883 and is the MQTT reserved port with IANA
- **Enable Websocket**
  - This denotes whether or not to enable plain Websocket connections. This is enabled by default.
- **Websocket Port**
  - This is the standard Websocket listening port for the MQTT Server. By default this is 8090

TLS Settings

- **Enable TLS**
  - This denotes whether or not to enable TLS connections. If TLS is used a Java Keystore file must be uploaded to secure the connection. This is not enabled by default
- **Secure MQTT Port**
  - This is the TLS enabled MQTT Server listening port if TLS is enabled. By default it is port 8883 and is a reserved port with IANA
- **Secure Websocket Port**
  - This is the TLS enabled Websocket port for the MQTT Server. By default this is port 9443
- **Keystore Password**
  - This is the Java Keystore password to use if TLS is enabled and a Java Keystore file is provided
- **Java Keystore File**
  - This is the Java Keystore file that contains the server certificate and private key files

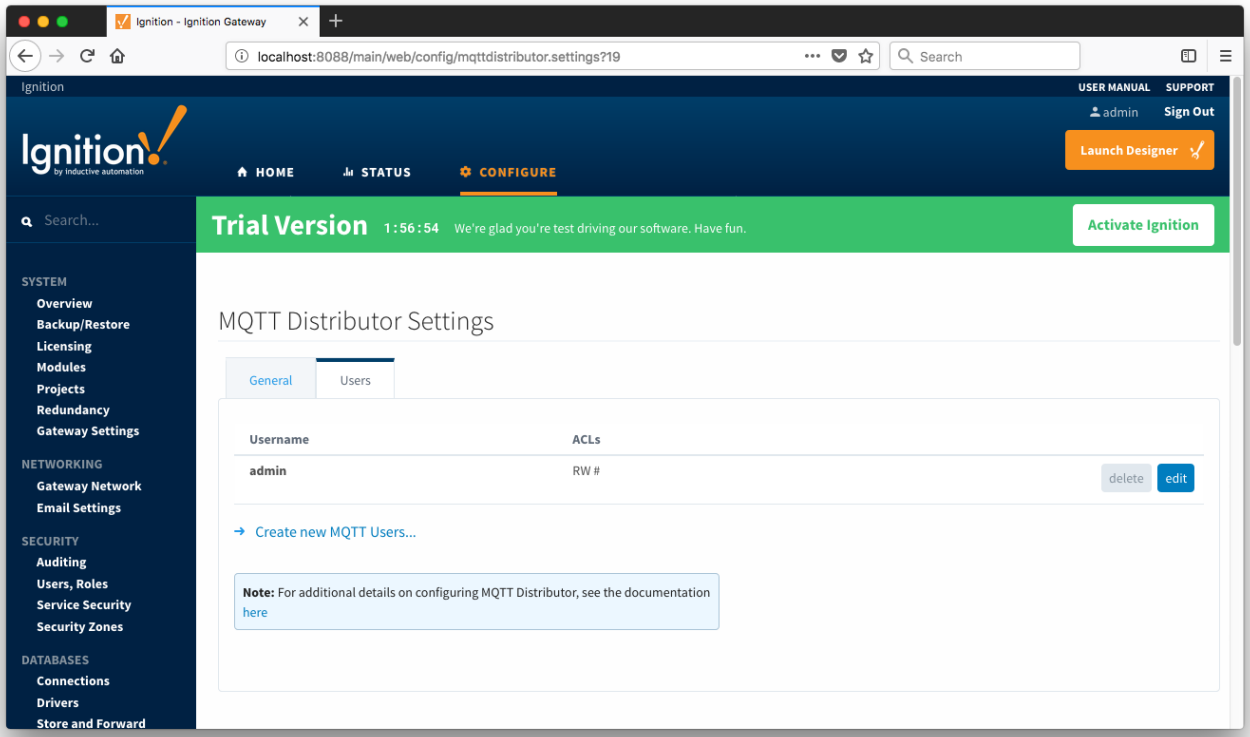
Advanced

- Allow Anonymous MQTT Connections



Users

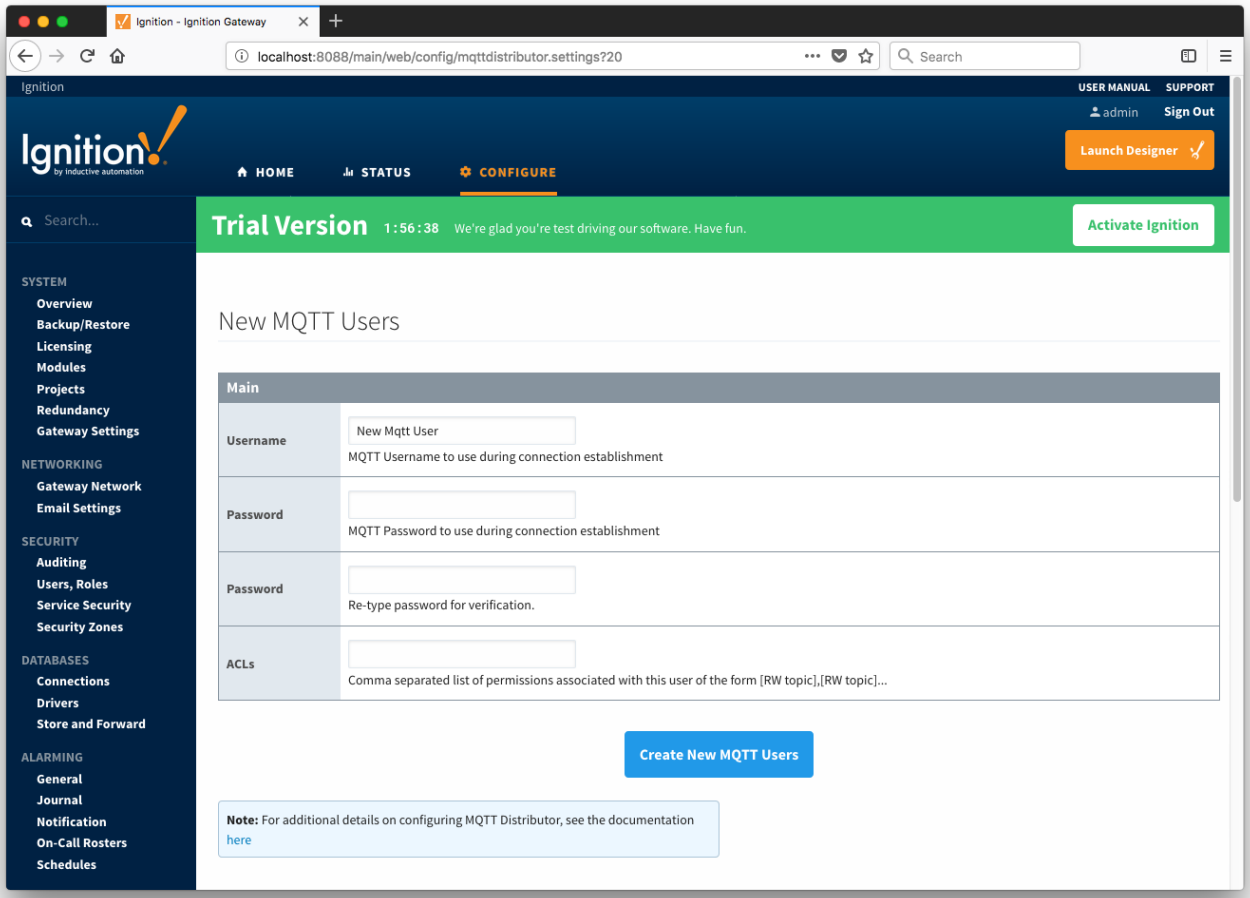
These are the username/password pairs that are allowed to connect to the MQTT Server and also contains the Access Control Lists (ACLs) for each user. MQTT Distributor requires that every client connecting to the MQTT Server must provide a valid username and password that is provisioned here. Any client attempting an anonymous connection will be rejected. ACLs control what topics a given username/password pair is allowed to publish and subscribe on. These are described later in this page.



Each user has the following configuration:

Main

- **Username**
  - The username that must be provided in the MQTT Connect packet
- **Password**
  - The password that must be provided in the MQTT Connect packet
- **ACLs**
  - The comma separated list of ACLs that clients connecting with this username and password are allowed to publish and subscribe on



ACL Format

ACLs are defined by the following format: *[R|W|RW] topic*

where:

R = Read or 'subscribe' privileges

W = Write or 'publish' privileges

RW = Read and Write (subscribe and publish) privileges

topic = The topic or wildcard topic representing the scope of the privilege

Examples:

RW #

- This allows clients connecting using this username/password to publish and subscribe on any topic

R #

- This allows clients connecting using this username/password to subscribe on any topic but not publish on any topics

W #

- This allows clients connecting using this username/password to publish on any topic but not subscribe on any topics

W device_one/temp/#,R state/#

- This allows clients connecting using this username/password to publish on device_one/temp/# and subscribe on state/# topics

ACLs should be designed with a 'principal of least privilege' model while also considering device management and maintenance. For example gateways and devices in the field should be limited to publishing and subscribing only on the topics for which they should be expected to. The same should be true of 'consumer' applications that will be either sending commands to devices in the field or consuming data coming from those devices.

It is also important to note that a username is not limited to a single MQTT client. A username/password pair could be used for multiple MQTT clients.

If you are new to MQTT topics, the Eclipse Foundation's Paho project provides good information here on the basics of wildcards.