

# Ignition 8-Based MQTT Architecture

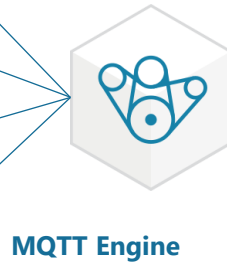
## Cloud Injectors:

Injector modules for AWS, Azure, Google and IBM



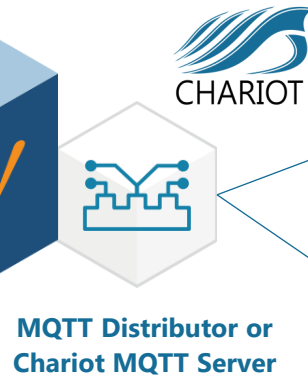
## Host Client(s):

Ignition with MQTT Engine



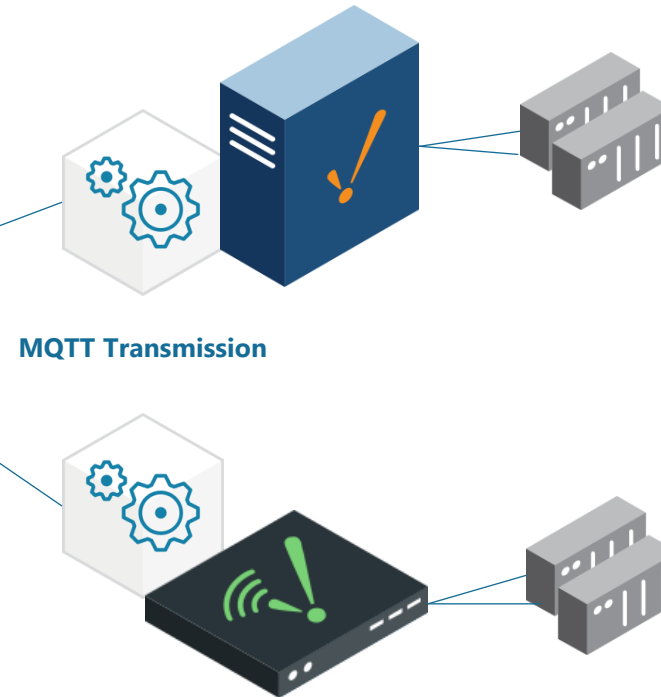
## MQTT Server:

Ignition with MQTT Distributor



## Edge Clients:

Ignition with MQTT Transmission  
and/or Ignition Edge MQTT



## Protocols

Emerson Roc – EFM

ABB Totalflow – EFM

DNP 3.0

Modbus

Allen Bradley

Siemens

Omron

BACnet

# Steps to Building an Ignition-Based MQTT Architecture

Click on the gray icons to go to links with instructions

## Cloud Injectors:

Injector modules for AWS, Azure, Google and IBM

Step 7



## Host Client(s):

Ignition with MQTT Engine

Step 1

Install central Ignition Gateway



Install MQTT Engine

Step 3

## MQTT Server:

Ignition with MQTT Distributor

Step 2

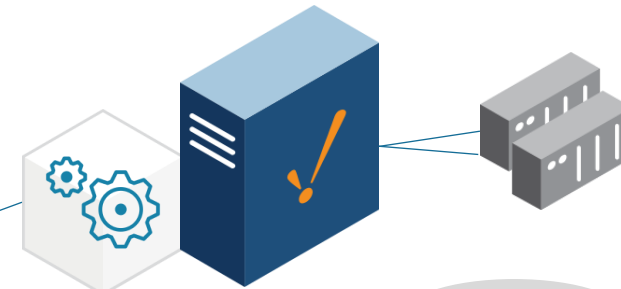


Install MQTT Distributor or Chariot MQTT Server

## Edge Clients:

Ignition with MQTT Transmission

and/or Ignition Edge MQTT



Install MQTT Transmission or Ignition Edge MQTT



Step 5

Step 4

Select edge hardware

## Protocols

Emerson Roc – EFM

Step 6

ABB Totalflow – EFM

DNP 3.0

Modbus

Allen Bradley

Siemens

Omron

BACnet

[Overview](#)

# MQTT IIoT Recipe

The Ignition based MQTT Architecture steps through the process of how to install and set up an end-to-end solution in order to gather data and send to the cloud. This architecture provides a superior OT solution using the Ignition platform by Inductive Automation.

The minimum requirements for this end-to-end solution are:

1. [Ignition Gateway platform](#)
2. [MQTT Server](#) - either MQTT Distributor running as a module on the Ignition Gateway or standalone Chariot Server
3. [Host Client](#) - MQTT Engine running as a module on the Ignition Gateway
4. [Edge hardware](#) - select appropriate edge hardware platform
5. [Edge Client](#) - Ignition Edge IIoT
6. [Protocols](#) - gather OT data from existing field equipment RTU's / PLC's using native protocols
7. [Cloud Injectors](#) - to transmit OT data to the cloud for big data analytics

# Step 1. Installing Ignition Gateway



The unlimited platform for SCADA and so much more.

Ignition Gateway is the SCADA host platform, which performs all the traditional SCADA functions as well as allowing other MQTT Modules, such as the MQTT Server and Cloud Injector modules, to be loaded.

[Quick Start Guide](#)

[Back](#)

# Step 2. Installing MQTT Server



Every MQTT solutions needs at least one MQTT Server for the MQTT Clients to connect to. There can be Edge Clients, which are typically data producers, and Host Clients, which are typically data consumers. However, they are both considered clients that connect to the MQTT Server. There are three options for the server depending on the size of the system:

[MQTT Distributor](#) – Supports up to 50 client connections and installs as a module on the Ignition Gateway.

[MQTT Distributor Plus](#) - Supports up to 250 client connections and installs as a module on the Ignition Gateway.

[Chariot](#) - Supports unlimited client connections and installs as a standalone application on a Linux or Windows server or VM.

[Click here](#) to download the MQTT Distributor module.

[Back](#)

# Step 3. Installing MQTT Engine

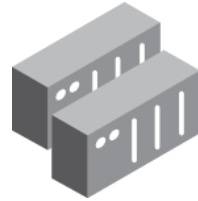


MQTT Engine is a host client that allows Ignition to connect to MQTT Servers and receive MQTT messages. MQTT Engine supports Sparkplug B which enables the auto-discovery of devices and tags with the rich tag metadata. With the correct configuration MQTT Engine can also parse non Sparkplug B messages.

- [Click here](#) to download the MQTT Engine module.
- [Click here](#) for further details on MQTT Engine, including how-to's and installation videos.

[Back](#)

# Step 4. Selecting Edge Hardware



Ignition Edge runs on most Edge Hardware platforms that support Java and have at least 1GB of free memory, this includes most Embedded edge hardware platforms or even a [Raspberry Pi](#).

[Click here](#) for several hardware manufacturers that we have worked with and loaded Ignition Edge on.

[Back](#)

# Step 5. Installing Ignition Edge IIoT Client



Ignition Edge is a condensed version of the Ignition Gateway. Ignition Edge has the same familiar UI and can be configured with MQTT Transmission module, which is an MQTT Client that takes Ignition tags and transmits them to the MQTT Server. Ignition Edge solutions come with unlimited tags and are equipped with OPC UA along with Modbus, TCP, and Siemens drivers, and the Allen-Bradley suite of drivers. Other [drivers supported by Ignition](#), such as DNP3 and Modbus RTU, can be added onto Ignition Edge solutions for an additional cost.

[Click here](#) for more details on Ignition Edge IIoT.

[Click here](#) for instructions on installing Ignition Edge IIoT.

[Back](#)



# Step 6. Installing Industrial Protocols

To connect to your device, go to the **Configure** section of the Gateway webpage and scroll down to **OPC-UA Server > Devices**. The Device page will appear showing all your installed devices. To add a new device, click on the on the blue link, **Create new Device**. Scroll through the list of devices, and click the radio button next to the type of device connection that you are interested in:

- [Emerson Roc – EFM](#)
- [ABB Totalflow – EFM](#)
- [Allen Bradley Ethernet](#)
- [Modbus](#)
- [Siemens](#)
- [UDP and TCP Driver](#)
- [DNP3](#)
- [Omron](#)
- [BACnet](#)

[Back](#)

# Step 7. Installing Cloud Injector Modules



The Cloud Injector Modules install on the primary Ignition Gateway and allow specified Ignition tags to be sent to the cloud with a Cloud Injector originated connection. Thus, no inbound ports are open on the Ignition Gateway server for this cloud connection. Data is sent using the same efficient report by exception methodology to reduce data bandwidth and cloud injection costs.

All major cloud providers are supported:

- [AWS Injector](#)
- [Azure Injector](#)
- [Google Cloud Injector](#)
- [IBM Cloud Injector](#)

[Back](#)