

Ignition 7.9 User Manual — Welcome and Getting Started

Generated 2026-10-12 from
<https://www.docs.inductiveautomation.com/docs/7.9/>

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Ignition User Manual

About this Manual

In this manual, you will find reference information along with step-by-step instructions. The final goal of the content and information provided here is to help you gain understanding of Ignition concepts and to accomplish tasks within Ignition so that you can be productive and effective while working on your own projects. You can search for specific items, or browse through the navigation tree to find the answers you need.

The table of contents is organized so you can easily find answers to your questions on all the Ignition features, modules, functions, and anything else you may be looking for. The **Getting Started** section provides a **broad overview** and information relating to **Modules**, **architectures**, **installation**, and a **quick start guide**. This core **User Manual** section is closely tied with the Inductive University so you can have a conceptual understanding from this manual and a visual walk-through of the feature in the videos. Just look for video links on the right side of the page to jump to the University. Finally, in the **Appendix**, you'll find an exhaustive list of all the **Components**, **Expressions**, and **Scripting Functions** in Ignition.

This section of the Ignition User Manual shows all of the documentation for Ignition in a format that is listed by feature or section of the platform. It provides reference and concept information along with step-by-step instructions on how to do just about everything in Ignition. Throughout this online help manual, you will find links to the various **Inductive University videos**. These videos correspond to the information in that page or section. This way, you can get a conceptual understanding from the information provided here, and a visual walk-through of the feature by watching the videos.

Take a look at the navigation section on the left and poke around, or use the search bar at the top of the navigation. To get back to this manual later, you can access it in three ways:

1. To access from a **web browser**, go to docs.inductiveautomation.com
2. To access from the **Gateway Webpage**, log into the Configure section and select **System > User Manual**.
3. To access from the **Designer**, press the **F1** key or select **Help > Help** from the menubar.

Ignition 7.9

This User Manual is specifically for Ignition 7.9. Newer major versions of the software will have separate User Manuals that focus on newer features.

You can also tell which User Manual version you're looking at by the selected version in the navigation bar at the top of the page.

Ignition User Manual

Docs

SDK Guide 

Resources ▼

Partner Links ▼

7.9 ▼

Other Manual Sections

Getting Started

The [Getting Started](#) section has all the basic information on Ignition. If you want a general overview and some basic steps to get Ignition up and running, this is the place to start.

New in this Version

Are you looking for something that has changed? Or maybe you want to see the things we have been adding to Ignition? Check out the [New in this Version](#) page for a list of improvements for each version.

Appendix

The [Appendix](#) section is a comprehensive list of all the [Components](#), [Expression Functions](#), and [Scripting Functions](#) in Ignition.

Glossary

A Glossary of common terms found in Ignition and in the industry.

SDK Documentation

Documentation for the [Ignition Module SDK](#). Anyone looking to create a new module should start here.

Partner Programs

Inductive Automation is proud to have a Partner Program that allows others to create modules to enhance Ignition's functionality. You can learn more about [Sepasoft's MES modules](#) and [Cirrus Link's MQTT modules](#) here.

Other Great Resources

Knowledge Base

In our [Knowledge Base](#), you can search and view articles created by our support team for troubleshooting, known problems, and workarounds.

Inductive University

The [Inductive University](#) is a free online learning platform designed to help you master Ignition. You can watch over 500 training videos and participate in our credential program to test your knowledge. It's the first step to joining our ranks of Certified Integrators.

Community User Forum

Our [User Forum](#) is *the* place to get help from our passionate community of Ignition users. The Forum is one of the most effective ways to get help from others; it is always available, and is actively patrolled by Inductive Automation staff and many knowledgeable users. Chances are you will find your question already answered in an existing post, but if not you can sign up and ask away.

Support Team

Can't find it here? Go to the [Support homepage](#) for one-on-one help from our support team. Submit a Ticket, and one of our Support Engineers will follow up with you quickly.

You can reach us during business hours 8am-5pm PST at 1-800-266-7798. Support charges may apply. 24-hour support is also available, at an additional fee.

E-mail support is available at support@inductiveautomation.com.

Our Founder's Blog

[Computing Without Boundaries](#) is Steve Hechtman's (Inductive Automation's President) blog about HMI and SCADA software and the industry as a whole.

More Resources on our Website

- **Ignition How-Tos:** Step by step guides
- **White Papers and Articles:** What's happening in the industry and with Inductive Automation
- **Webinars on Demand:** Watch recordings of our past live webinars
- **Tip Sheets:** Tips and references put together for you

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Getting Started

Welcome to Ignition by Inductive Automation

Ignition is not like any other HMI / SCADA system you've seen before! Ignition is a single install, runs from a single location, is server based, and is sold by the server not by the client. It uses current web-based technology and allows you to bring the IT department and the plant floor closer than ever. It uses Java for fast secure runtime clients and that means it can be used on **any system, any version**. Windows 10? Yes. Mac OSX? You bet. Linux? Bring on your Cloud servers! You can even switch between systems or use one for the server and another for your Runtimes. If IT wants to use Windows servers to host and operations wants to use their new MacBooks, it isn't a problem.

What can Ignition do for you? It has built-in, drag and drop functionality for just about anything you can imagine. Ignition does HMI/SCADA controls, Dashboards, Historical Trending, Database access, Reporting, Alarming, Security, Sequential Function Charts, Redundancy, Failover control, Enterprise Administration, and more! Ignition does all this and does it well because of its **Modular Architecture**. That means you get to choose exactly what Ignition does for you. Pick the features you want and ignore the rest. Want to try some of the other features? You always have a full version of Ignition with a resetable timer to try new things. Play around as much as you like and decide for yourself what you want.



Ignition Basics

Ignition is completely different from what you are used to using. See what Ignition is and some of the many things it can do for you. Learn the basics and how it can work for you.

Introducing Ignition

Available Modules

Modules are the building blocks of Ignition, and how it can do so much and be so flexible. Adding the functionality you need is simple with Ignition Modules.

Ignition Modules

System Architectures

Ignition can support many different types of architectures from single standalone systems to vast enterprise level structures. Learn about the types of architectures and how they work.

System Architectures

Installing and Upgrading

You can install Ignition in under 3 minutes, and staying current is just as easy. No hassles, no secret handshakes, only one lightweight install and you can have clients running everywhere.

Installing and Upgrading

Quick Start Guide

New to Ignition? Lets get started by getting Ignition running on your computer in minutes with our Quick Start Guide! Download Ignition and evaluate the full version right away.

Quick Start Guide

Features and Functions

Are you ready to **really** get to know Ignition? The user manual is chock full of reference material and step-by-step instructions to get your project set up just the way you need it.

Ignition User Manual

Ignition™, Ignition Edge™, and Inductive Automation® are trademarks of Inductive Automation. All other product or company names that may be mentioned in this publication are tradenames, trademarks, or registered trademarks of their respective owners.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Introducing Ignition

What is Ignition?



INDUCTIVE
UNIVERSITY

What is Ignition?

[Watch the video](#)

Ignition is a software platform for creating custom applications to suit your needs. It is HMI (Human Machine Interface) and SCADA (Supervisory Control And Data Acquisition), and much more! You can create anything you need with little effort and easily change your system any time you want. Ignition is designed from the ground up to be approachable and easy to get started with, while also flexible and capable of scaling up to the largest projects.

Ignition has a core focused on everything you need and nothing that you don't.

- **Server Based** - One computer controls everything in Ignition. That means you only install in one place and connect everything else to that server. No separate install for the Designer. You can have as many Clients (runtimes) and Designers as you want, no installation necessary.
- **Simple Licensing** - One license for the server gives you unlimited access. Unlimited Clients, unlimited Tags, unlimited projects, and more. Want to add more Tags or Clients? Just do it now, no need to deal with another license.
- **Web-based Deployment** - Ignition is installed, deployed, and managed using standard web technologies. Connect and change settings, update projects, create new Tags, and more all from any computer on the network.
- **Web-launched Designer and Clients** - Ignition's Designer tool and runtime Clients are launched using the web. If you have access to the network, you can launch another client. If it has a connection back to the server, you don't need to go update it manually. All changes are automatically propagated out to the Clients.
- **Rapid Development** - All our tools are designed with ease-of-use and rapid development in mind to quickly create displays, controls, charts and more. Make generalized templates and controls that will automatically update throughout the project any time one is modified. Quickly pull in existing templates from other projects or from Inductive Automation's library.

- **Built-in Security** - Security is built in to every aspect of Ignition. You can manage users, edit screen permissions down to the individual Tag or component level, and even secure certain assets in your projects.
- **Scalable Modular Architecture** - Ignition uses modules to provide its functionality. This means you can pick and choose what you want for your project, and change your mind later. Projects can be made small and expanded out indefinitely as your business grows, or you can have smaller installations connected to and interacting with other larger systems.
- **Cross Platform** - Any system, any version! Ignition is built on Java, which means you can install and run it on any modern operating system. Ignition will run on Windows, Mac OS X, and Linux, and any 32 bit or 64 bit versions. Backups of your system are even cross platform, you can load a backup from any system. Upgrading from a 32 bit server to a new 64 bit server? No problem! Moving from Windows to Linux? No problem! Just make a backup and restore it on the new computer.

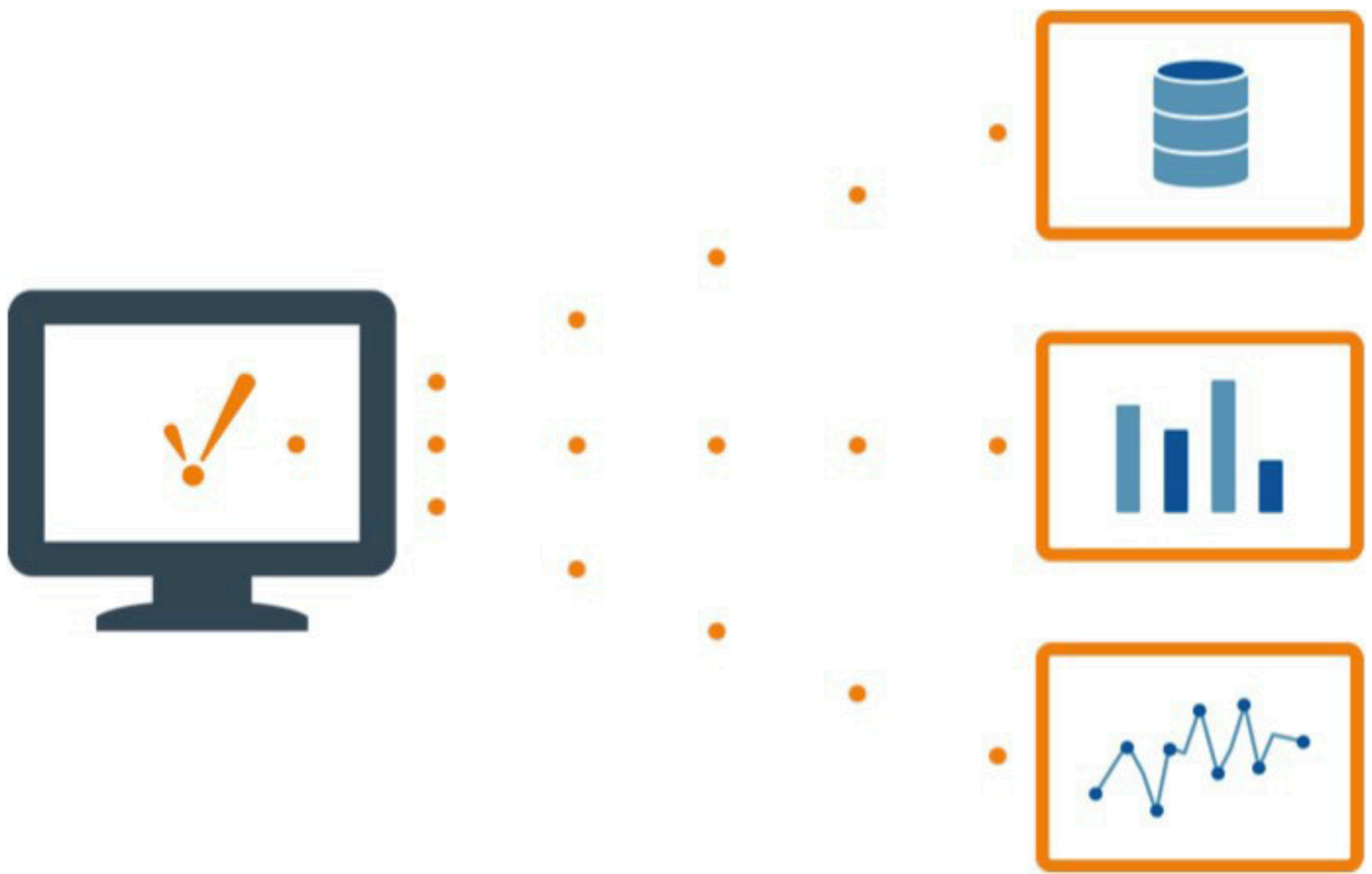
What can it do?

Because of its modular nature, Ignition can do a lot or a little and be completely customized to your needs. A lot of the basic functionality is shared between modules and stored in the Gateway to make everything as smooth and seamless as possible. Some of these basics are database connections, device connections, Tags, Security profiles, Alarming, and more. Because these are all shared resources, you don't have to worry about managing multiple connections for the different modules in Ignition: it just works!

Modules provide the core functionality for Ignition where the Gateway provides the backbone of shared resources. There are no limitations to Ignition, here are **just a few** of the things Ignition can do for you:

Web-based HMI and SCADA applications deployment

Using web technologies, you can access PLCs and connect to SQL databases on any platform. Ignition provides the common **HMI/SCADA capabilities**, but in a lot better way! Create Clients that show current or past data, interact with your other systems and devices, create displays and controls for anything wired into your system.



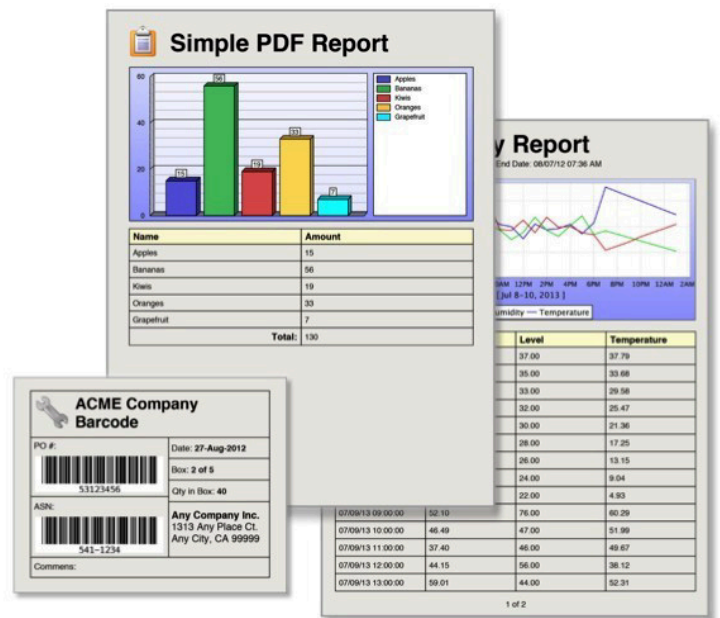
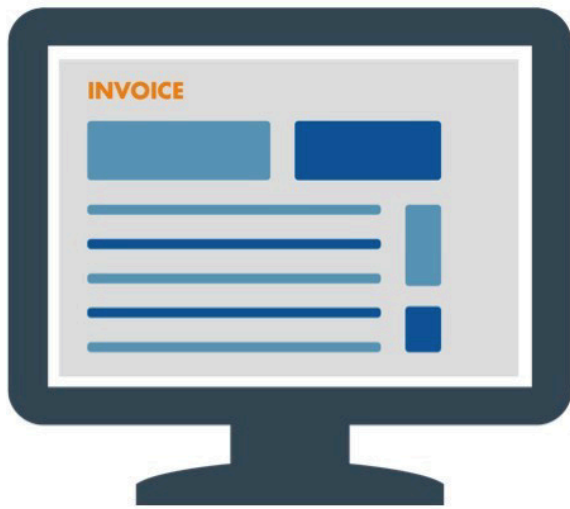
High performance historian

You can use any SQL database with Ignition's **high-performance historian**. It's simple to set up and Ignition will monitor and control the data for you. The Historian includes partitioning and uses compression algorithms for fast data storage and retrieval.



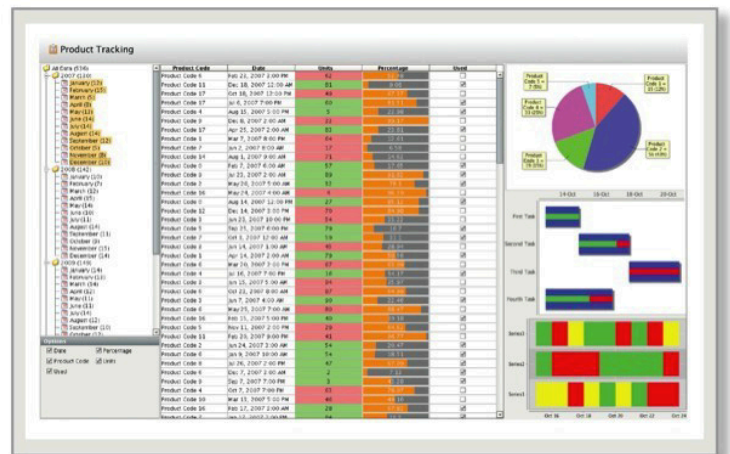
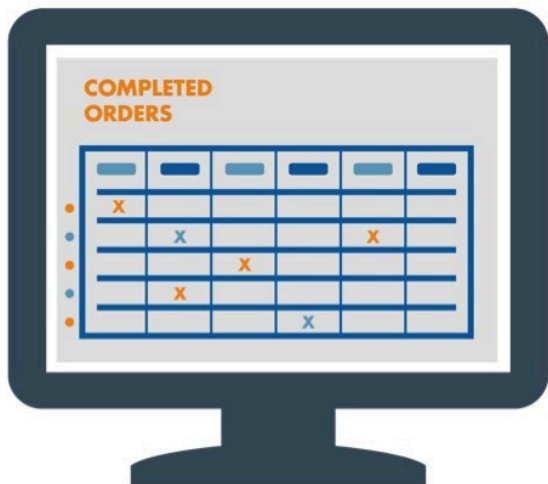
Reporting Engine

You can **create dynamic PDF reports** that users can see and change with on the fly. Give users the ability to change date ranges or select past reports without any changes to the system. Create schedules to email or save compliance reports when you need them.



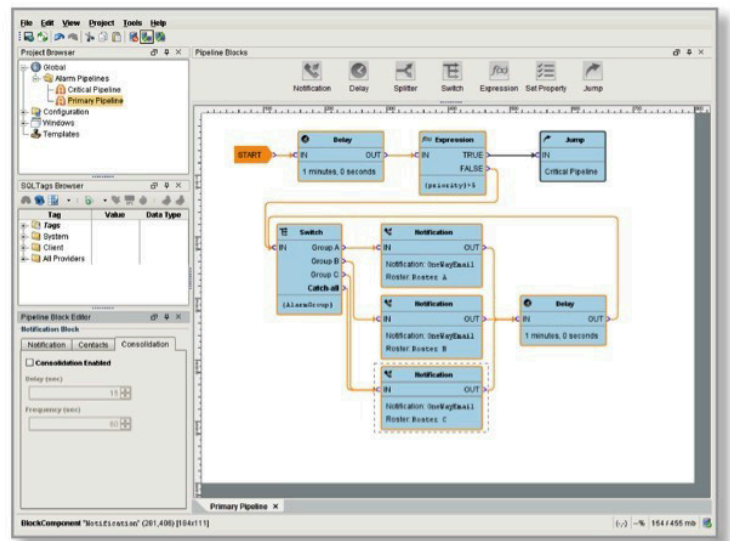
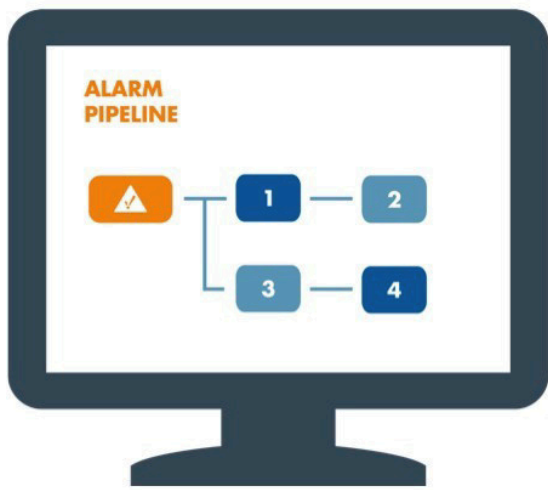
Database applications

You can build **database** applications using Ignition. You can create POS, CRM, and inventory tracking systems. You can bring in data from any SQL database in your plant and interact with other existing ERP and Access systems.



Alarming

You can use Ignition's state-of-the-art alarming engine to **create alarms** for any condition you can imagine. Alarms can be recorded, acknowledged, or dealt with by sending out notifications. Email, SMS and voice messages can be sent to individuals or whole lists of users based on any logic you need.



Communications Hub

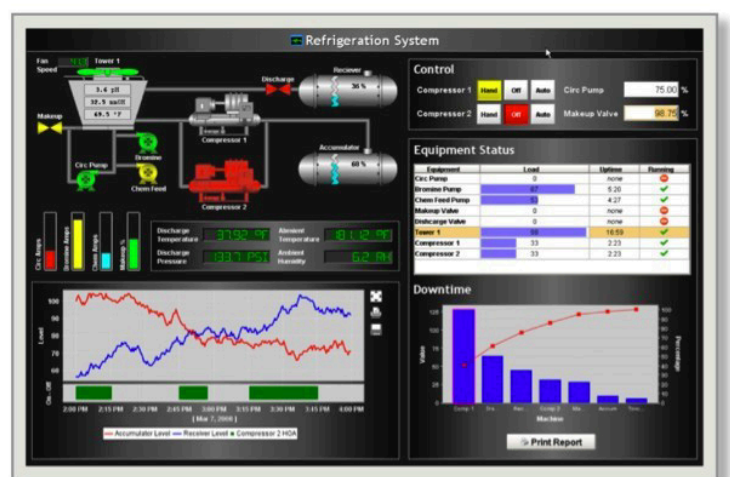
You can use Ignition as an effective communication hub in your network. You can use some or all of these features to make Ignition move data back and forth throughout your facility. Connect everything together with:

- OPC-based and communicates with virtually **any OPC server**
- **SQL-based** (JDBC) and connects to any number of Databases
- Built-in **PLC drivers** and connects to any PLC (Allen-Bradley, Siemens, Modbus, etc.) through OPC
- Supports any web services and **Launches anywhere**
- Connects with **other enterprise systems** (ERP)
- **Connects to devices** such as barcode scanners, scales, and sensors
- Connects to any **mobile device** such as wireless smartphones and tablets
- Connects to any **touch panel screen**



Mix and match

You can mix and match all of these features and more in any way you want!



Understanding Ignition's Flow

Ignition is simple to set up and use, unlike anything else you've encountered in the HMI/SCADA world before. Getting installed and running is easy, and once you have your project set up it's extremely simple to start using. After installation, setting up and using a project follows this simple set of steps:

- **Launch the Gateway Webpage-** The Gateway Webpage is where all of your information is stored, and where all of the general configuration happens. Here you can set up your Device connections, your Database Connections, alter your security, make backups, set up redundancy, and much more.
- **Launch the Designer-** Once your connections are in place, you can start using them to create a project. Opening a Designer is as simple as clicking a link in the Gateway. No installation is necessary, and you can even launch the Designer from any other computer on the network. Add Tags, query databases, show status, history trends, alarms, and more in your project.
- **Save your project** - When you save, Ignition is sending all your data and all your changes back to the Gateway. Everything is centrally stored in the gateway so you never have to worry about managing several versions or trying to combine many changes into one. You can even launch many Designers concurrently to allow multiple people to work on the same project at the same time. Projects aren't limited in scope, and you can create anything you can think of. HMI, SCADA, Reporting, Historian, Alarming, and much more.
- **Launch Clients-** Launch a Client, or 2, or 20! Clients launch the same way as the Designer: simply click a link in the Gateway, no install is necessary. All of your clients talk directly to the Gateway to get new information on the fly, so there's no need to worry about them ever getting out of sync or loading changes individually to Clients.

Ignition's modular platform can support requirements of any type by selecting the modules for your needs and building processes suited for your business. Just create your project, it's all managed in one central platform and can be launched anywhere you want.



This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Ignition Modules



INDUCTIVE
UNIVERSITY

About Ignition's Modules

[Watch the video](#)

Start Small or Large, and Grow From There

Get the functionality you need now, with the freedom to seamlessly add on to your system and match your company's ever-growing needs. The key to Ignition's flexibility is its modular architecture, which means you can buy different modules to perfectly fit your company's requirements.

Ignition is a platform that allows you to put together any system you want, and its modules are the building blocks. You can load all of them for a complete system that can do anything, or pick and choose only the ones you need right now.

What are Modules?



Modules are additional files that allow Ignition to do more. Much like adding a new tool to the workbench, adding new modules extends Ignition's capabilities while offering new ways to approach and solve problems. Most of the main features of Ignition are actually provided by different modules such as the [Vision](#) and [SQL Bridge](#) modules.

Modules integrate seamlessly into the system and provide things like new Designer workspaces, new Gateway settings, new drivers, and much more. Newly installed modules immediately integrate with other modules, so installation is quick and painless.



Why Modules?

The modules in Ignition offer many benefits.

- **Flexible Licensing** - Only license the modules that you need, saving money and reducing complexity compared to big monolithic applications that try to do everything. At the same time, the modules have been designed to offer a broad swath of functionality, to avoid having too many pieces.
- **Hot-Swappable** - Modules can be dynamically loaded and unloaded, allowing you to install, remove, and upgrade them without affecting other parts of the system. This can have huge implications for big projects where up-time is important.
- **Increase System Stability** - Building modules on a common platform means fewer bugs, better isolation, and all around increased stability.
- **Pluggable Module Architecture** - The pluggable module architecture allows quick integration of new modules into the Ignition platform. From time to time new modules will be released which add additional features. Third-party modules that provide a wide range of functionality are also available.

What Can Modules Do?

Modules are what drive Ignition's functionality and allow for unlimited possibilities. Here are some common examples of what modules can do:

- **Create Screens for all Occasions** - Dashboards, detail windows, historical displays, and more can be created with the Vision module. Read real-time values from any OPC-UA server with the OPC-UA module, and display the data for all users to see. Interactive screens with user-driven forms can also be created, so users can easily write back to a PLC, store data in a database, and much more!
- **SCADA on the Go** - Ignition applications can be viewed on mobile devices with the Mobile module. Travel through the plant floor with quick access to all your applications. Break free from attached panels and monitors by viewing projects from tablets, phones, or any other mobile device.
- **Lightning-Speed Alarm Detection** - Long gone are the days when an operator must monitor every aspect of the system to look for problems. Ignition can generate alarms when certain conditions are met, such as when temperature readings reach a certain point, or communication

with a device has dropped out. Alarms can be easily displayed with pre-built components, and alarm events can be stored into any SQL database. Additionally, with the Notification modules, Ignition can generate messages when an alarm triggers. Send emails, SMS texts, or phone calls to operators once an issue is detected. Users will always know as soon as a problem occurs.

- **Easy Data Collection** - With either the SQL Bridge or Tag Historian module, Ignition makes data logging a breeze! History collection can be enabled in just a few steps, and the data can be feed into other parts of Ignition, such as charts, tables, and even reports! All history is stored in a SQL database, so there is no need to worry about proprietary systems locking away the data. History can easily be accessed by third-party software such as ERP or SAP systems.
- **Dynamic Reports** - Instead of printing individual tables or charts, the Reporting module creates powerful, visually stunning, reports on-demand. Historical data recorded by the Tag Historian or SQL Bridge module can be inserted into a report, but data from any SQL database can easily be visualized and presented in a report. Reports can also be scheduled to run automatically, and be delivered in a number of ways such as emails, or saving to a network drive.

Third-Party Modules

In addition to the modules provided by Inductive Automation, several third-party companies have developed their own modules using our Module Software Development Kit. These modules add even more possibilities and functionality to help you design any project.

- **MES** - [Sepasoft](#) offers a suite of **Manufacturing Execution Systems (MES) modules** that allow you to track OEE, implement a SPC system, track resources, and much more! The core purpose of these modules is to spot deficiencies in your process, and increase profitability by detecting problem areas, or catching problems before they become an issue.
- **MQTT** - Modules from [Cirrus Link](#) bring the power of **MQTT** into Ignition. These modules expose massive amounts of data-points as tags, and allow Ignition access to data that would otherwise be lost. Modules from our Strategic Partners can be found on our [downloads](#) page.

How Can I Create My Own Modules?

Want to help make Ignition better? Check out our [SDK documentation](#) where you will find all of the resources necessary to develop your own module. Modules can be as simple as adding scripting functions or components, or as complex as creating whole new systems for Ignition.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

About Ignition's Modular Architecture



INDUCTIVE
UNIVERSITY

About Ignition's Modular
Architecture

[Watch the video](#)

Ignition platform has a modular architecture. Modules are software applications that are built and integrated into the platform to offer additional functionality. The modules are similar to applications for a smartphone in how they are seamlessly integrated and provide additional capabilities.

Most of the main features of Ignition are actually provided by different modules such as the **Vision** and **SQL Bridge** modules.

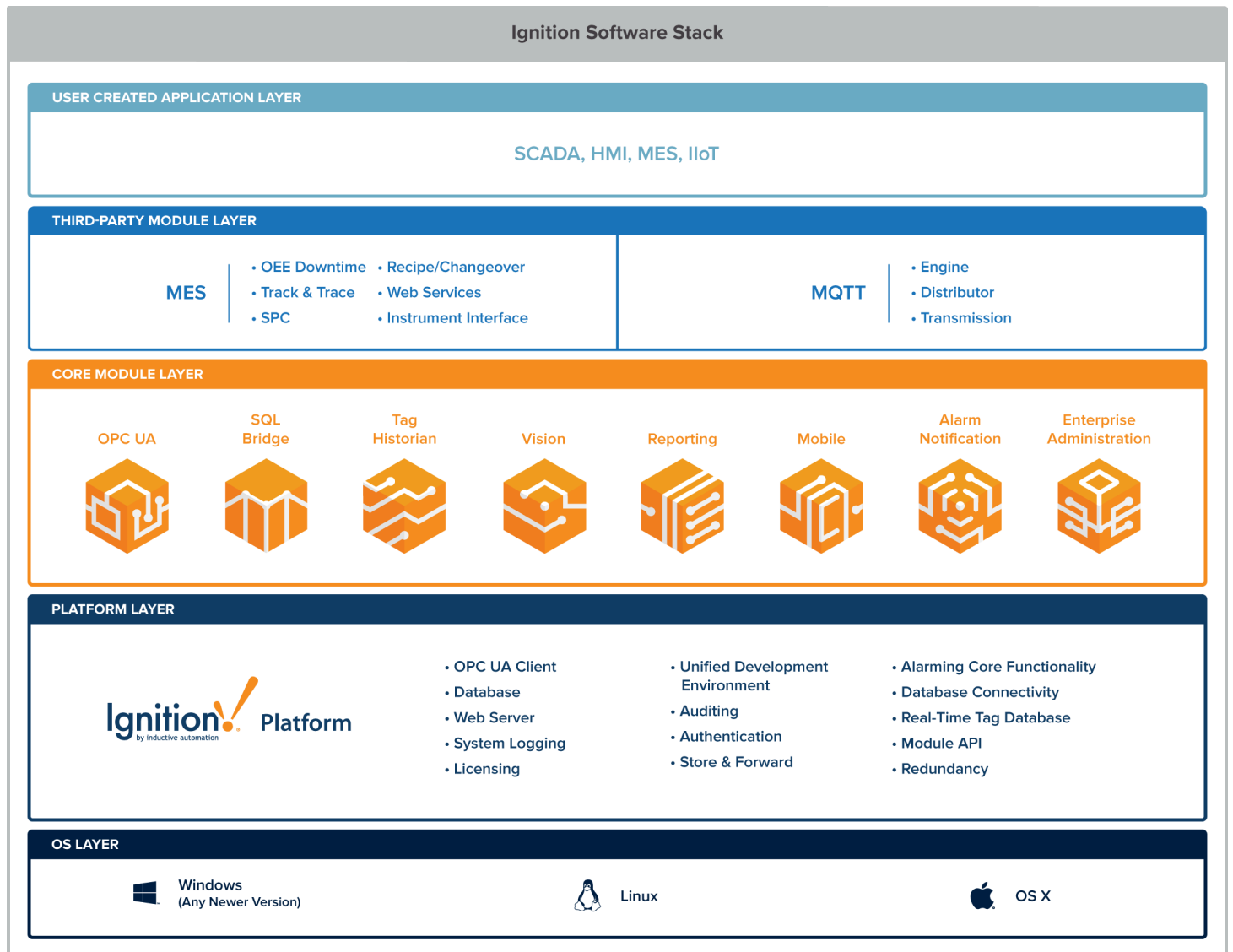
Ignition Software Stack

Looking at the Ignition software stack below, you can see that the HMI/SCADA/MES module layers are built on the Ignition platform.

Here are the different software layers in Ignition's modular architecture:

- **OS Layer – the Operating System Layer** - Provides basic computing resources such as the file system and access to the network.
- **Platform Layer** - Provides all the basic functionalities such as connecting to devices and databases, licensing, launching clients, managing all Ignition modules over the web, and more.
- **HMI/SCADA Module Layer** - Provides the core modules that enable real-time and historical data access, trends, and control
- **Third-Party Module Layer** - Additional modules provided by Strategic Partners and other developers to further extend Ignition's capabilities.

- **User Created Application Layer** - The resulting project created for your organization. Developed internally, or by a third-party.



Steve Hechtman Blog

Click the link below to see the blog:

Why your SCADA system should be more like your smart phone

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Core Modules

Build any kind of HMI (human machine interface) or SCADA (supervisory control and data acquisition) system using Ignition's powerful offering of HMI/SCADA modules.

What Are Core Modules?

Core Modules are commonly used modules included in a typical installation. Most systems have at least one of these modules installed. Each core module encapsulates a core SCADA or HMI principle such as data-logging or process visualization. These modules are the foundation upon which great projects are built.

SQL Bridge



Bridge the gap between OPC data and SQL databases and get a highly efficient SQL-based Tag historian and transaction manager. Easily log data, call stored procedures, and synchronize data bi-directionally.

Vision



Use this module to launch an unlimited number of powerful Ignition Vision clients anywhere on the network. Display charts, graphs, trends, HMI screens, and more.

Tag Historian



Turn a SQL database into a high-performance time-series Tag historian.

Reporting



Create dynamic, database-driven PDF reports with pixel-perfect layout - with ease.

OPC-UA



Connect to most major PLCs (Programmable Logic Controllers) with the Ignition OPC-UA Module. This module serves as a totally cross-platform OPC-UA server with an open, pluggable driver system.

Mobile



Get mobile access to your control system via smartphones and tablets.

Alarm Notification



Configure the logic for how, why and when alarm notifications are delivered, manage alarm notification for groups of users, and send notifications via email.

SMS Notification



Add text alerts with the SMS Notification Module for even more options.

Sequential Function Charts



Streamline the creation, organization, and visualization of robust logic systems.

Enterprise Administration



The EAM provides a secure and intuitive way to manage many Ignition installations from one location.

Symbol Factory



Vector graphics clipart library with an emphasis on industrial symbols.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

SQL Bridge Module

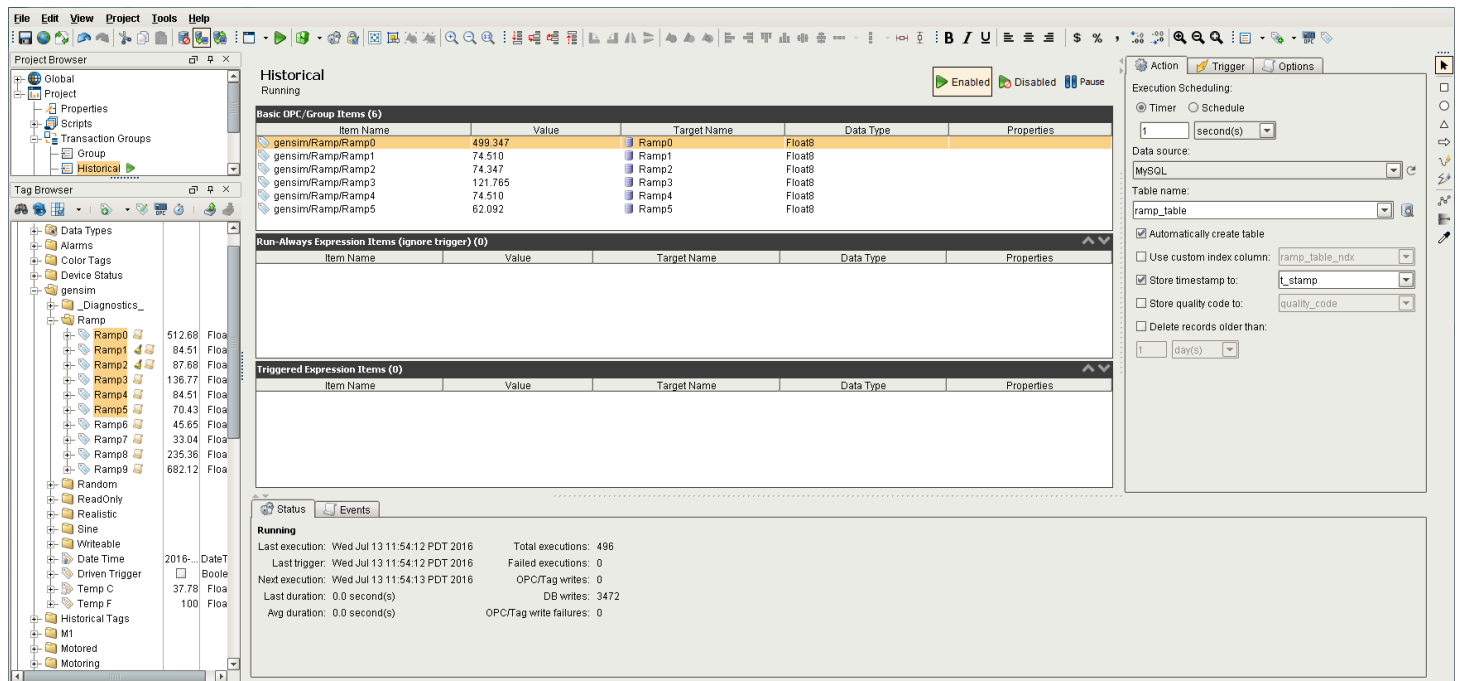


High-Powered Data Acquisition

Share Plant Floor Data With Any Database

The Ignition SQL Bridge Module brings the power of structured query language (SQL) databases to your industrial process. The module allows you to leverage Ignition's unlimited Tags and database connections at an astonishingly low price — even if you don't know much about SQL databases.

Think of the SQL Bridge Module as a "Swiss Army Knife" for integrating programmable logic controllers (PLCs) and SQL databases in unlimited ways. Move data bidirectionally, log huge amounts of data easily, synchronize PLCs with databases, create powerful recipe and batching systems, sequence entire processes, and more — your imagination is the limit.



Move Data Bidirectionally

The module uses transaction groups that function as a bidirectional data bridge between an OPC server and a SQL database. Transaction groups can perform many data-centric tasks, such as event logging, vertical and horizontal data storage, inserting rows into a table, storing data to a specific row, and more.

Log Data Easily

The easiest, most powerful data logger on the market includes triggers, handshakes, expressions, and other items that help you store data in the exact way you need. Triggers offer endless data-storage possibilities: storing production summaries, logging the sequence of event information, tracking downtime events, and more. Simply map your process data to any table structure and let the module do the work of creating and managing new tables or writing to existing tables.

Synchronize PLCs with Databases

Extend your PLC address space into your SQL database with bidirectional mirroring, which immediately reflects any value change in the PLC or database end in the opposite end. Bidirectional mirroring enables real-time control via common web technologies and other database-compatible front-ends. Set the update rate for rows/fields (down to sub-second rates), and create two-dimensional arrays by adding rows. The SQL Bridge Module is the first and only software in the industry to offer this functionality.

Build Recipe & Batching Systems

By extending your process into the database, the full bidirectional access of the SQL Bridge Module enables you to create a robust, customized recipe and batching system. Use the database to store a virtually unlimited number of recipes, and modify recipes however and wherever you want, while tracking all changes historically. Set up reliable database-to-PLC transactions using triggers and handshakes. Leverage stored procedures to perform complex tasks or replace aging transaction systems.

Sequence Your Processes

Flexible triggers and database tables enable you to easily sequence entire processes. Use dynamic references to step the index to a row of values that are then written to the PLC. Create complex triggers, which reference multiple process variables and expressions, by using a simple expression language. Use rows/fields as timers and counters by leveraging the unique built-in hour meter and event meter functionalities.

Flexible and Easy to Use

Across the board, SQL Bridge is the clear winner for tying industrial systems into SQL databases. Online editing lets you make changes on the fly, without stopping your process or affecting other parts of Ignition. You can stop locking data away in proprietary boxes because all data is open and accessible in the database. Use flexible data modeling without a lot of work: the module automatically creates tables, columns, and more, in any format. The built-in table browser allows you to work entirely within the Ignition Designer, so there is no need for external programs.

Powered by Ignition

The SQL Bridge Module is a SCADA module for Ignition, the powerful HMI, SCADA, MES and MQTT software from Inductive Automation. It integrates seamlessly with Ignition and all other modules in the Ignition SCADA suite. Because the module is built on the power of Ignition, it shares the same advantages, such as cross-platform compatibility, unlimited free clients, robust out-of-the-box SQL database support, and fast installation. Leveraging the full power of Ignition, it is a data management tool unlike any other in the SCADA software market.

Looking for No-Frills Data Logging?

SQL Bridge Limited provides an affordable way to create **Historical Groups**, and record history into any SQL database. Speak to our sales department for more details.

Want to Learn More?

For more information on the SQL Bridge Module, please see the [SQL Bridge \(Transaction Groups\)](#) section of the manual.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Vision Module

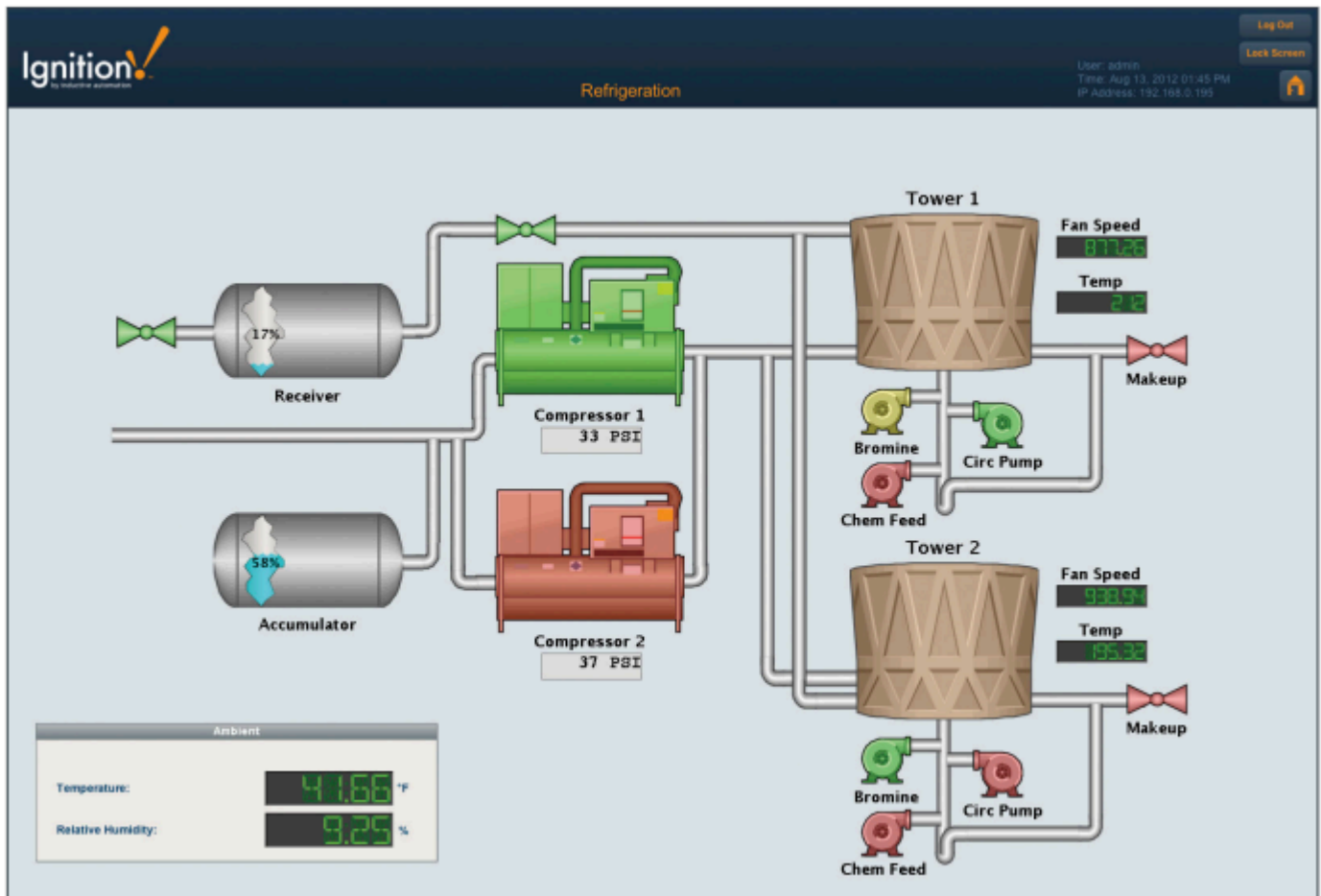


HMI/SCADA Visualization

Get Real-Time & Historical Data on Control Screens

The Vision Module delivers a dynamic, powerful, and beautiful HMI experience unlike any other. Display real-time and historical data on everything from status and control screens to interactive charts and tables so you can analyze data at a glance and quickly act on it. Since all clients are connected to a central server, everyone in your company will stay up to date and can easily access the real-time data they need.

You'll never be limited by how many Clients (runtimes) or screens you can use because the Ignition Vision Module includes unlimited clients per central server. Every member of your company can have insight into operations at no additional cost.



Real-Time Status & Control

Ignition's real-time status and control functionality, coupled with advanced analytic capabilities provided by Ignition's SQL database integration, gives you the power to quickly analyze and easily display the status of your facility.

Alarming

Set up screens to display real-time alarms from the Alarming module so your operations team can respond quickly to resolve the issue, either in the facility or out in the field via the Ignition Mobile Module.

Analyze Real-Time and Historical Data

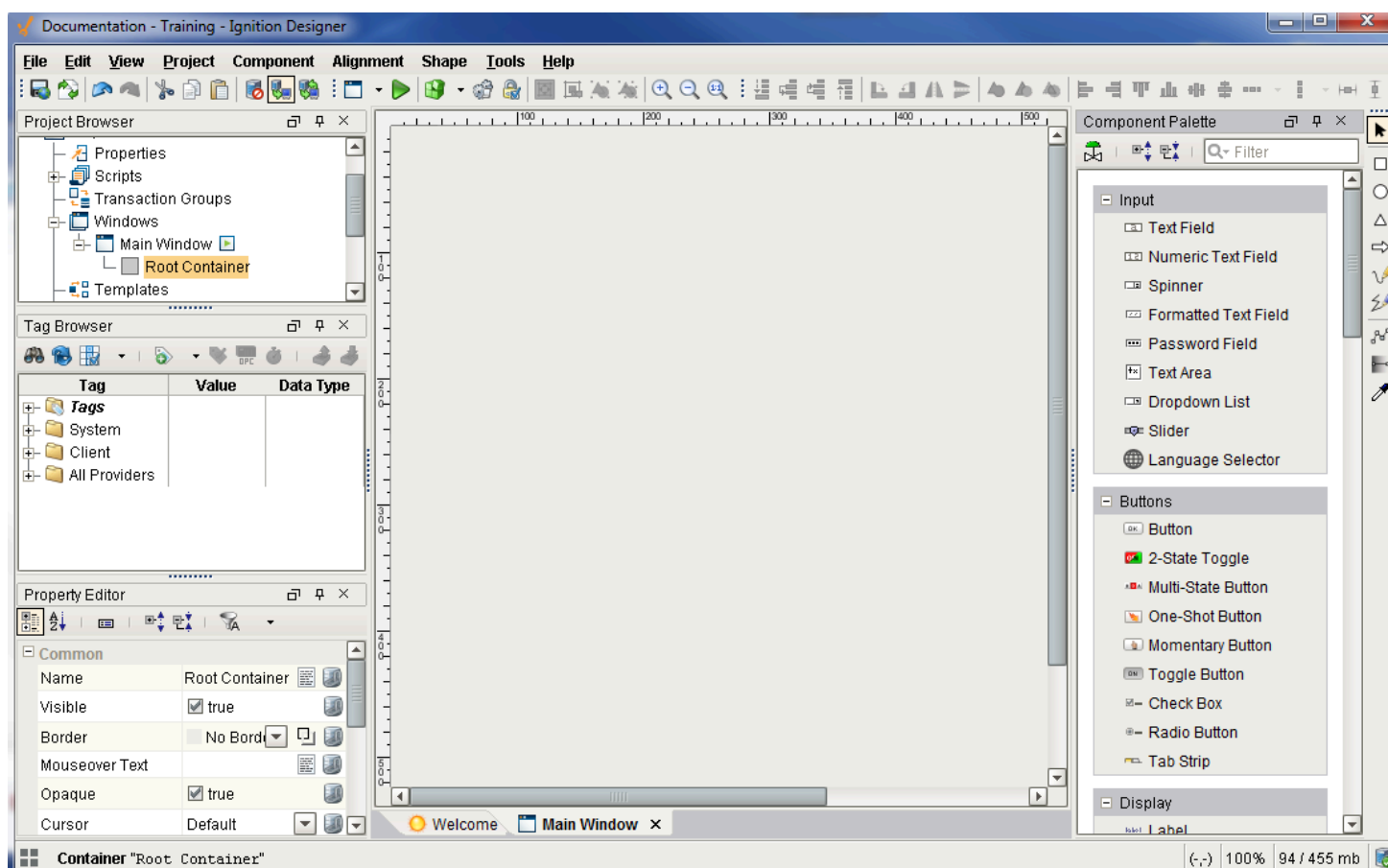
Included customizable charts, tables and graphs display your real-time and historical data, making at a glance analysis easy. You'll be able to quickly set up executive dashboards, track key performance indicators (KPIs), and view trends.

Database Front-End

Easily integrate data from SQL databases with your SCADA system. Data entered by operators can be written directly to any database. The powerful property binding system makes it easy to display information from databases on any screen.

Designer Included

Clients are configured through the included Ignition Designer. Drag-and-drop configuration makes screen development intuitive; simply drag Tags and drop them onto components to instantly bind data to inputs, displays, tables, charts, and more.



Vector-Based 2D Drawing Tools

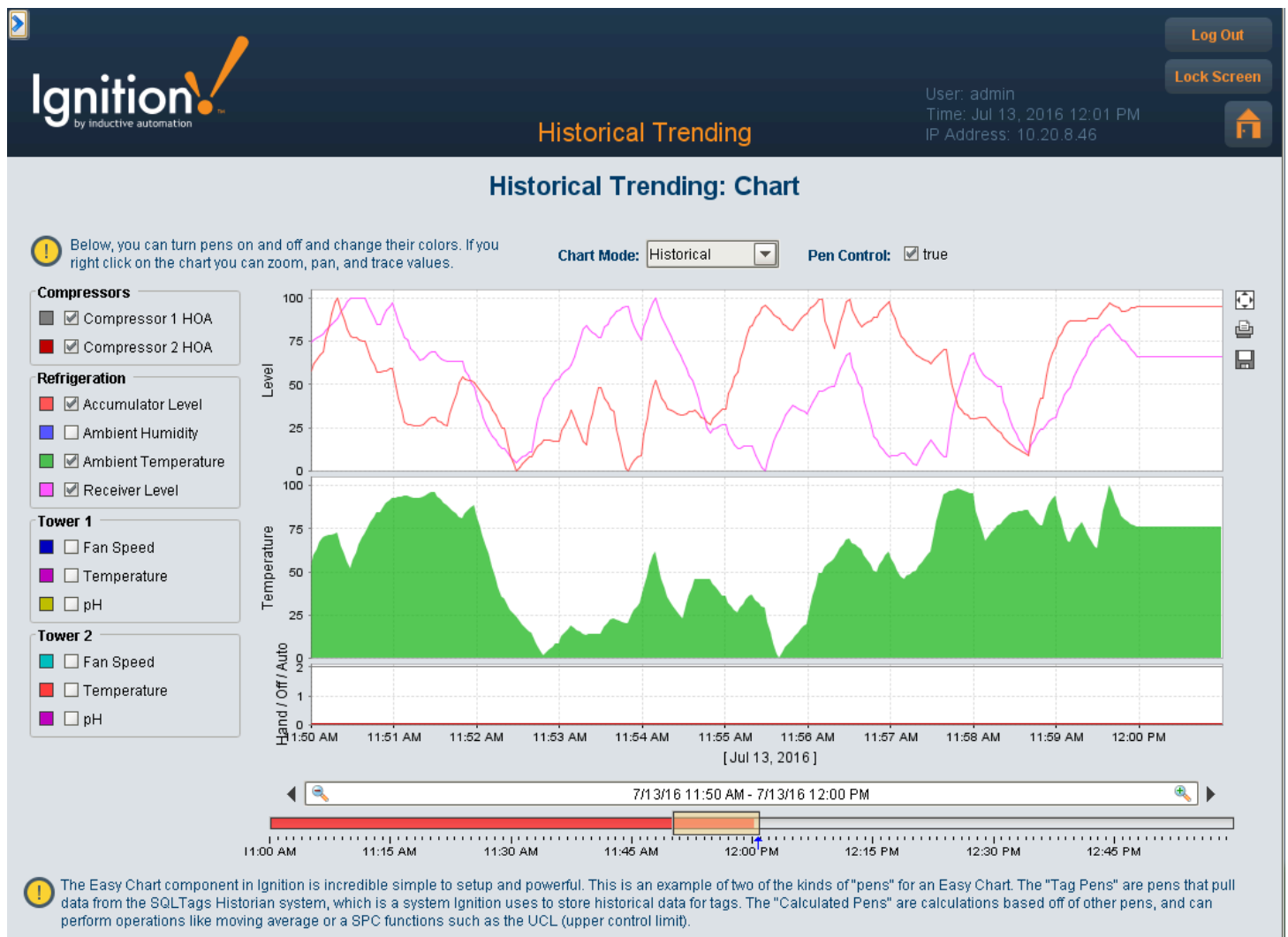
Create distinctive screen designs and layouts with shape-building tools for lines, polygons, rectangles, arrows, and circles. Vector graphics let you create precision shapes and Bézier curves that look great on any size monitor; shapes scale to any size without losing image quality.

Easily Create Animations

State dependant settings and dynamic positioning lets you bind shape properties to data values so you can easily create animations.

Speed Development With Templates

Develop large projects at record speeds by using component templates. Instances of each template can be placed on as many screens as you want. Any changes made to the template are automatically updated to any instance of that component, saving a huge amount of tedious development time.



Python Scripting

Advanced functionality can be achieved using the built-in Python scripting engine. Python is an extremely popular, powerful, and easy to learn scripting language. Python is highly regarded for its readability, making your projects easy to maintain.

Secured Via SSL

Every client you launch from the Vision Module is secured using SSL protocols. In order to keep your data secure, authentication groups can be used to strictly limit access to certain areas of a project to only the users with appropriate access levels.

Unlimited Clients & Instant Deployment

With the Ignition Vision Module installed on your server, you can launch as many clients as you want, instantly. No individual client install required!

Update All Screens With One Click

With Ignition, managing five clients is just as easy as managing 50. All clients can be updated instantly by making changes to the master project screen on your server, and then clicking the publish button.

View Anywhere, Any Time

View your HMI / SCADA system from anywhere, any time, on any machine. Ignition uses your computer's web browser to launch rich Java applications – not thin web pages – which are fast and cross-platform, making accessibility easy for anyone on your network.

Looking for Less?

Some systems only require a handful of clients at any given time. In cases like these the **limited version** of Vision is a perfect fit: all the functionality of the full version but at a greatly reduced cost! Contact our sales department for more details.

Want to Learn More?

For more information on the Vision module, please see the [Visualization and Dashboards](#) section.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

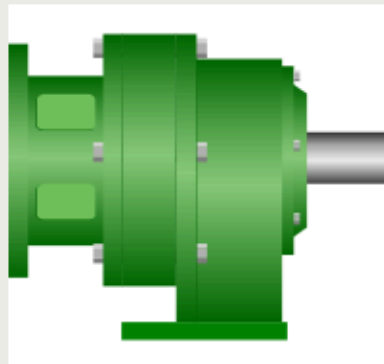
For documentation on the latest stable release, see the [latest version](#) (8.3).

Symbol Factory Module



Jump-Start Your Project with Industrial Vector Graphics.

The Symbol Factory module integrates Symbol Factory 2.5 from Software Toolbox into the Ignition Designer. The intuitive interface allows you to browse and search through nearly 4,000 high quality vector graphics symbols. Once you find the symbol you're looking for, simply drag-and-drop it onto a Vision window to use it right away.



Speed reducer

- Food
- HVAC
- Material Handling
- Motors**
- People
- Pipes
- Process Cooling
- Process Heating
- Pumps
- Renewable Energy
- Scales
- Segmented Pipes
- Sensors
- Tanks
- Valves
- Water & Wastewater

☐ Basic ☒ Enhanced

small / **large**

Search



This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Reporting Module

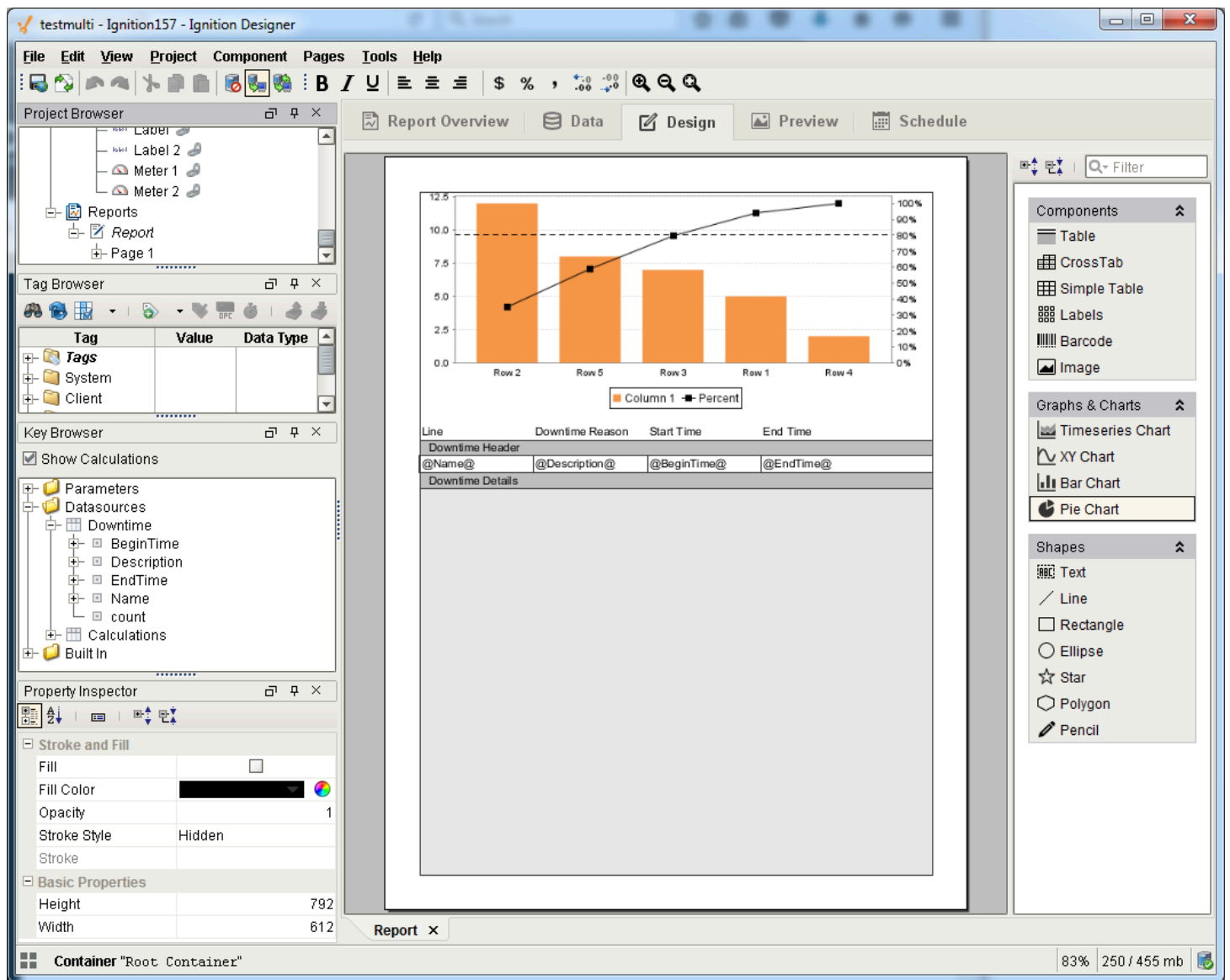


Dynamic Report Generation

Quickly and Easily Create Rich, Dynamic PDF Reports

Now it's easier than ever to make and distribute dynamic, professional reports in Ignition. The Ignition Reporting Module is a standalone reporting solution that simplifies and enhances the reporting process from beginning to end. Generate reports from existing files or create them from scratch. Pull data from a variety of sources, and design reports with a simple to use interface. Add charts, graphs, tables, crosstabs, shapes, and other components for visual impact. Save your reports in PDF, HTML, CSV, and RTF file formats, and set up automatic scheduling and delivery.

Use all of these powerful capabilities to quickly create many common types of reports such as production management, efficiency monitoring, downtime tracking, SPC, QA, OEE management, and historical data analysis.



Select Data from Multiple Sources

The Reporting Module makes it easy to define the data behind your reports. Specify any number of report parameters and data sources in your reports.

Access information from any SQL database. Also, you can also easily query data from Ignition's Tag historian and event data from the alarm journal.

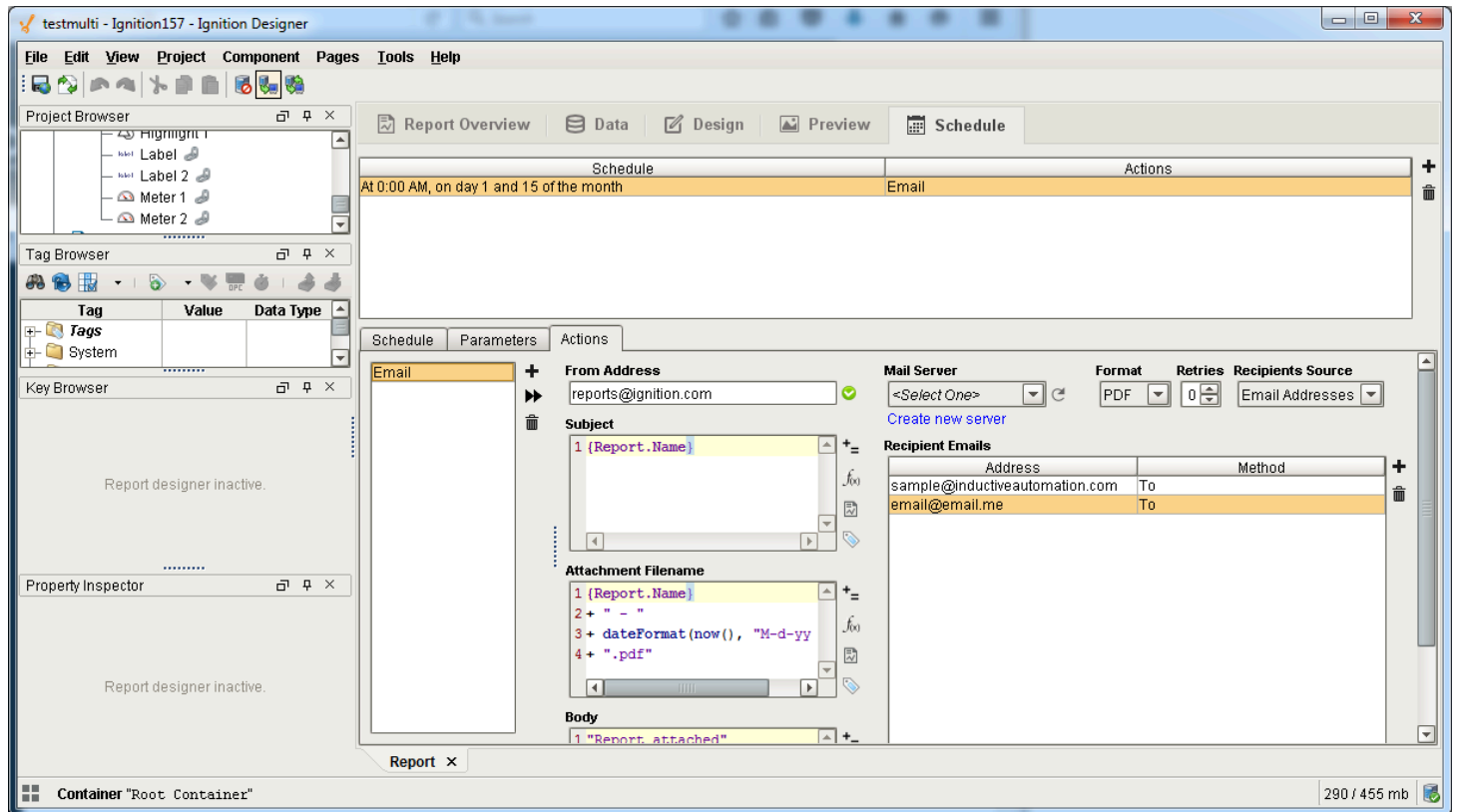
Design Reports with an Easy-to-Use Interface

Design your reports easily inside Ignition's Designer application. The interface is far more user friendly than those of other reporting software, because it enables you to design reports with the same intuitive feel and rapid application development tools that you get when designing applications in Ignition. This allows you to design your reports using the drag-and-drop method.

Schedule Reports to Run Automatically

The Reporting Module allows you to schedule reports within the Ignition Gateway. You can schedule a report to run on a regular basis; for example, on every Monday at 5:00 p.m. Then, your report will run as scheduled whether or not any Ignition clients are open.

You also have the option to set up triggered execution, which means that reports run on-demand when triggered by an event such as a Tag change, a shift change, the finish of a production run, etc.

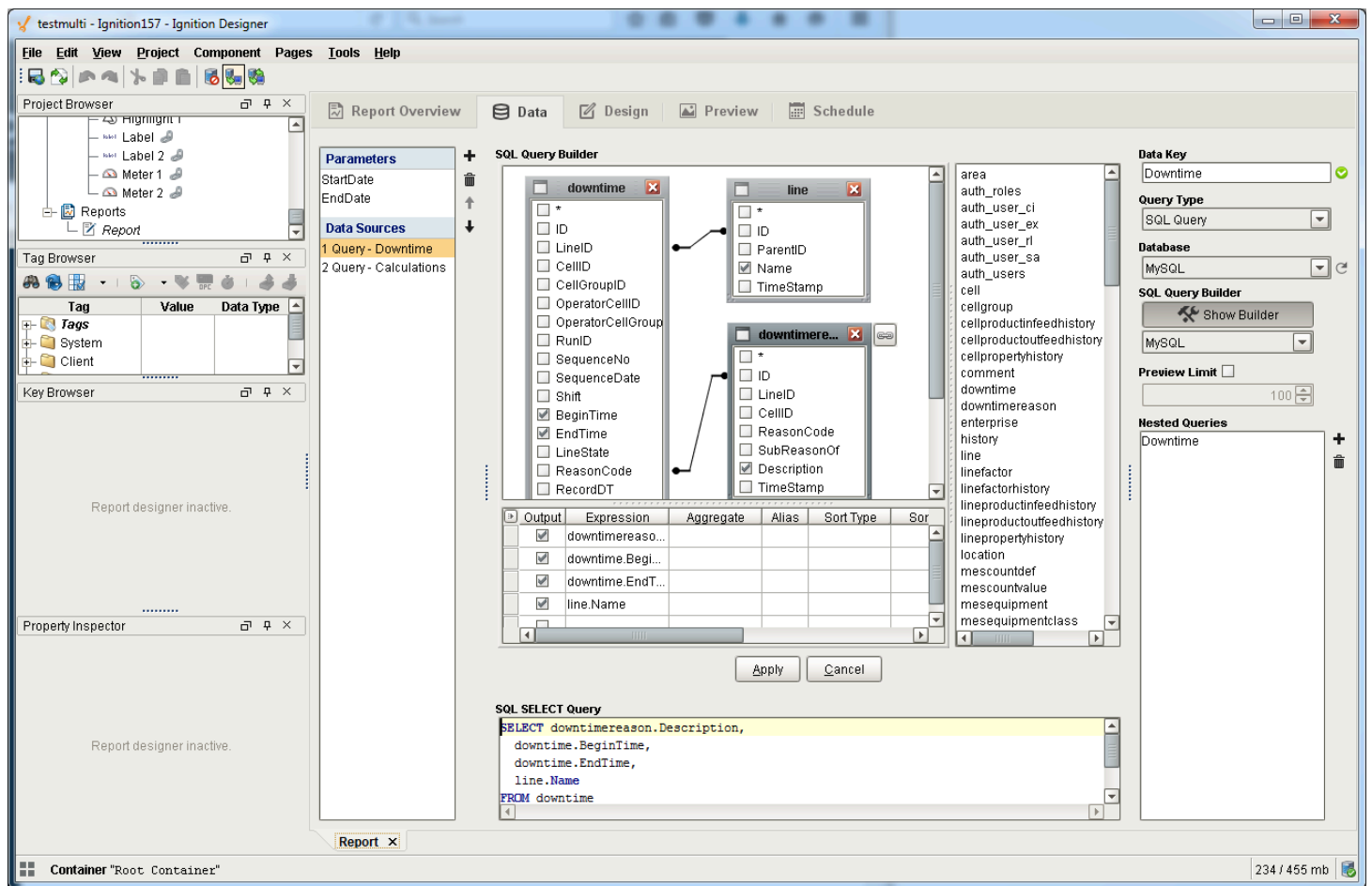


Print, Send, or Save Automatically

Scheduled reports can be automatically delivered in many ways: print, email, save to a file server, upload to an FTP server, or use scripting to customize the method of delivery. Have your reports distributed in the way you want, every time, automatically.

Powerful Query Tools

The Reporting Module has a powerful drag-and-drop query interface that allows you to build complex SQL queries easily. The new query structure allows you to create complex nesting which can correlate relational data and process historian data in a seamless fashion.



Powered by Ignition

The Reporting Module is a SCADA module for Ignition, the powerful software from Inductive Automation. The Reporting Module integrates seamlessly with Ignition and with all other Ignition modules. Because the module is built upon the power of Ignition, it shares the same advantages, such as cross-platform compatibility, unlimited free clients, robust SQL-database support, and fast installation. Leveraging the full power of Ignition, the Reporting Module is unmatched by any other SCADA reporting product on the market.

Unlimited Options, Limited Price

The limited version of the Report Module provides an affordable way to create and schedule just a few reports. Speak to our sales department for more details.

Want to Learn More?

For more documentation on the Reporting module, please see the [Reporting](#) section.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

OPC-UA Module



Cross-Platform OPC-UA Functionality

The Ignition OPC-UA module offers OPC-UA server functionality with a variety of device drivers and a robust, open driver API.



OPC-UA Server Functionality

This module turns Ignition into an OPC-UA server, capable of handling connections from any OPC-UA compatible client. Outgoing connections can also be made to 3rd-party OPC-UA servers. In this way Ignition functions as both an OPC-UA server and an OPC-UA client.

Built-in Device Drivers

The OPC-UA module offers a number of device drivers for common protocols out of the box, and is easily expandable thanks to the hot-swappable module architecture in Ignition. New drivers can be downloaded and installed in minutes without requiring a system restart or otherwise affecting other parts of the Ignition platform.

Redundancy Support

The OPC-UA module ties into the Ignition redundancy in order to provide efficient access to device data along with failover redundancy, with no additional configuration.

Public Driver API

Anyone can create new drivers thanks to the open driver API, and users can download and install drivers created by other developers. This is the first time such an API has been made publicly available for a product like Ignition. For more information about creating drivers for Ignition, visit the [Inductive Automation website](#).

Driver Modules

Drivers for the OPC-UA module are deployed as modules themselves. While they don't add a visible element to the system, they are loaded and upgraded in the same manner as other Ignition modules. A list of available drivers can be found on the [Drivers](#) page.

Ready to Get Started?

Documentation on the OPC-UA module can be found in the [OPC-UA and Device Connections](#) page.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Drivers

Available Drivers

The following drivers are available for Ignition's OPC-UA module.

Allen-Bradley Ethernet

The Allen-Bradley brand of programmable logic controllers (PLC) is a popular choice in many industries. Ignition can connect to the following Allen-Bradley controllers:

PLC	Capability
Logix Driver	Connect to Allen-Bradley Logix family devices, including devices with firmware v21+.
ControlLogix	Connect to ControlLogix firmware v20 and prior processors.
CompactLogix	Connect to CompactLogix firmware v20 and prior processors.
MicroLogix	Connect to MicroLogix 1100, 1200, 1400 and 1500 series PLCs.
PLC5	Connect to PLC5s via Ethernet.
SLC	Connect to SLC 5/05s via Ethernet.

See the following links for detailed step-by-step instructions on how to connect to these PLCs:

- [Connecting to ControlLogix v21](#)
- [Connecting to ControlLogix](#)
- [Connecting to CompactLogix](#)
- [Connecting to MicroLogix](#)
- [Connecting to PLC5](#)

- [Connecting to SLC](#)
- [Allen-Bradley Connection Paths](#)

Modbus

The Modbus Driver module allows the Ignition OPC-UA server to communicate with any device that supports the Modbus protocol, that is, the Modbus TCP and Modbus RTU over TCP protocols.

The Modbus Driver module can connect directly to devices that support Ethernet communications or it can also connect to Modbus devices through a Gateway device.

See the following sections for detailed step-by-step instructions:

- [Connecting to Modbus Device](#)
- [Modbus Addressing](#)
- [Modbus Address Mapping](#)

Siemens

The Siemens Drivers module provides support for connecting to the following PLCs via TCP/IP using the S7 protocol:

- S7-300
- S7-400
- S7-1200

For step-by-step instructions on how to connect to Siemens Drivers, see the [Siemens](#) page.

UDP and TCP

The UDP driver is a strictly passive listener. The UDP driver is configured to listen to one or more ports on a given IP address.

The TCP driver is configured to connect to one or more ports on a given IP address. This driver can listen, as well as write back to the connected port.

Rules are configured that dictate how the incoming data is interpreted.

For step-by-step instructions on how to connect to TCP Drivers, see the [UDP and TCP Driver](#) page.

DNP3

The DNP3 module allows the Ignition OPC-UA server to communicate with other devices that support the DNP3 protocol (DNP3 outstations).

See the following sections for detailed step-by-step instructions, see the [DNP3](#) page.

Omron

The Omron Driver module allows the Ignition OPC-UA server to communicate with NJ-series controllers.

More information on the Omron driver can be found on the [Omron](#) page.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Mobile Module



Mobile Manufacturing

Access Control Systems on Mobile Devices

Get mobile access to your control system via an iPhone, iPad, Android and other smartphones and tablet devices. The Ignition Mobile Module gives you instant access to any HMI / SCADA project created with the Ignition Vision Module.

Mobile Control System

Access control systems on the go and interact with real-time manufacturing production monitoring, efficiency and downtime data. Get instant, easy data visualization with custom graphical executive dashboards. Easily interact with your control system using any smartphone or tablet device that is equipped with a modern web browser.

No App Installation Required

With the Ignition Mobile Module installed on your central Ignition Gateway server, a supervisory control and data acquisition system (SCADA) can be accessed by cell phones, mobile tablets or PDAs with a modern web browser and WIFI connection. Wireless devices display and interact with the same real-time data that is viewed on the manufacturing plant floor.

Data Security

SSL encryption can be enabled to provide enterprise-grade SCADA security for mobile devices. Sensitive data like log-in credentials are always securely encoded when using a cell phone or wireless tablet, even without SSL encryption turned on. Ignition's flexible role-based security allows mobile projects to authenticate using various sources, such as Active Directory™.

Easy Implementation

The Ignition Mobile Module launches standard Ignition Vision Module projects on mobile devices. This means that existing HMI / SCADA screens can be launched as-is without any redevelopment. Mobile-optimized control system projects can be created using the familiar, powerful component set provided by the Ignition Vision Module software.



Want to learn more?

Read more about the Mobile module in the [Mobile Client Settings](#) section.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Alarm Notification Module



Alarm Notification

Manage SCADA Alarm Notifications Any Way You Choose

The Ignition Alarm Notification Module enables you to configure the logic for how, why and when alarm notifications are delivered, manage alarm notification for groups of users, and send notifications via email. Notifications can be easily be acknowledged by the recipient via email.

Control Your Alarm Notification Logic

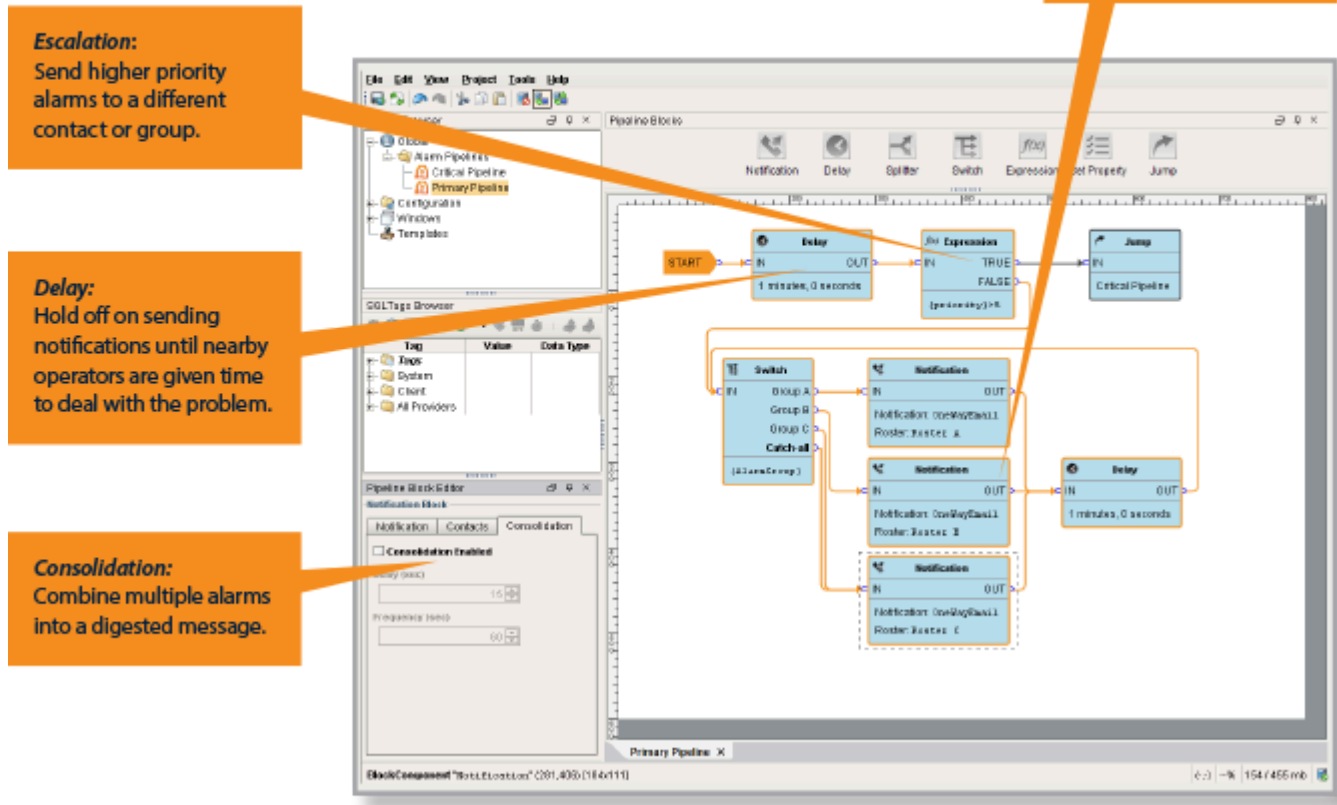
Ignition's alarm notification pipeline feature is an innovative tool that lets you easily chart out the routing of your alarms. Make your alarm notification logic as simple or as multifaceted as you want. Prevent alarm flooding by directing the right alarms to the right people at the right time.

With Ignition and the Alarm Notification Module, you'll have greater control of what happens between when an alarm or alert goes active and when it is sent out to personnel, and of who gets alarm notifications and when.

By using a simple drag-and-drop interface, you can build various types of alarm notification logic, including:

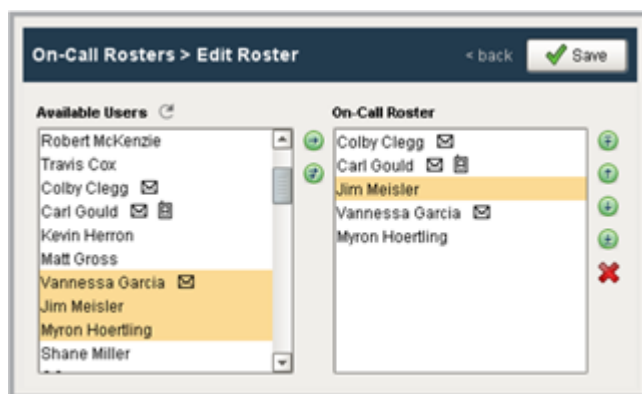
- **Delay:** Hold off on sending notifications until nearby operators are given time to deal with the problem first.

- **Escalation:** Elevate alarm notification when operators are not handling alarms.
- **Consolidation:** Combine multiple alarms into a digested message.
- **Selection:** Distribute different alarms to different groups of contacts.



Manage Users & Schedules to Ensure Timely Delivery

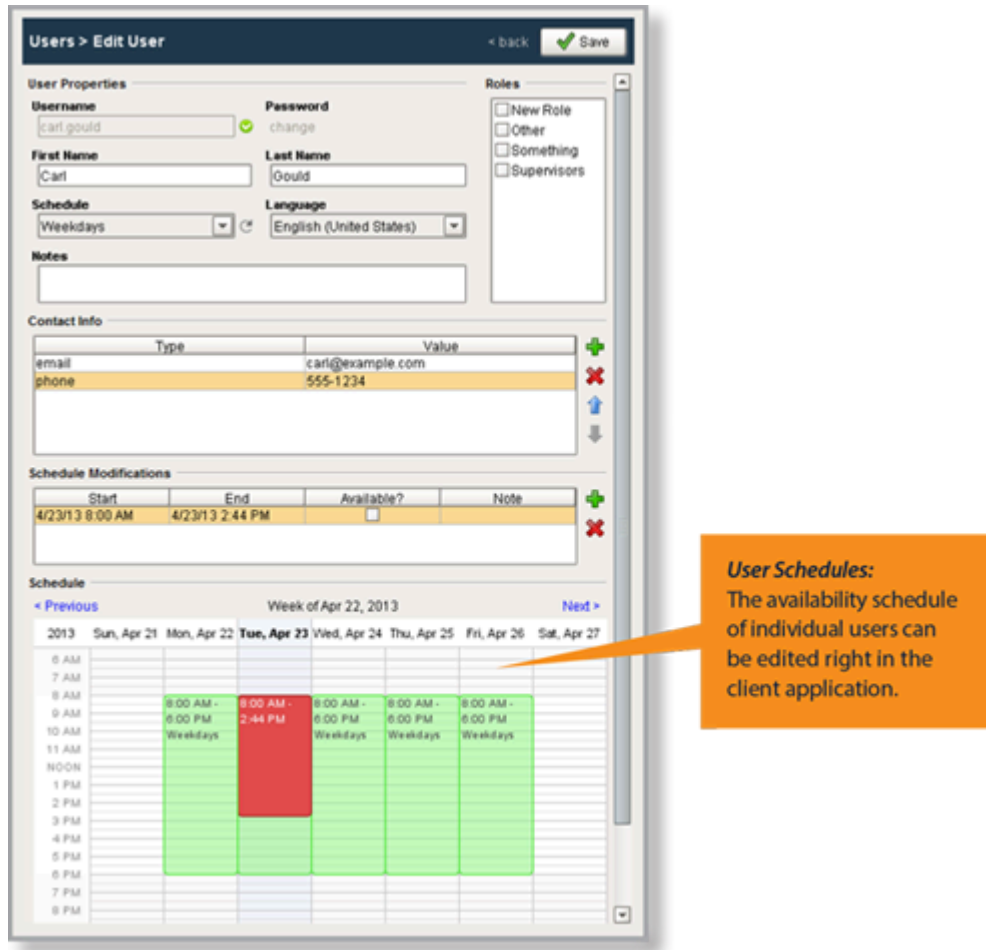
Make sure the right people – and especially those on shift – get notified quickly. The Alarm Notification Module lets you organize individuals into groups based on their roles with the On-Call Roster feature.



When an alarm notification needs to be sent, it goes to a designated call roster and then the system evaluates individual schedules so that only those currently on shift will be notified. You can easily

change schedules, assign roles and specify who should be notified when alarms pass through different notification blocks in the pipelines.

The Alarm Notification Module lets you edit rosters in the client in real time and to edit individual schedules right in the authentication profile. With permission roles, you can let individuals edit their own information on-the-fly to account for irregular time away from work so that alarms will skip over them and be sent quickly to an available person.



Users > Edit User < back Save

User Properties

Username: change Password:

First Name: Last Name:

Schedule: Language:

Notes:

Roles

☐ New Role
☐ Other
☐ Something
☐ Supervisors

Contact Info

Type	Value
email	carl@example.com
phone	555-1234

Schedule Modifications

Start	End	Available?	Note
4/23/13 8:00 AM	4/23/13 2:44 PM	☐	

Schedule

[< Previous](#) [Next >](#) Week of Apr 22, 2013

2013	Sun, Apr 21	Mon, Apr 22	Tue, Apr 23	Wed, Apr 24	Thu, Apr 25	Fri, Apr 26	Sat, Apr 27
6 AM							
7 AM							
8 AM							
9 AM		8:00 AM - 6:00 PM	8:00 AM - 2:44 PM	8:00 AM - 6:00 PM	8:00 AM - 6:00 PM	8:00 AM - 6:00 PM	
10 AM		Weekdays	Weekdays	Weekdays	Weekdays	Weekdays	
11 AM							
NOON							
1 PM							
2 PM							
3 PM							
4 PM							
5 PM							
6 PM							
7 PM							
8 PM							

2-Way Email Notification

With the Alarm Notification Module, users in your organization can receive and acknowledge alarm notifications by email.

Let's Get Started!

Learn more about alarm notifications in the [Alarm Notification](#) section.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

SMS Notification Module



SMS Alarm Notification

Add SMS Functionality to Your Alarming System

Quickly let key people know about any alarm by sending a text message to any cell phone. The Ignition SMS Alarm Notification Module enables both alarm notification and acknowledgment via text messages. You can enable phone-call alerts by adding the Voice Notification Module to your Ignition system.

Notify Users Quickly with a Text Message

By adding the SMS Notification Module to Ignition, you can send and acknowledge alarm notifications via text messaging. SMS notification is an efficient and immediate method that makes it easier for users to respond to alarms no matter where they are.

This module requires the [Alarm Notification Module](#) and a cellular modem.



Get Started

Read more about this module on the [SMS Notification Profile](#) page.



This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

SFC Module



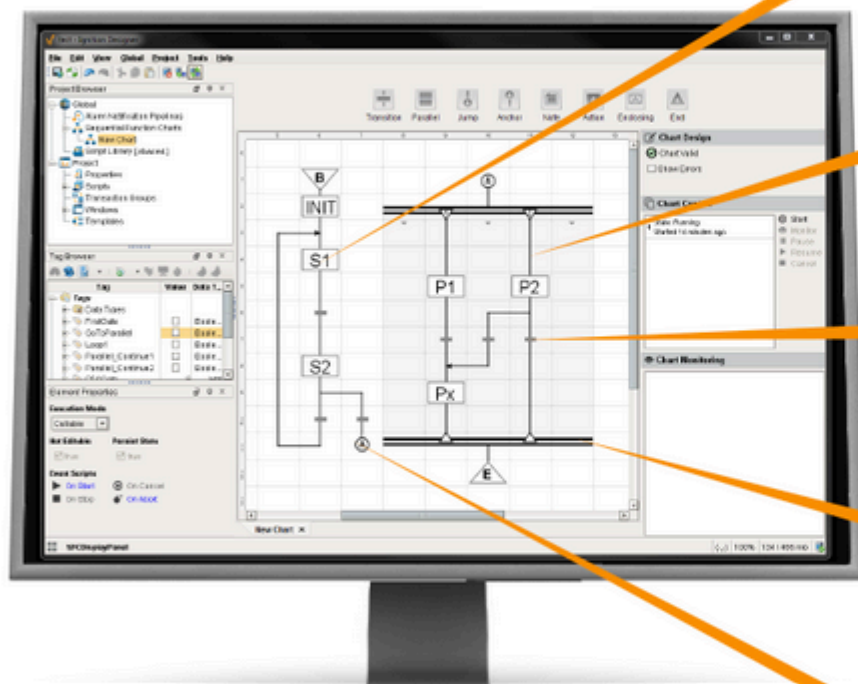
Sequential Function Charts

Streamline the creation, organization, and visualization of robust logic systems.

The Ignition SFC Module's powerful visual programming language allows you to write complex logic in a simpler, more intuitive way. The SFC Module works seamlessly with Ignition, and represents a leap forward in the capabilities of the software.

Visual Logic Programming

The SFC Module's visual, drag-and-drop programming tool makes the process of writing complex logic much smoother than using the basic Python scripts included in Ignition. Its standardized programming method, part of the IEC 61131-3 specification, will be familiar to many users who have experience in programming PLCs. With a simple visual interface, you can use steps, transitions, parallel blocks and other drag-and-drop elements to create complex charts. SFCs within Ignition can make use of scripts, Tags, and other elements already available in the Ignition platform.



Steps:
These elements perform a variety of useful functions.

Directed Links:
Use these to connect steps and transitions to each other.

Transitions:
Conditions that evaluate to true or false, either allowing the flow to proceed or blocking it.

Parallel branches/syncs:
Use these to set up parallel actions.

Jumps:
Set up connections without crossing links.

Drag and drop powerful components to create robust logic systems.

Chart Monitoring

You can monitor charts visually in the Ignition designer and clients, watching each chart execute its logic, down to which specific step is currently running. You can even inspect the chart's internal variables as it runs. The visual nature of SFCs make them much easier to troubleshoot than Python scripts.

Build Robust Logic

Now it takes far less time to build any type of robust logic your enterprise may need: long-running tasks, parallel tasks, tasks that pause for certain conditions before proceeding, loops, sub-charts, and more.

Hot Editability

In some circumstances, changes can be made to charts that are already running. This can be a powerful way to fix mistakes without requiring a chart to be restarted.

Multiple Chart Instances

The module allows you to instantiate multiple instances of any chart. Each instance can be configured with different starting parameters, allowing the instances to be independent from each other.

Long-Running Charts

Charts may be designed to run for a very long time. They can even keep running if a Gateway is restarted by pausing on shutdown and resuming on startup. SFCs also support redundant Ignition setups to ensure the chart keeps running even if the primary Gateway fails.

Powered by Ignition

The SFC Module is a SCADA module for Ignition, the powerful, award-winning, HMI / SCADA software from Inductive Automation. The SFC Module integrates seamlessly with Ignition and with all other modules in the Ignition SCADA suite. Because the module is built upon the power of Ignition, it shares the same advantages, such as cross-platform compatibility, unlimited free clients, robust out-of-the-box SQL database support, and fast installation. Leveraging the full power of Ignition, it is a SCADA logic engine unlike any other on the market.

Learn More

Documentation on the SFC module is available in the [Sequential Function Charts](#) section.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Enterprise Administration Module



Enterprise Administration

Enhance the reach of Ignition with the Enterprise Administration Module (EAM)

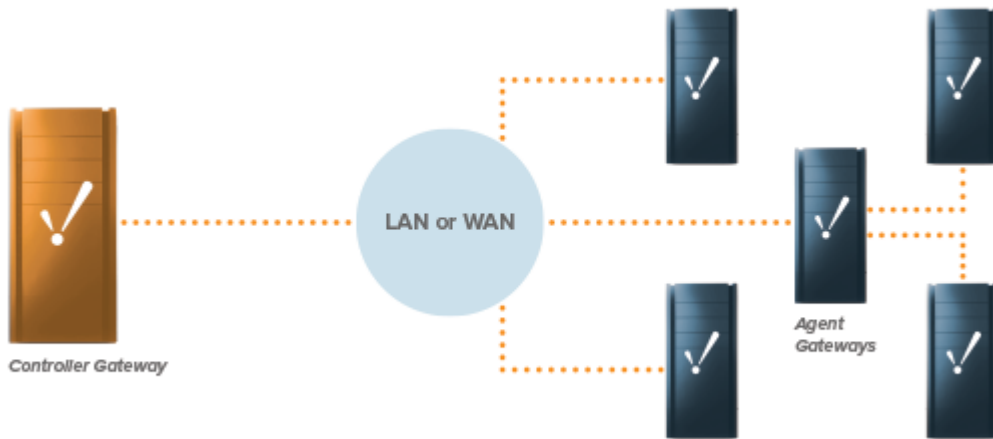
The EAM provides a secure and intuitive way to manage many Ignition installations from one location. It is ideal for large enterprises that deploy multiple Gateways across vast geographical distances, but even companies with a few Ignition Gateways on a single plant floor can benefit from its ability to synchronize projects, monitor performance, and automate backup and recovery from a central location.

Manage Multiple Gateways from One Gateway

With the EAM Module installed, an Ignition Gateway can function as the central Controller Gateway that connects with, monitors, and manages any number of remote Agent Gateways. Use the Controller Gateway to coordinate and automate many administrative tasks for Agent Gateways, including:

- Assigning, updating, activating, and unactivating licenses
- Deploying new and updated modules
- Disaster Recovery
- Remote Backups

All of these actions can be executed immediately or scheduled for later.



Synchronize Projects and Resources

The EAM gives you the ability to design resources and projects centrally in the Controller Gateway and send them out to the Agent Gateways.

Communicate Over a Secure Network

The EAM is built on a high-performance, high-security communication network that facilitates simultaneous access to multiple Gateways without causing firewall issues.

Monitor Agent Health and Performance

In the Controller Gateway, the EAM provides health indicators for Agent Gateways and alarms for impending lifecycle events such as Agent Gateway failure. Memory usage notifications, warnings, and error events are also provided.

Automate Gateway Backup and Recovery

The EAM enables automated backup and quick recovery for all Gateways through a Gateway backup archive.

Powered by Ignition

The EAM is a SCADA module for Ignition, the powerful HMI / SCADA software from Inductive Automation. The EAM integrates seamlessly with Ignition and all other modules in the Ignition SCADA suite. Because the module is built on the power of Ignition, it shares the same advantages, such as cross-platform compatibility, unlimited free clients, robust out-of-the-box SQL database support, and

fast installation. Leveraging the full power of Ignition, it is an enterprise management tool unlike any other in the SCADA software market.

Get Started

Read more about EAM in the [Enterprise Administration](#) section.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Tag Historian Module



The Tag Historian Module for Ignition allows you to turn a SQL database into a high-performance time-series Tag historian. It requires minimal work to implement and is simple to use, even if you aren't very experienced with SQL databases. The power, ease of use, and incomparable price-to-performance ratio of the Tag Historian Module make it the clear choice for storing your organization's valuable historical information.

Store History with a Click

With the Tag Historian Module, you can store history on an Ignition Tag to your database with the click of a mouse - no complex configuration, tuning, or data modeling required.

Create Tables Effortlessly

The module creates and manages tables for you, so you don't need to be knowledgeable about database management to use it.

Easy Query Binding

Use the data intuitively throughout Ignition with simple, visual query binding screens – no need to write SQL or complex queries.

Powerful Historian Engine

The module's high-powered historian engine automatically supports compression, partitioning, interpolation, and aggregation for any SQL database.

Make Data More Accessible

By recording your data to a SQL database, the module makes your process data available to any application with database access.

Real-Time Charts and Trends

Combine with the Vision Module to simply pull the information you need into individual charts and graphs.

Great Performance and Value

By leveraging the power of SQL databases, the Tag Historian Module delivers tremendous performance at a fraction of the price of traditional historian products.

Powered by Ignition

The Tag Historian Module is a SCADA module for Ignition, the powerful HMI / SCADA software from Inductive Automation. It integrates seamlessly with the Ignition platform and with all other modules in the Ignition SCADA suite. Because the module is built on the power of Ignition, it shares the same advantages, such as cross-platform compatibility, unlimited free clients, robust out-of-the-box SQL database support, and fast installation. Its ability to leverage the full power of Ignition places it in a class above any other Tag historian on the SCADA software market.

Learn More

Read more about Tag Historian in the [Historian](#) section of the manual.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Additional Modules

Enhance Ignition's capabilities beyond traditional SCADA and HMI limits. These additional modules provide Ignition with even more functionality. Take your project to the next level with these powerful modules.



Voice Notification

Enable both alarm notification and acknowledgement via a phone call.



OPC COM

Integrate Ignition with Classic COM-based OPC Servers.



OPC COM Tunneller

Exposes OPC servers accessed through the OPC-COM module to Ignition's OPC-UA server. Allowing you to easily expose tags from a DA server to any UA server.



Web Browser

View web-pages directly from Ignition clients.



WebDev

Create web-pages that are hosted on Ignition's web server.



Twilio Alarm Notification

Adds integration with Twilio via an SMS Alarm Notification Profile and scripting functions to send SMS.



Serial

Adds platform-level serial comm support and serial-specific scripting functions.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Voice Notification Module



Text to Speech Alarm Notification

Send Notifications Via a Phone Call

Ensure no serious issue is overlooked by speeding alarm notification delivery to the right people as soon as possible. The Ignition Voice Notification Module uses state of the art text-to-speech technology to add phone-call notification to your alarming / alerting process.

Set up voice notification for alarms in Ignition with the Voice Notification Module. Use text-to-speech (TTS) technology to notify users of an alarm via a phone call and to allow users to acknowledge it with a simple key press. The VoIP-over-SIP voice notification system can be tied in with a VoIP phone system or online VoIP service, or bridged to a standard phone line using an existing phone gateway.

The Voice Notification module adds the ability to deliver alarm notifications to users via telephone, using any SIP compatible phone system. Messages are constructed in text, and are delivered through a high quality text-to-speech engine.

Alarm Notification Module Compatibility

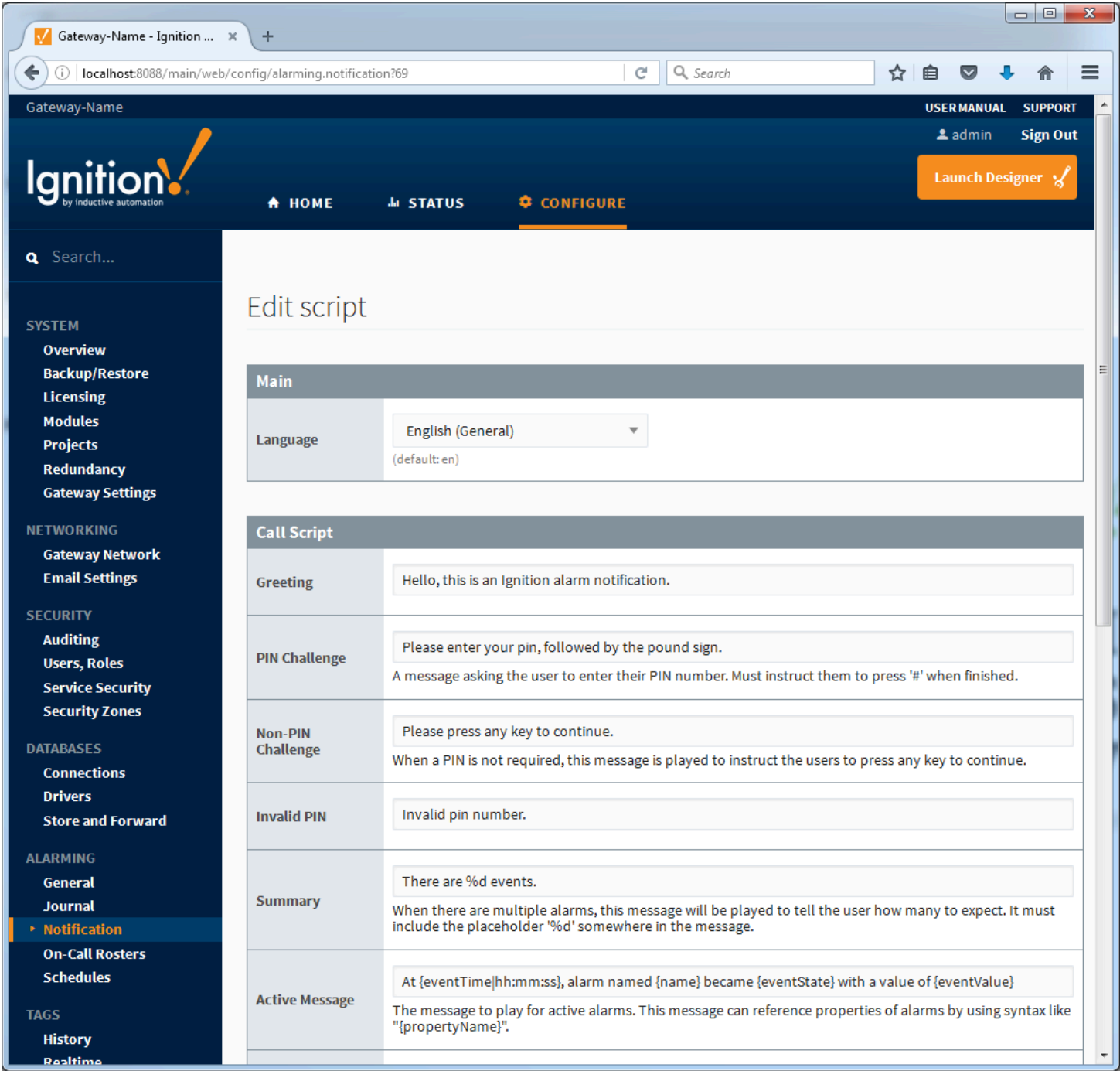
The Voice Notification Module is fully compatible with the Alarm Notification Module, so all pre-existing on-call rosters and pipelines will work with Voice notification.

Two-Way Acknowledgement

The Voice Notification Module can be configured to allow users to quickly acknowledge alarms. With the Two-Way property enabled, the notification is sent out a response code in the message. The user simply replies with the response code and the alarm will be acknowledged.

Custom Notification Messages

The High-Quality Text-to-Speech Engine allows you to customize the alarm message the user hears, so you are not limited to a number of pre-recorded messages.



Audit Log Integration

The Voice Notification module can be configured to log successful message delivery and user acknowledgement into any connected database. Once recorded, a simple query will return all notification activity.

Multi-Language Support

Write your messages in any of the provided languages. A list of the available languages can be found on the [Voice Languages](#) page.

NOTE

This module requires the Alarm Notification Module and some SIP-compatible VoIP service, such as your office VoIP server, Skype Connect or an IP Telephony Gateway appliance to bridge to a standard phone line (not included).

Examples of a IP Telephony Gateway appliance are:

- Atcom IP02/IP04 (the number indicates the phone lines. Ignition can only use 1 line at a time)
- GXW410x (It is the FXO port that connects to the phone line)

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

OPC COM Module



Connect to Legacy Devices

The OPC-COM module offers Ignition the ability to connect to legacy ("classic") COM-based OPC-DA servers. It supports OPC-DA 2.0 and 3.0.

Connectiong to Classic OPC Servers

With the OPC-COM module installed, there will be a new option for COM-based OPC servers when creating a server connection in the Gateway. The OPC-COM module is primarily inteded for use with local OPC servers, although it also provides basic support for remote connections.

Even when connecting locally, the application may run into the traditional difficulties of connecting to Windows based OPC servers. DCOM security settings on the machine can interfere with connections, and the OPC Core Components package (from the [OPC Foundation](#)) must be properly installed before connections can be established.

Using Data from Classic OPC Servers

After a connection to a server has been defined, the server will appear along side of other OPC servers (both COM and UA based) in the OPC Tag Browser. You can use these tags like any others - bring them into Ignition's Tag Browser, use them in Transaction Groups, and so on.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

OPC COM Tunneller Module



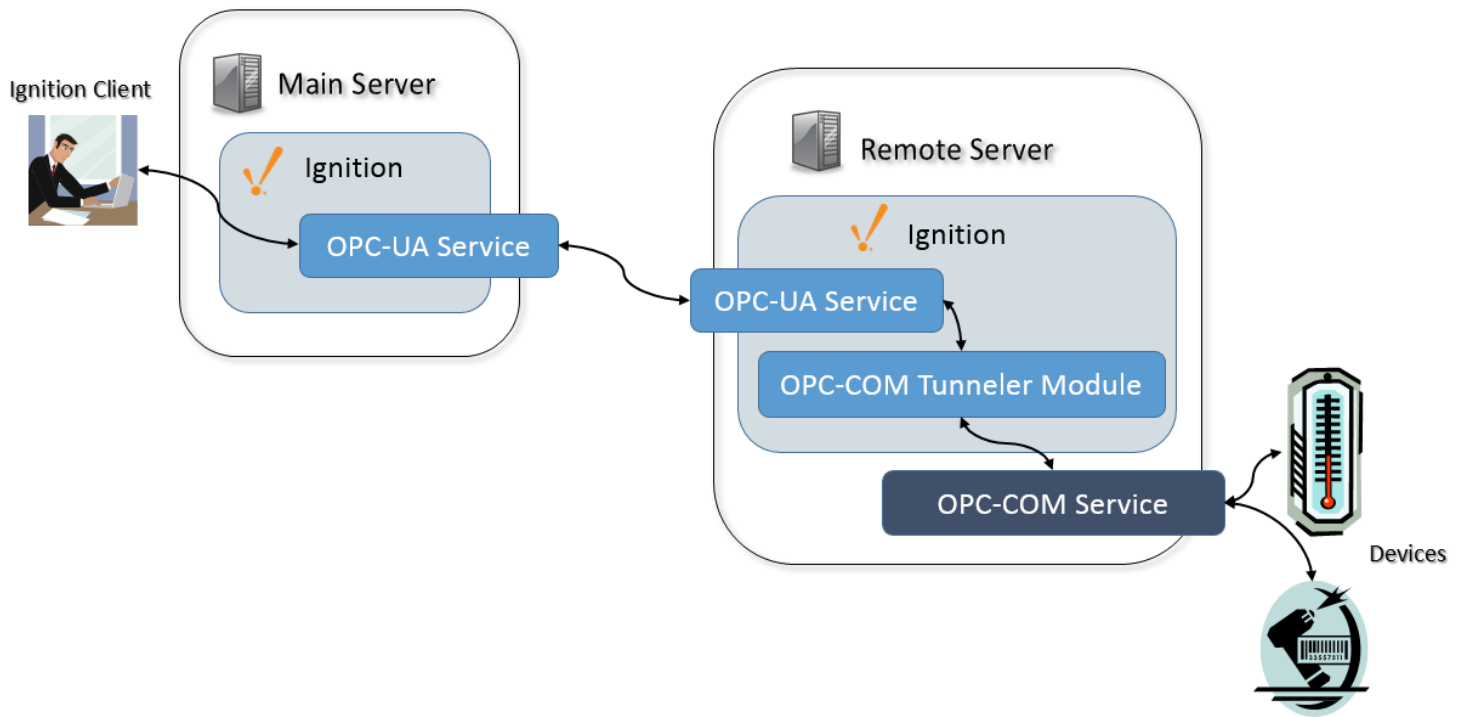
Create Remote Connections through OPC-UA

Simplify COM Connections

The OPC-COM Tunneller module eliminates many difficulties associated with establishing remote OPC-COM connections. The OPC-COM Tunneller module exposes OPC-COM connections to Ignition's OPC-UA server, allowing you to easily access them over the network and eliminate the need to use a Remote DCOM connection.

Modernize COM Servers

Ignition manages the OPC-COM connection in the context of the OPC-UA standard resulting in easier access over the network, eliminating the need to establish remote DCOM connections, and incorporating the security benefits of the TCP standard such as authentication and encryption.



This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Web Browser Module



Access the Web Directly From a Client

The Web Browser module adds a web-browser component to the Vision module. This component allows you to embed other webpages from the internet or from your local intranet inside of your projects. The browser is based on the open source project Chromium, and supports modern web technology such as HTML5, CSS, and Javascript.

More information on the Web Browser Module can be found on the [Web Browser Component](#) page.

File Edit View Project Component Alignment Shape Tools Help

Project Browser

- Header
- Main Window
- Main Window 2
- Main Window 3
- Main Window 4
- Root Container
 - Web Browser
- Main Window (1)

Tag Browser

Tag	Value	Data Type
Tags		
System		
Client		
All Providers		

Property Editor

Common

Name: Web Browser

Visible: ☒ true

Border: No Bord

Enabled: ☒ true

Behavior

Popups Allowed: ☒ true

Zoom Level: 0.0

Show Navigation ...: ☒ true

Data

Mode: URL

Starting URL: https://docs.ind...

Starting HTML: <html><body>&

Use Proxies: ☐ false

https://docs.inductiveautomation.com/display/DOC78/Web+Browser+Module

Log in

Ignition!

by Inductive automation

Search

Web Browser Module

Access the Web Directly From a Client

The Web Browser module adds a web-browser component to the Vision module. This component allows you to embed other webpages from the internet or from your local intranet inside of your projects. The browser is based on the open source project Chromium, and supports modern web technology such as HTML5, CSS, and Javascript.

More information on the Web Browser Module can be found on the [Web Browser Component](#) page.

CONTENTS

- Getting Started
 - Introducing Ignition
- Ignition Modules
 - About Ignition's Modular Architecture
- Standard Modules
 - Voice Notification Module
 - OPC COM Module
 - OPC COM Tunneller Module
 - Web Browser Module**
 - WebDev Module
 - Twilio Integration and Alarm Notification

Documentation Powered by Atlassian Confluence 5.7.4 - Get Support - Download Your Free Trial of Ignition

Component ...

- Input
- Buttons
- Display
- Tables
- Charts
- Calendar
- Admin
- Alarming
- Containers
- Misc
- Reporting
- Web Browser
- Web Browser

Welcome Main Window 4 x

Web Browser "Web Browser" (5,17) [980x649] Drag to move.

(,-) 100% 93 / 455 mb

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Web Dev Module



Use Web Services in Ignition

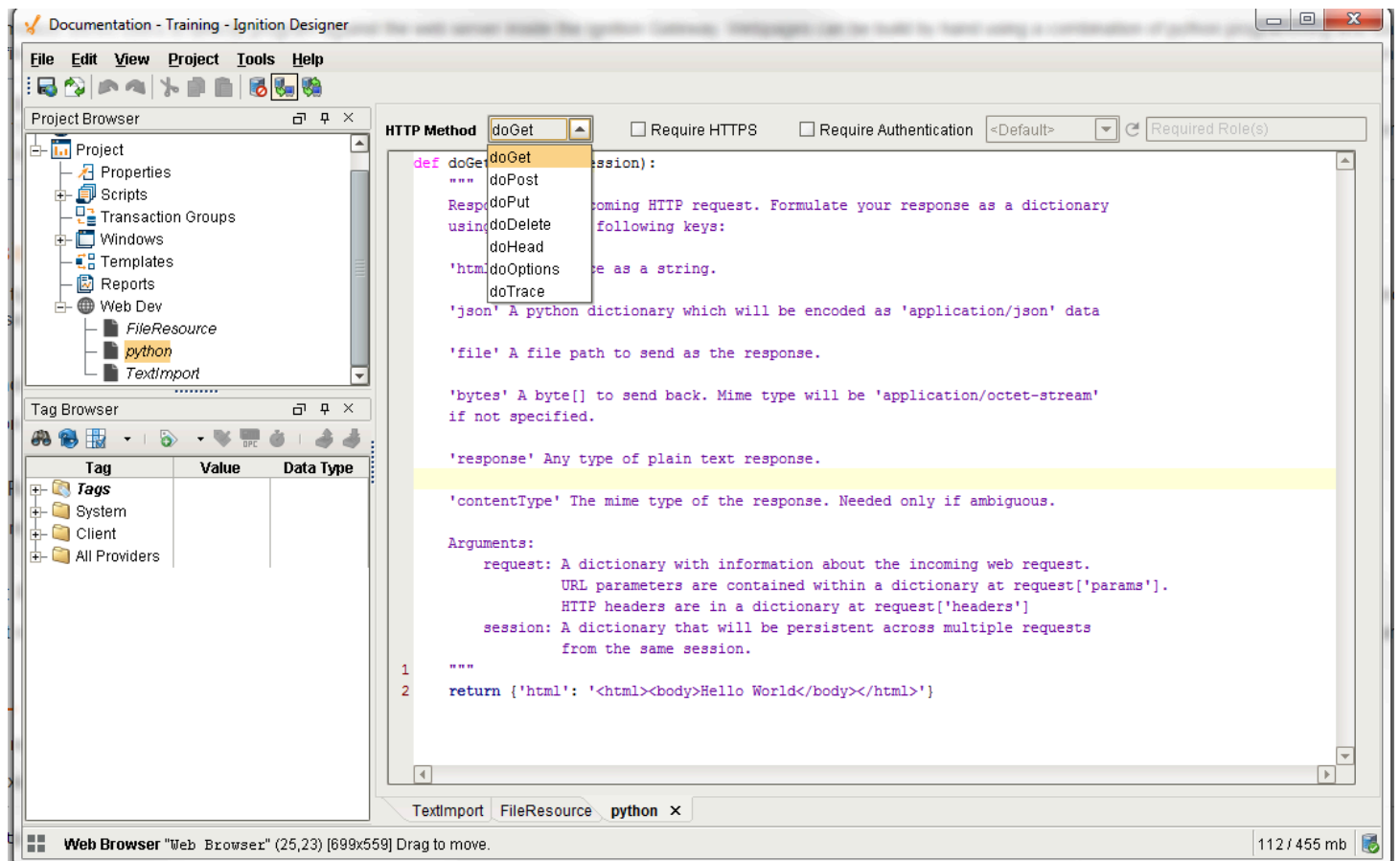
Host a Web Page on an Ignition Gateway

This module allows users to directly program against the web server inside the Ignition Gateway. Web pages can be built by hand using a combination of python programming and static web resources such as images, css files, javascript files, and html files. Design and host custom web pages directly against the Ignition service.

Configure Custom Web Service Calls

External systems can interact with the Ignition service through web service calls using the Web Dev module. Allow SAP and ESP systems to request data directly from Ignition by developing a custom, full-fledge, API.

More information on the Web Dev module can be found on the [Web Dev](#) page.



Barcode Scanners and Web Enabled Devices

Many newer barcode scanners are web-enabled and don't use serial ports to connect. For these types of scanners you can use the Web Dev Module to write whatever connect functions that you need. This opens up a whole new world of possibilities for connecting to various mobile devices.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Twilio Alarm Notification



Send SMS Messages through Scripts

The Twilio Alarm Notification module adds integration with Twilio via an SMS Alarm Notification Profile. Send SMS alarm notifications without a physical device. This module unlocks the `system.twilio` functions. These allow you to send SMS messages via a python script. See the [system.twilio](#) section of the appendix for more details.

Avoid the Hardware Hassle

The Twilio Integration and Alarm Notification module does not require additional hardware, so there is no need to worry about firmware updates or hardware failure.

More information on using the Twilio Integration and Alarm Notification module can be found on the [SMS Notification Profile](#) page.

CAUTION

The Twilio module is dependent on the Alarm Notification module. You must have both to send alarms.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Serial Modules



Serial Support

The Serial Module adds serial comm support and `system.serial` functions to python scripts. These functions allow you to open, read from, write to, and close serial ports. More information on these functions can be found in the [system.serial](#) section of the appendix.

Multiple Scopes

There are two variants of the Serial Module. Both perform similar functionality, but at different scopes:

- the Serial Support Client Module allows the `system.serial` functions to be accessed from client based scripts, as well as connect to serial devices plugged into the computer running the client.
- The Serial Support Gateway Module allows the `system.serial` functions to be accessed from gateway based scripts, as well as connect to serial devices plugged into the gateway server.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

SECS/GEM Module



Communicating with Semiconductor Fab Equipment

Ignition's SEC/GEM Module is a highly specialized module that provides the ability to communicate with semiconductor fab equipment using the SECS-II, GEM, and HSMS standards. The SECS/GEM Module is implemented as a host that can talk to one of more fab tools via equipment connections configured within the Ignition Gateway. It also provides the capability to create and configure equipment simulators that can be used for testing applications which is advantageous when getting started with the SECS/GEM Module. Connections are made over Ethernet using TCP/IP (HSMS protocol) or over a direct serial connection (SECS-I protocol) using a RS-232 cable. Once installed, the SECS/GEM functionality is fully integrated into the Configure section of the Gateway webpage where you can configure connections to the equipment, and setup equipment simulators.

The SECS/GEM Module provides the framework to create and configure a common set of scripting functions to control equipment behavior and communication capabilities that monitor and support semiconductor manufacturing equipment. Through scripting, it provides the flexibility to start and stop equipment processing, set equipment configuration parameter values, collect data, send alarms to the host, select recipes, and more as long as the host and OEM adhere to SECS/GEM standards.

Communication between Ignition and the equipment is in the form of SECS messages defined by the SECS-II standard. SECS messages are typically sent and received using scripting functions within Ignition using the SECS/GEM defined messages (SECS Definition Language (SDL)) that are loaded into the equipment. SECS messages that contain data can also be written to Tags, updating the Tag's values as new messages come in. This functionality is used to capture data coming back from equipment, such as tool traces within S6F1 messages. The default SDL file that is included with the

SECS/GEM Module is already configured to create Tags for S6F1 messages sent to the module from the tool. For other messages, the SECS/GEM Module can be configured to update Tags by adding some additional parameters to the equipment's SDL file.

To learn more, refer to the [SECS/GEM](#) section.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

System Architectures

Overview

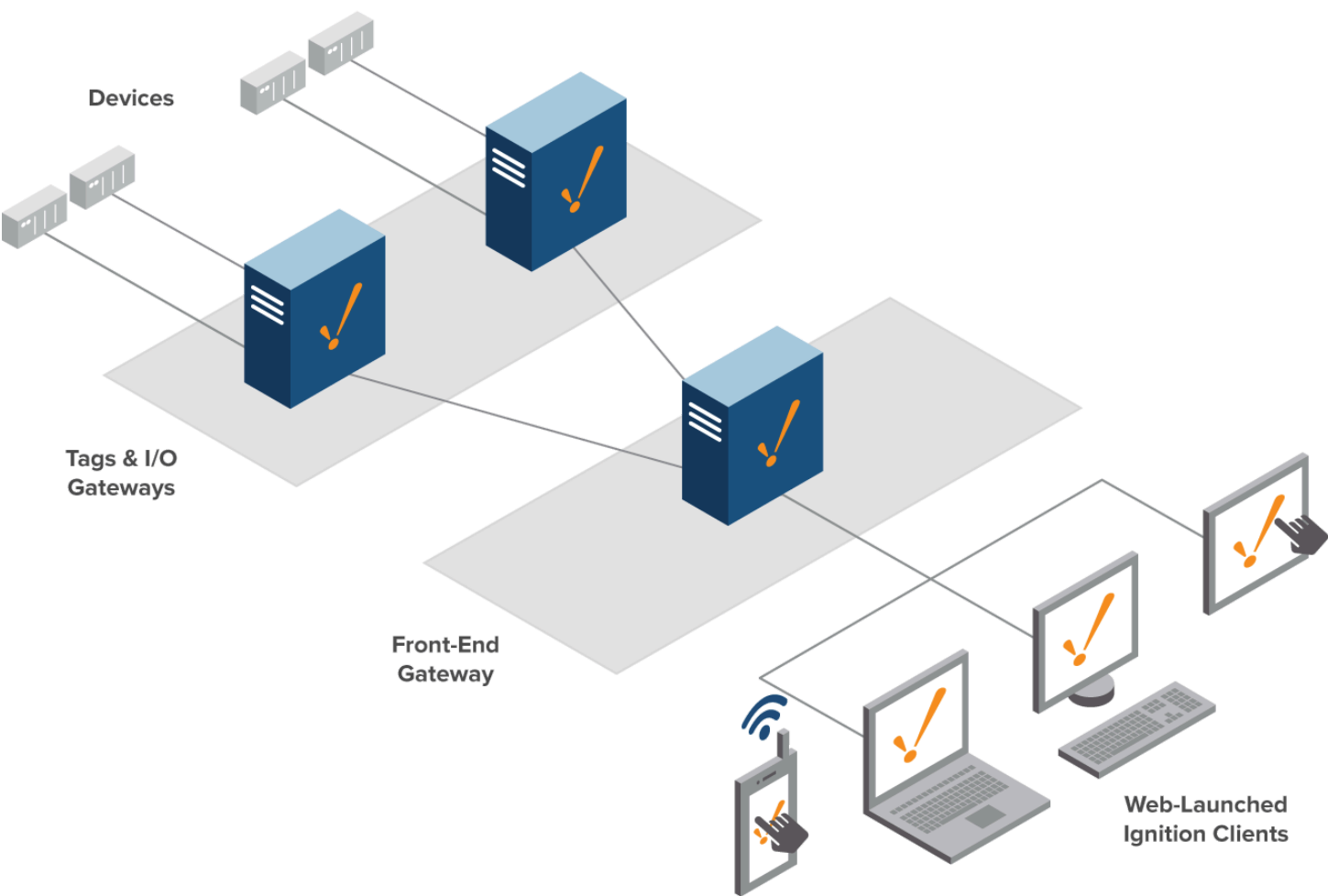
Ignition is flexible enough to meet the demands of any enterprise infrastructure, whether it be large, small, or somewhere in-between. You can deploy Ignition at one site, multiple sites, or host it in the Cloud. This section displays several common architecture examples. Many are simply examples, but are a great starting point. Additionally, the architectures shown below can be modified or scaled out to accommodate a number of different requirements.

Click on each architecture to learn more about it.

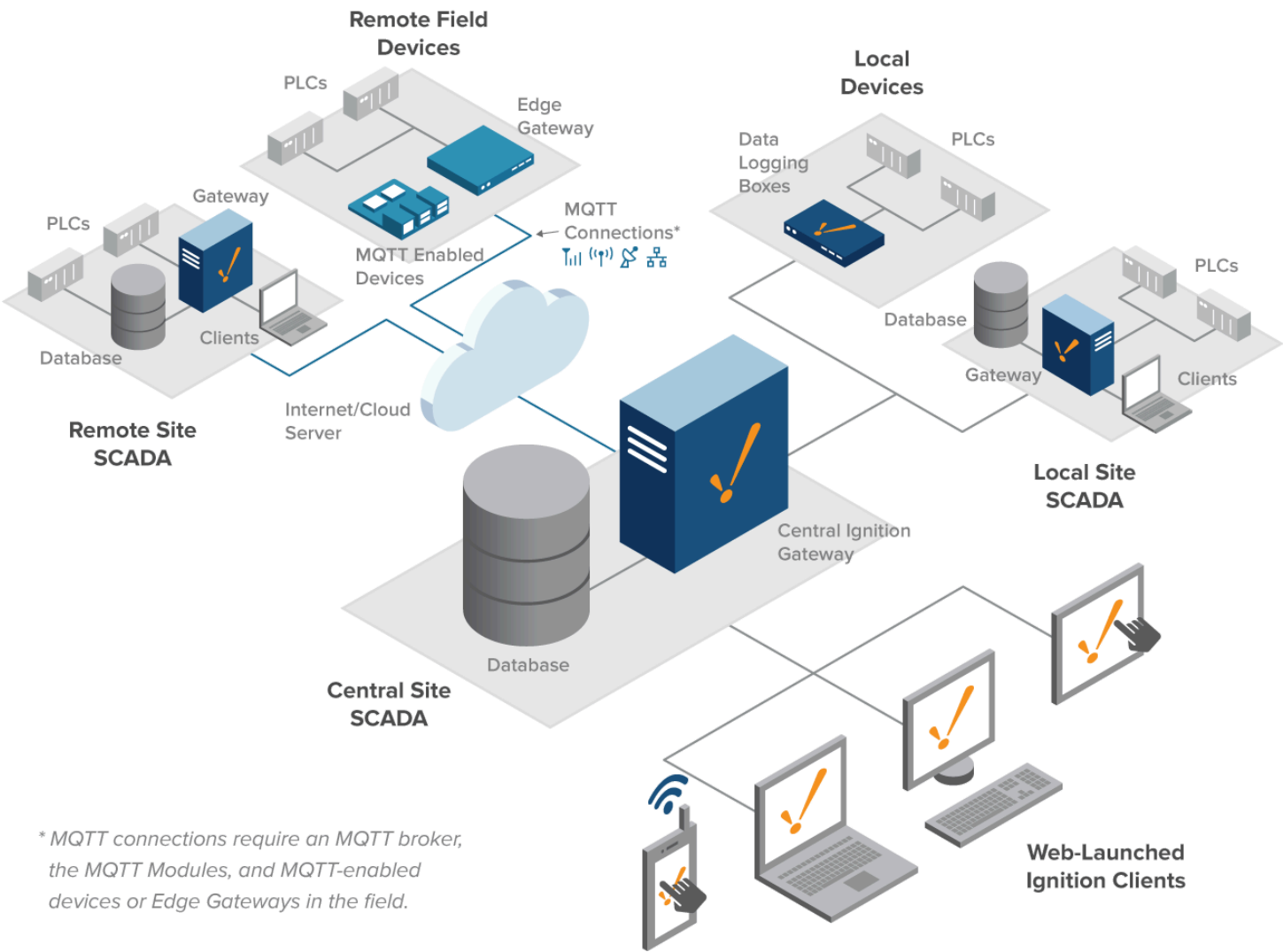
Standard Architecture



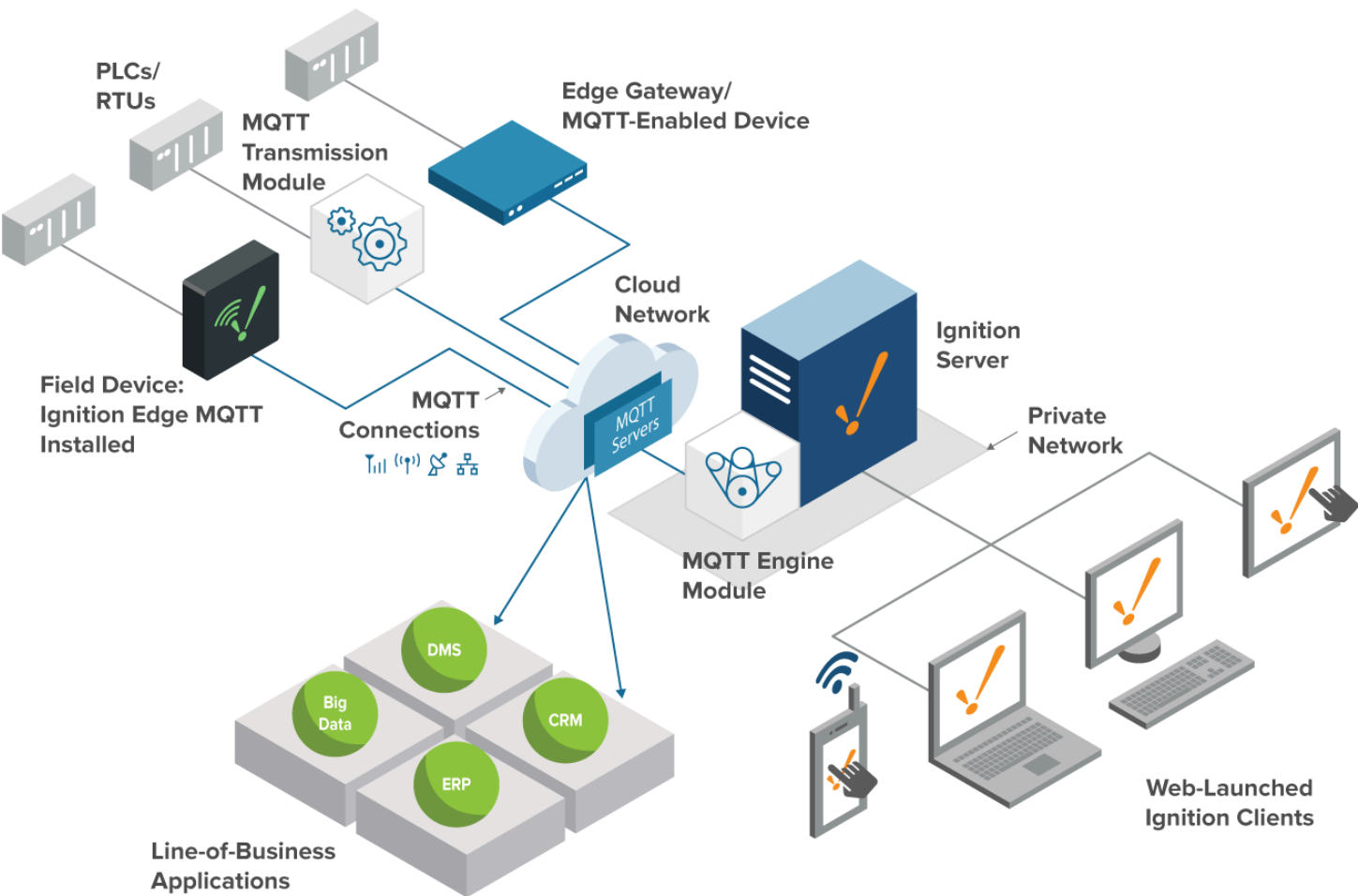
Scale Out Architecture



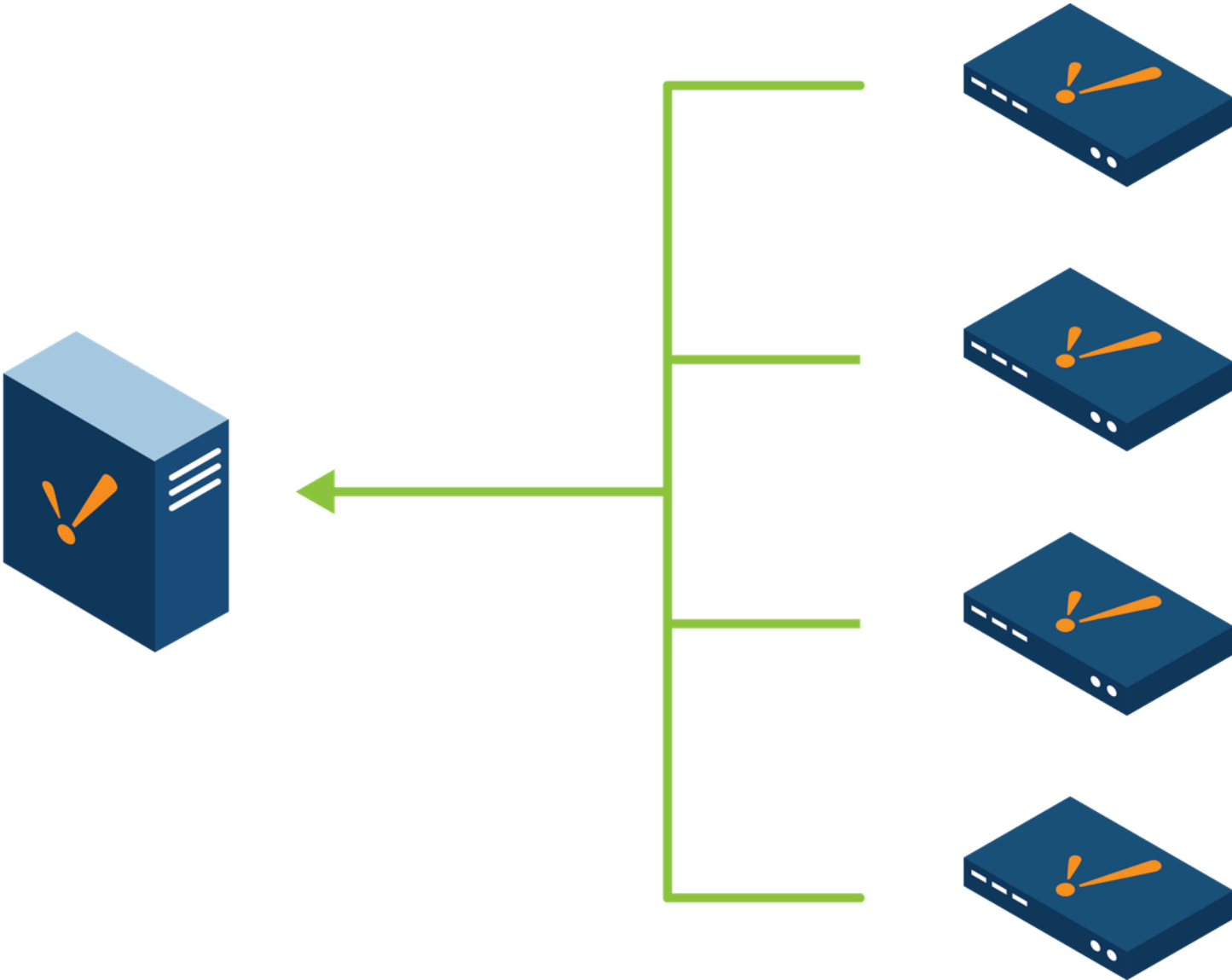
Hub and Spoke Architecture



IIoT Architecture



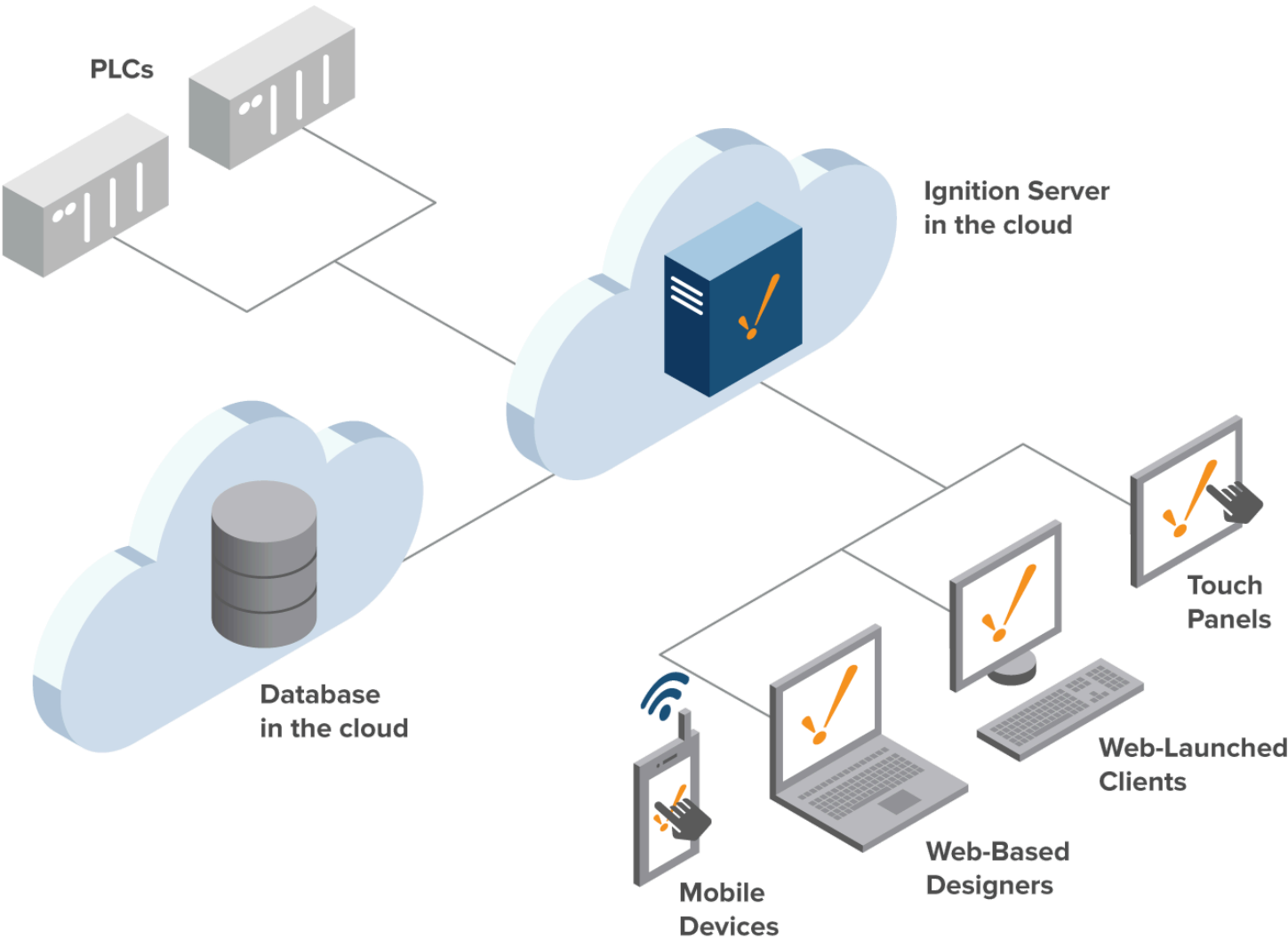
Enterprise Architecture



Redundancy Architecture



Cloud Based Architecture



This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Standard Architecture



INDUCTIVE
UNIVERSITY

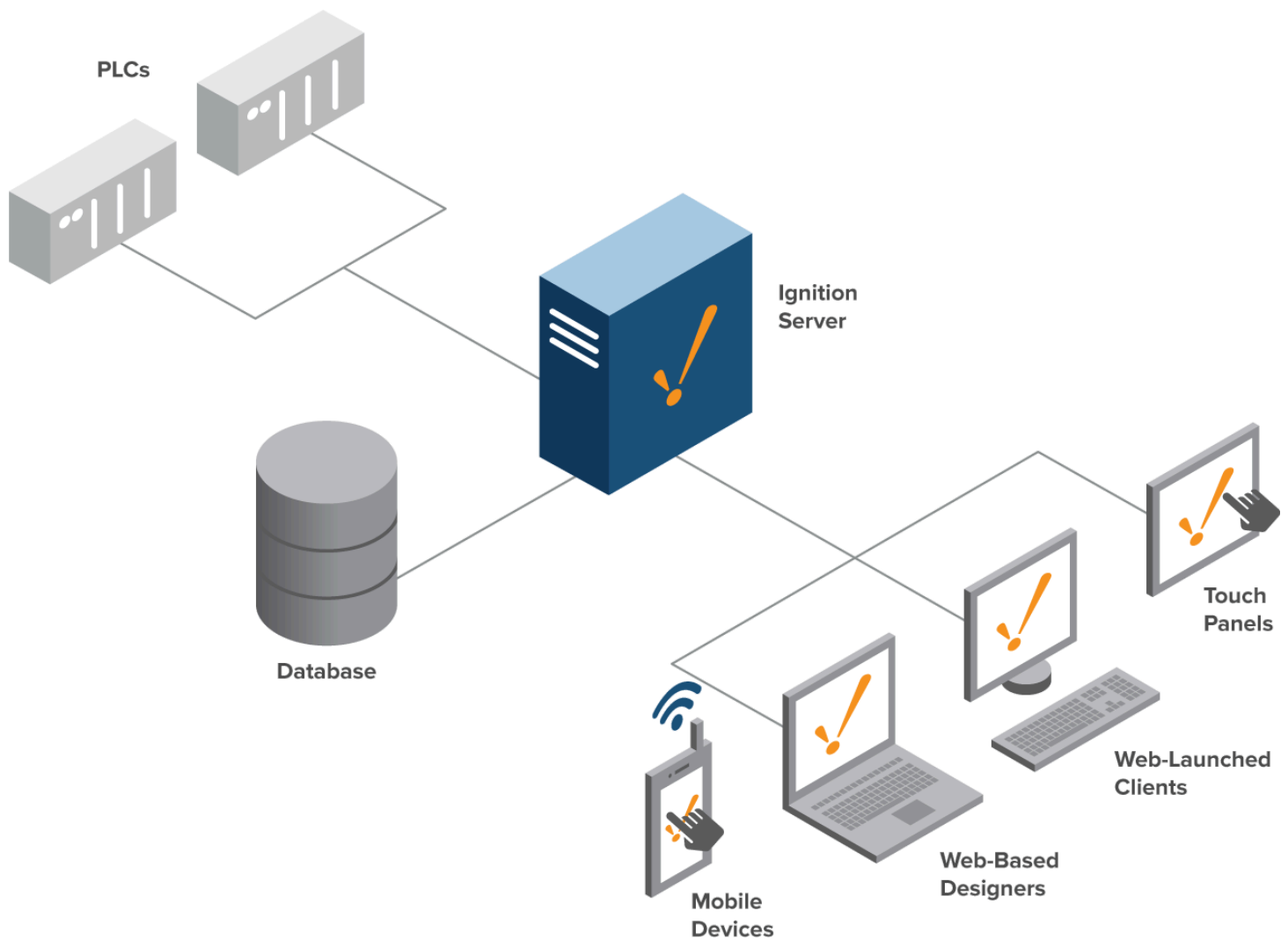
Standard Architecture

[Watch the video](#)

Single Server Install

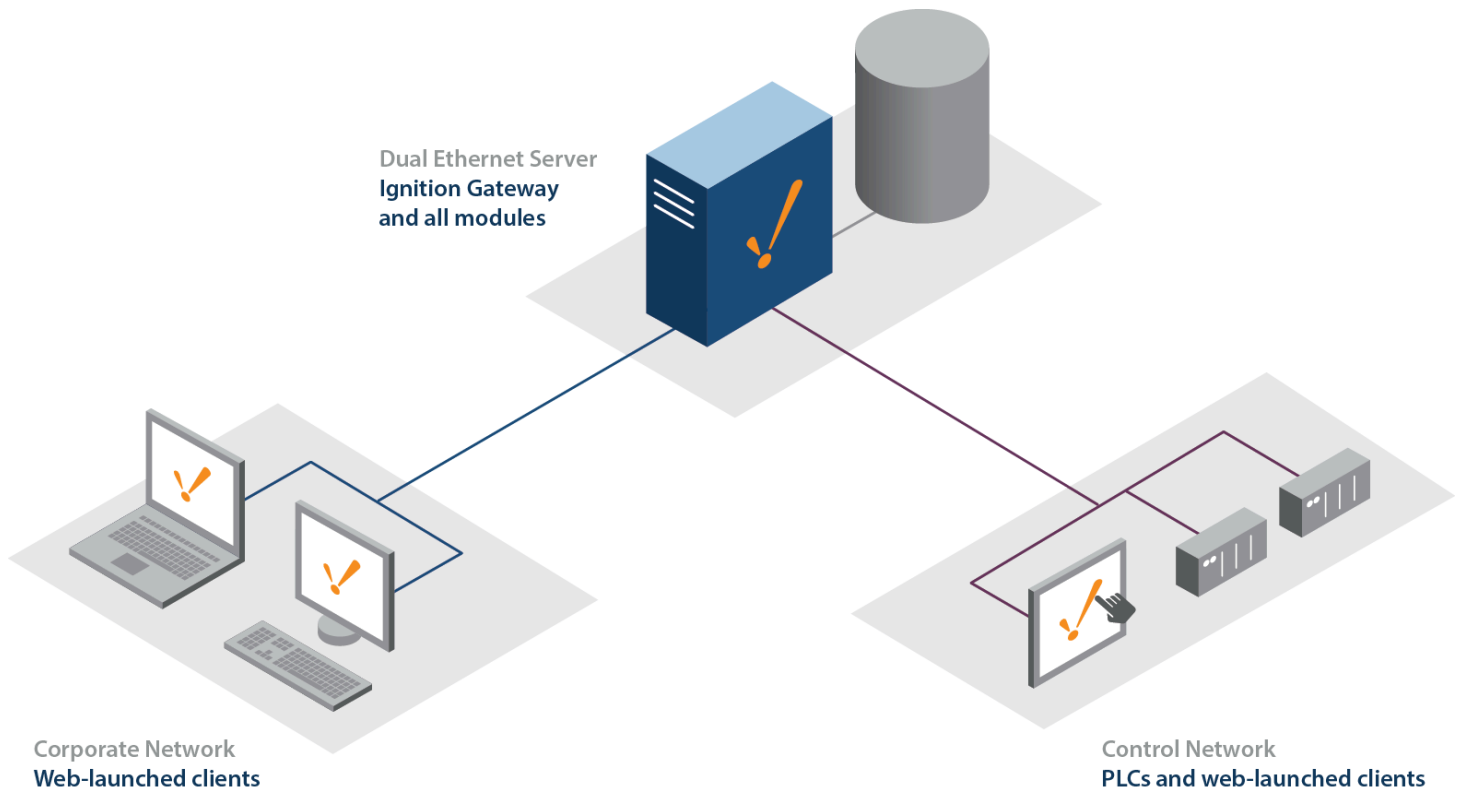
Ignition's most basic architecture is a standard single installation. This is very simple to setup and it provides the powerful functionality of the Ignition Gateway and it can serve as a starting point for other more complicated architectures.

The Standard Architecture consists of a single Gateway that can connect to multiple PLCs, databases, and other devices, all with a single licensed install. With the Ignition licensing system, adding new devices, databases, clients and more is easy. Simply connect them to the Ignition Gateway, no licensing changes required. The Ignition Gateway can launch multiple clients on everything from touch panels to workstations and even mobile devices, all from one Ignition installation.



One Server, Multiple Networks

The Ignition Gateway supports dual-NIC servers, and can act as a bridge between multiple networks, or communicate with multiple sites over a corporate WAN. Since clients talk to databases and PLCs through the Gateway, clients can be launched from both a corporate network and an isolated control network, and provide full access to both. Built-in **security settings** can restrict project access to users on different networks, either by restricting certain things in a project, or denying access to a whole project based on user role, network location, and more.



This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Scale Out Architecture



INDUCTIVE
UNIVERSITY

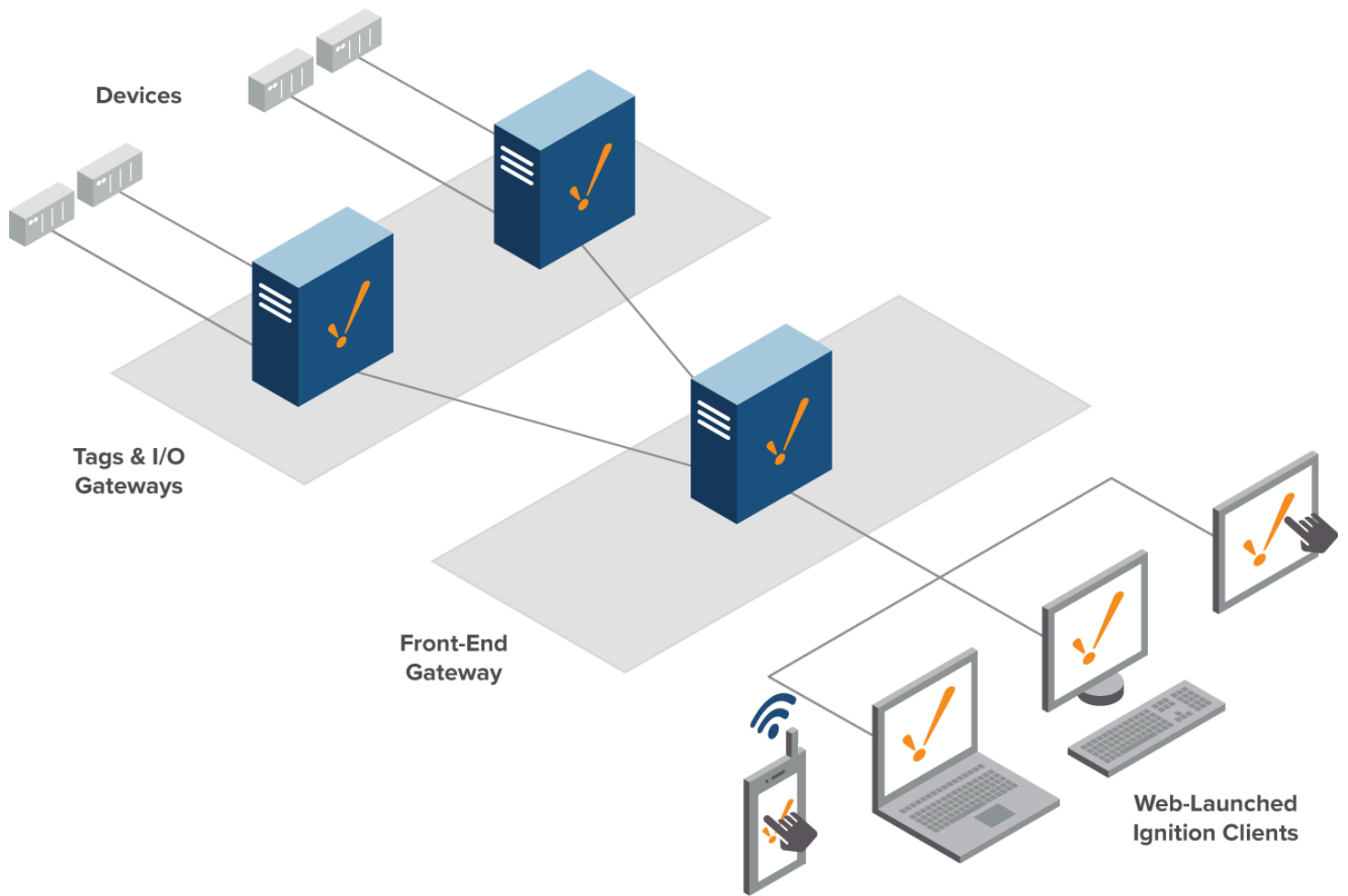
Scale Out Architecture

[Watch the video](#)

Scaling Out Ignition for Larger Systems

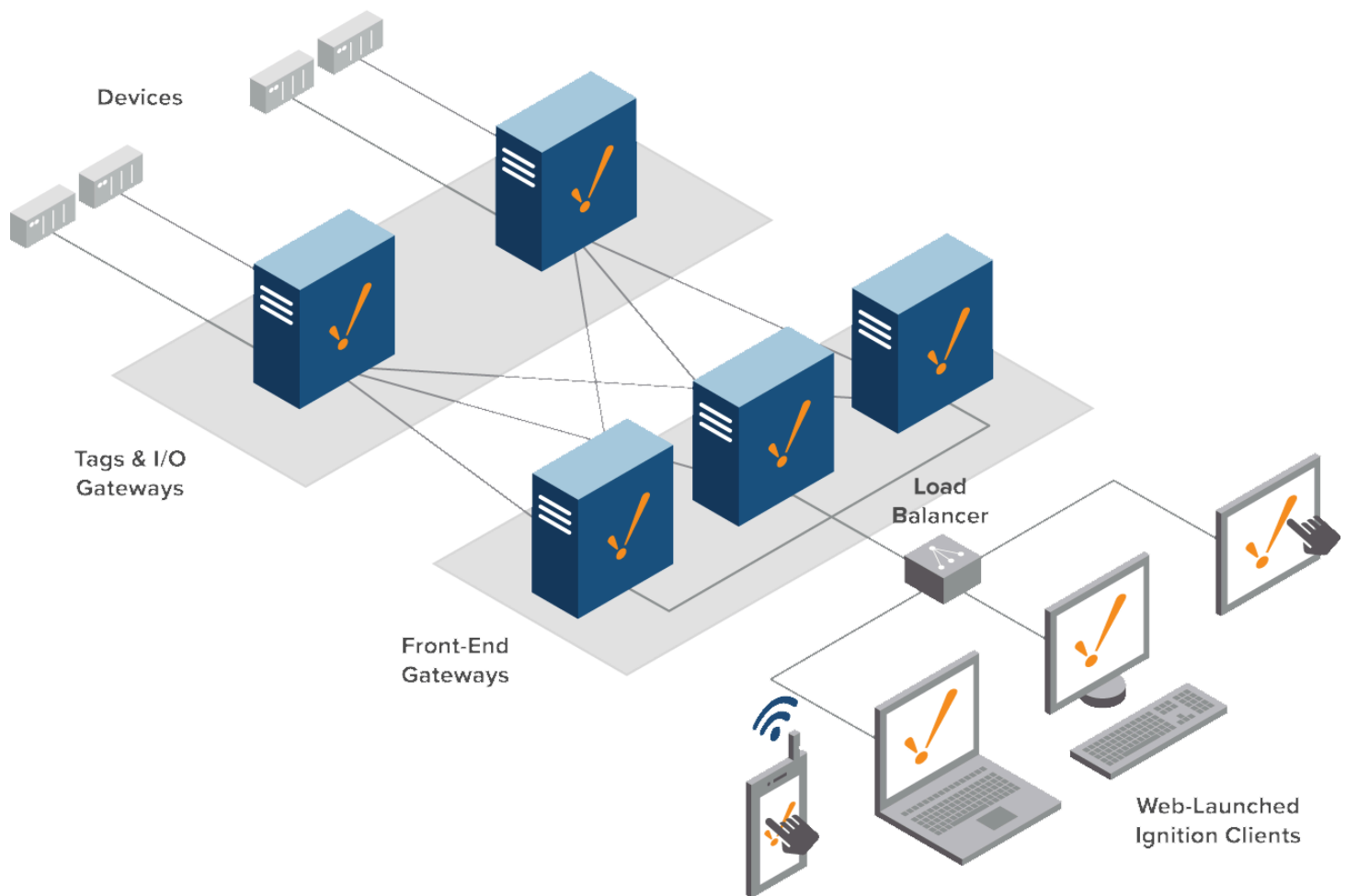
In larger systems, it is often easier to have multiple Ignition installations to help split the load between the front-end tasks and the back-end tasks. This is perfect for single large systems that aren't split up into different sites. In those cases, the [Hub and Spoke Architecture](#) is usually a better fit.

In the Scale Out Architecture, we have at least one Gateway that handles back-end communications. The back-end Gateway deals with all PLC and device communications. The front-end Gateway handles all of the Clients, serving up the data pulled from the back-end Gateway. This is made possible through the [Gateway Network](#), connecting Gateways to each other, and allowing Tags to be shared through remote Tag providers.



Load Balancing

The best thing about the Scale Out Architecture is that it is easy to scale up Ignition as your system grows. In the image below, we added more front-end Gateways to help handle an increase in clients, and a Load Balancer to automatically distribute the clients between them.



NOTE

New in 7.9.15 Launching clients behind a load balancer works as of 7.9.15, but the projects on each Gateway MUST have identical UUIDs, meaning that all visualization gateways must be restored from the same Gateway backup (GWBK), as project imports won't work.

Gateway Types

When utilizing Scale Out, an Ignition Gateway is given focus due to its placement. There are two possible types:

- **Tags & I/O:** Also known as a **Back-End Gateway**, is responsible for communicating with the PLCs, and executing Tags. They share their local Tags with the Front-End Gateways via a Remote Tag Provider, and are generally responsible for recording history. Additionally, alarms can be added to Tags locally, and exposed elsewhere in the system.
- **Front-End Gateway:** Sometimes called a **Visualization Gateway**, this Gateway is responsible for hosting the clients, allowing the Back-End Gateways to focus solely on PLC communication and history.

Database Placement

With this type of architecture, both types of Gateways will need database access: the Tag & I/O Gateways need to be able to store data, while the Front-End Gateways need to query the data out.

Scaling Out the Databases

Multiple on-site databases may also be utilized with this architecture. This typically involves configuring a database cluster.

Why Use the Scale Out Architecture?

Scale Out is utilized in large systems to distribute the workload across multiple servers. This is in contrast to a 'Scale Up' ideology where you have a single powerful server manage the whole system.

Decentralized Servers

In this architecture, each server only deals with a small part of the overall system. Should a server fault, only that part of the system is hindered, allowing other parts in organization to continue uninterrupted. Applying Redundancy to the Tag & I/O servers is ideal to maximize uptime.

Pooled Resource Usage

Since multiple servers are working together bottlenecks are reduced. A massive amount of work can be performed in this system without slowing down other servers.

Easy Growth

Improving performance in this architecture is simple: add another Gateway where appropriate. Because the system is spread out on purpose, it's easy to determine which kind of Gateway to add. A Back-End Gateway would help if you're adding new PLCs, and a Front-End Gateway will help when more clients need to be active at a time.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Hub and Spoke Architecture



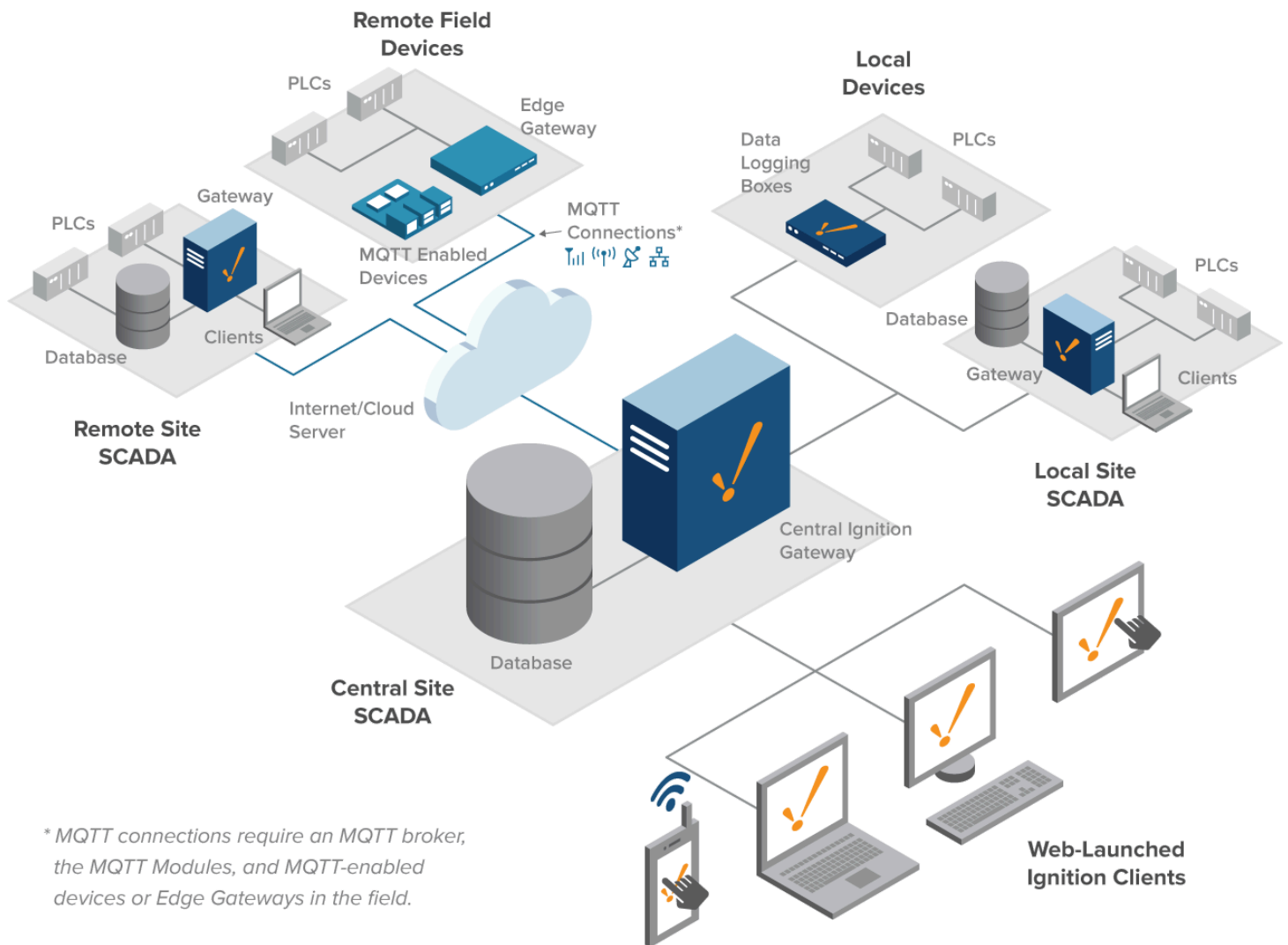
INDUCTIVE
UNIVERSITY

Hub and Spoke Architecture

[Watch the video](#)

Centrally Managed Systems

The Hub and Spoke Architecture involves multiple Ignition installations (the spokes) communicating and forwarding data points to a centralized Ignition installation (the Hub). The Hub and Spoke Architecture is unique in that you start with a single Ignition install at a central site. Ignition can then be installed at as many other remote or local sites as needed. These sites can then be customized or configured to suit that specific site's needs.



Reliable Remote Data Logging

Because connections between a database and remote PLCs can be unreliable, we can install either the **SQL Bridge** module or **Tag History** module. The **Store-and-Forward** system then ensures that data is never lost. The system may still be accessed through the OPC-UA server built into Ignition, or database-based Tags may be used to communicate current status. By making use of Ignition's Store-and-Forward system, data loss is prevented, ensuring that history always reaches its destination in the database.

The Hub and Spoke Architecture is ideal for centralizing historical data from multiple remote sites, and allows access to locally collected data via Ignition's **Gateway Network**.

Local and Remote Logging

The Hub and Spoke Architecture can be modified to store data at each spoke, in addition to the hub by splitting the Tag history. Redundant data is stored at each site as well as the hub database to protect against hard drive failure.

Local Client Fallback

Typically, the Hub and Spoke Architecture does not include the **Vision Module** at the remote sites. You can add the Vision Module at each spoke to take advantage of **local client fallback**. This way, each site can continue to operate at full efficiency in the event communication with the hub Gateway drops. Users at each spoke will never lost visibility of the system. **Vision Limited** licenses help lower costs on the spokes.

Remote Site Alarming

By adding the **Alarming Notification** module to a remote site, you can notify users when a problem occurs at each spoke. It is best to handle alarms at each remote site rather than centrally, to avoid any problems in case the connection to the central location is lost.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

IIoT Architecture



INDUCTIVE
UNIVERSITY

IIoT Architecture

[Watch the video](#)

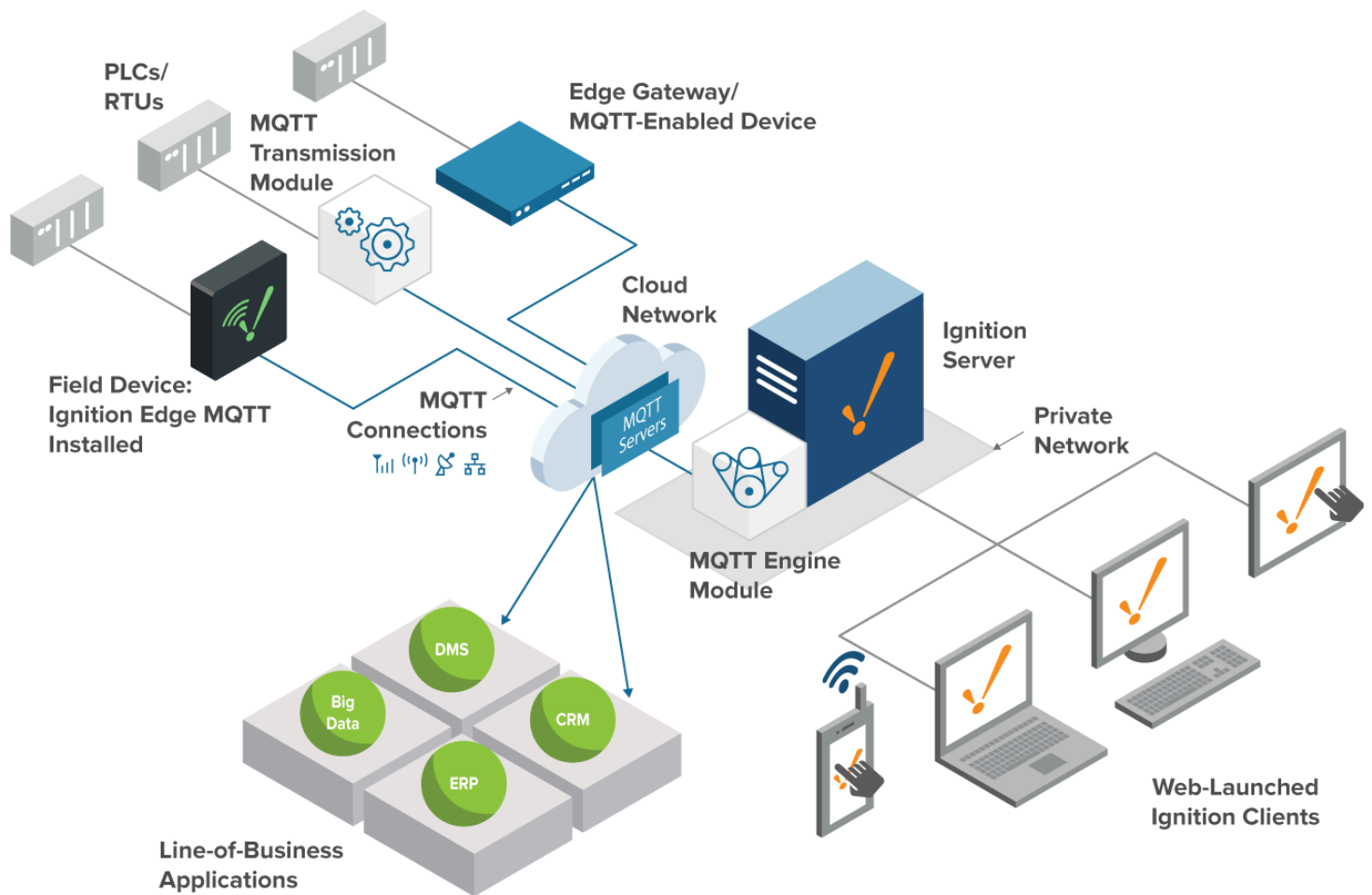
Overview

Ignition IIoT can collect data from any devices at the edge of the network, publish that data to a central broker, and push that data to subscribed industrial and line-of-business applications. Ignition IIoT can connect to PLCs in the field through the use of the MQTT Transmission module, field devices with Ignition Edge MQTT installed, and/or MQTT-enabled Edge Gateways and field devices that use the Cirrus Link Sparkplug MQTT specification. This data is published to an MQTT broker; the broker can be located on-premise, in the cloud, or a hybrid of the two. The MQTT Engine module located on an Ignition Gateway can subscribe to any data published from the broker. This data can be used in any Ignition application.

IIoT Architecture Example

IIoT Architecture typically involves the use of the MQTT protocol to communicate with a large number of devices. MQTT is a lightweight and secure protocol that makes use of a unique publish/subscribe transportation method. Additional information on the MQTT protocol and modules can be found in the [Cirrus Link documentation](#).

The [Standard Architecture](#) can easily be expanded with the MQTT modules to collect data from numerous Edge of Network devices. Data collected by the MQTT Engine can be historized and presented in a client with the Vision module, or via reports with the Reporting module.



Gateway Network vs IloT Architecture

The Gateway Network can be utilized to collect Tags at remote sites and share them with a central Ignition Gateway. However, in some cases, it is preferable to use the IloT Architecture instead. Below are a couple reasons why.

Lightweight Protocol

IloT uses the MQTT protocol, which is incredibly lightweight and efficient, making it ideal in scenarios when connections to a remote PLC are achieved via satellite or cellular communication. Additionally, it can safely support a large number of devices without placing a large strain on the network.

Simple Scalability

Adding new nodes to IloT Architecture is simple and cost efficient. New Edge Gateways or MQTT-enabled devices can be deployed with minimal effort, while keeping costs low.

Absolutely not! The two protocols will continue to coexist, as they currently do. MQTT is a great light-weight protocol that is ideal for communicating with devices when available bandwidth is limited. If your Gateway is attempting to communicate with a PLC over a cellular or satellite connection, then an MQTT solution may be a better fit than OPC-UA.

IloT Architecture Objects

The following are key pieces in the IloT Architecture.

Field Device

A Field Device of some sort can be used to connect to a PLC. This device also needs to be able to publish Tag values to an MQTT server. Any of the following hardware can act as a Field Device:

- Ignition Edge with the MQTT plugin
- Ignition server with the MQTT Transmitter Module
- Third-party MQTT enabled device

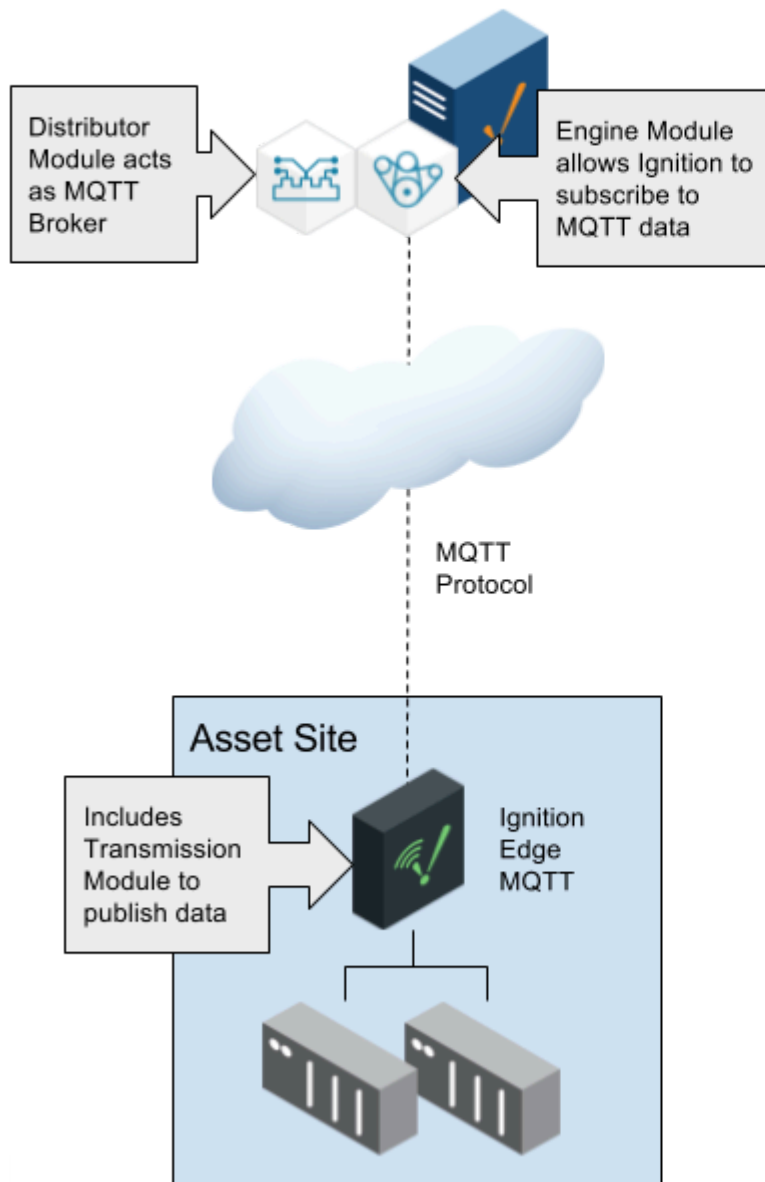
MQTT Server (Broker)

An MQTT Broker is a server that can communicate with your Field Devices via MQTT. This can be an Ignition Gateway with the MQTT Distributor Module installed, or a third-party server, such as a [Chariot MQTT Server](#).

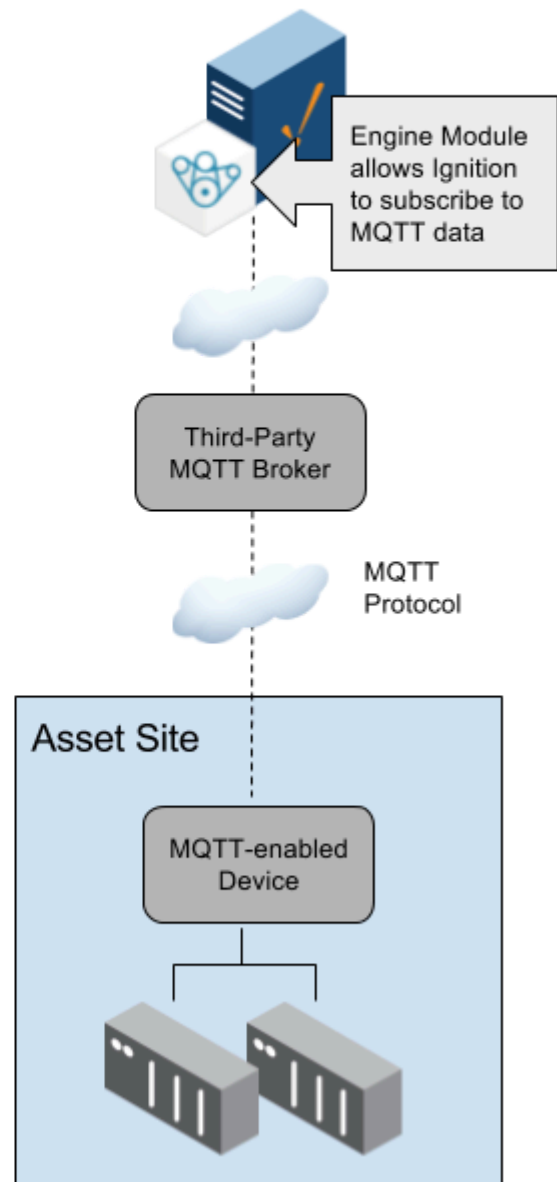
Subscriber

A subscriber is some application that subscribes to topics in the Broker. In regards to Ignition, this is the Gateway with the MQTT Engine Module installed. More details on the MQTT modules can be found in the [Cirrus Link documentation](#).

Architecture with Ignition MQTT



Architecture with Third-Party MQTT



This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Enterprise Architecture



INDUCTIVE
UNIVERSITY

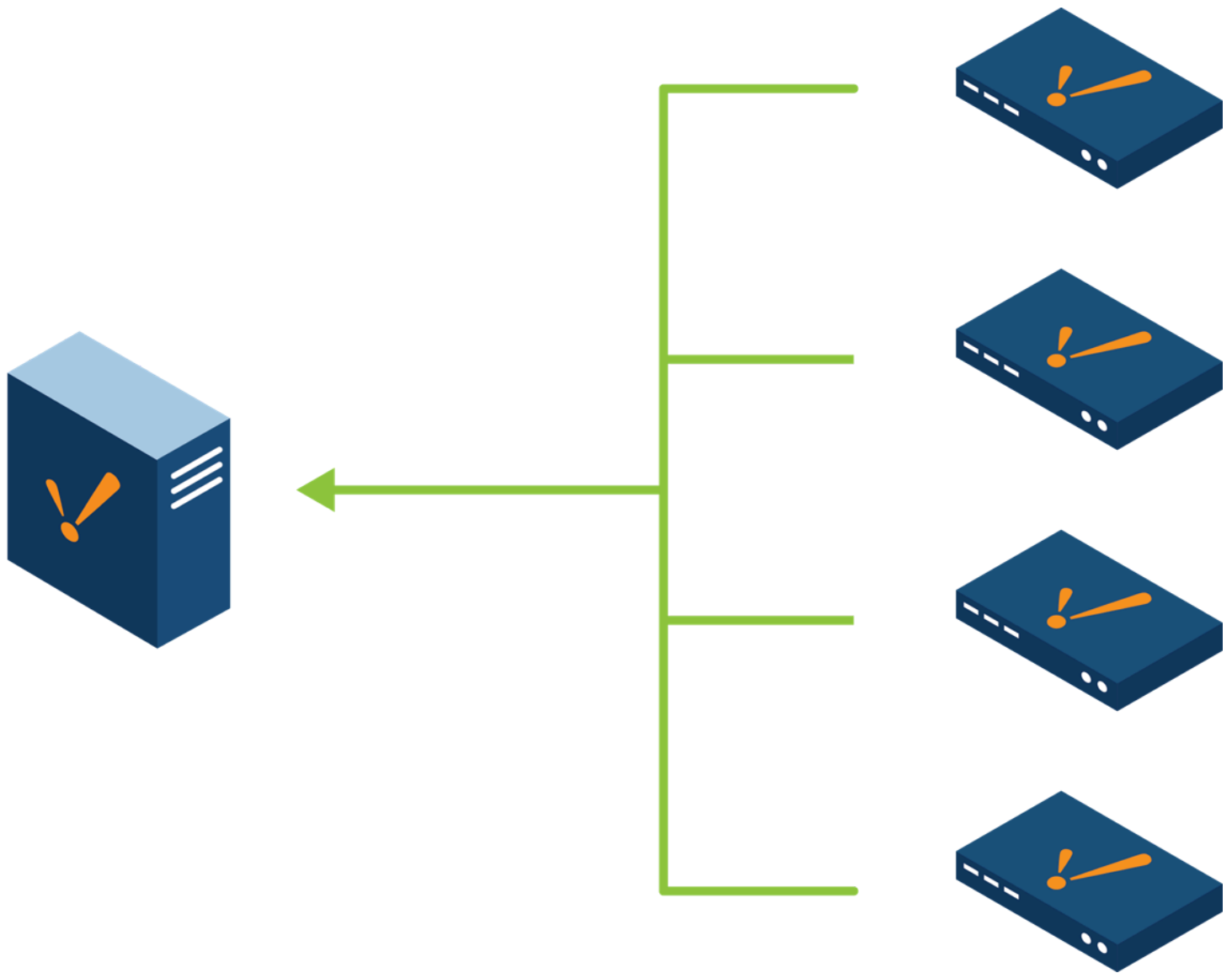
Enterprise Architecture

[Watch the video](#)

Overview

Many types of architecture involve multiple Ignition servers. Eventually, some assistance overseeing each server can be helpful. Enterprise Architecture typically involves having a mechanism to monitor and manage multiple Ignition Gateways. This involves utilizing the [Enterprise Administration Module](#) (EAM).

EAM specifies one Gateway as the Controller, and the others as Agents. Connections between these Gateways take advantage of the [Gateway Network](#), which is a secure method multiple Gateways can use to share information.



Why Use Enterprise Architecture

System Management

Monitoring the health of several systems can be time consuming. EAM streamlines system management by having each Agent Gateway report to a single Controller Gateway. This Controller can then generate alarms when problems occur, or assist in recovering from a hardware failure. Agent events can be stored in a SQL database, and easily incorporated into an Ignition Report. The best part is that the EAM can easily be incorporated into any architecture where multiple Gateways are present.

Project Synchronization

When multiple Gateways are used to launch clients, such as in the [Hub and Spoke Architecture](#), standardizing the look and feel of projects across both Gateways is important for a consistent

organization. With EAM, the Controller can synchronize resources across multiple Agents, allowing development to occur on one Gateway, and updates pushed out to other Agent automatically.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Edge Architectures



INDUCTIVE
UNIVERSITY

Edge Architecture

[Watch the video](#)

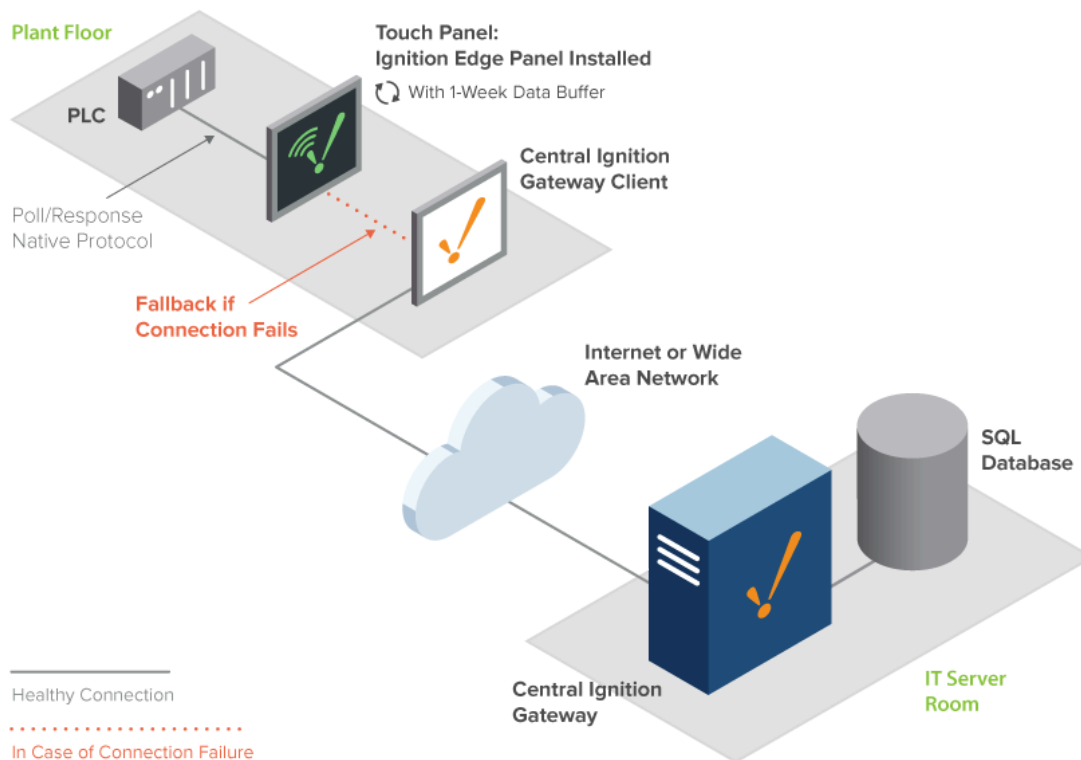
Overview

The Ignition Edge system is unique in that it can be very flexible and fill in the gaps in any system architecture. Ignition Edge is a smaller version of the standard Ignition install. Edge is used to sit at the edge of a network and do a few specific tasks like client failover or push MQTT data.

Each Edge plugin (Panel Plugin, Enterprise Plugin, and MQTT Plugin) offers specific functionality that works best in certain scenarios, but because any combination of plugins can be installed, an Edge Gateway may fill multiple roles and help bridge two different architectures. The diagrams below offer a look at each plugin's distinct functionality, but they can be combined to form an architecture to fit any situation. Additionally, these architecture diagrams show an Edge Gateway paired with an Ignition Gateway, which shows how an Edge system architecture can be added on to any of the other Ignition architectures.

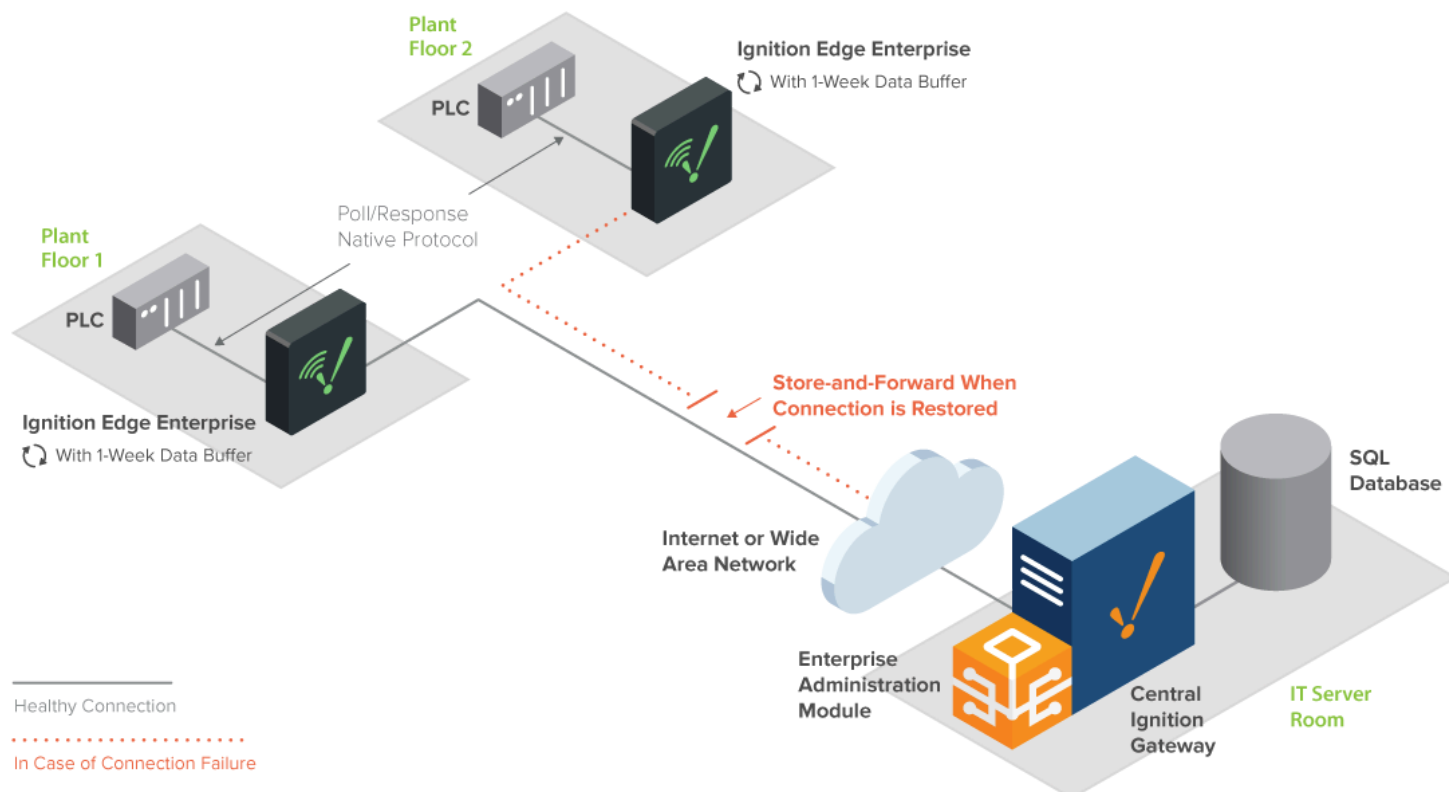
Panel Plugin

Edge with the Panel plugin can act as a stand-alone panel by simply connecting to a local PLC and building a basic screen. Paired with a central Ignition Gateway, the Edge can also act as a local client fallback, providing control on site in the event of a lost network connection.



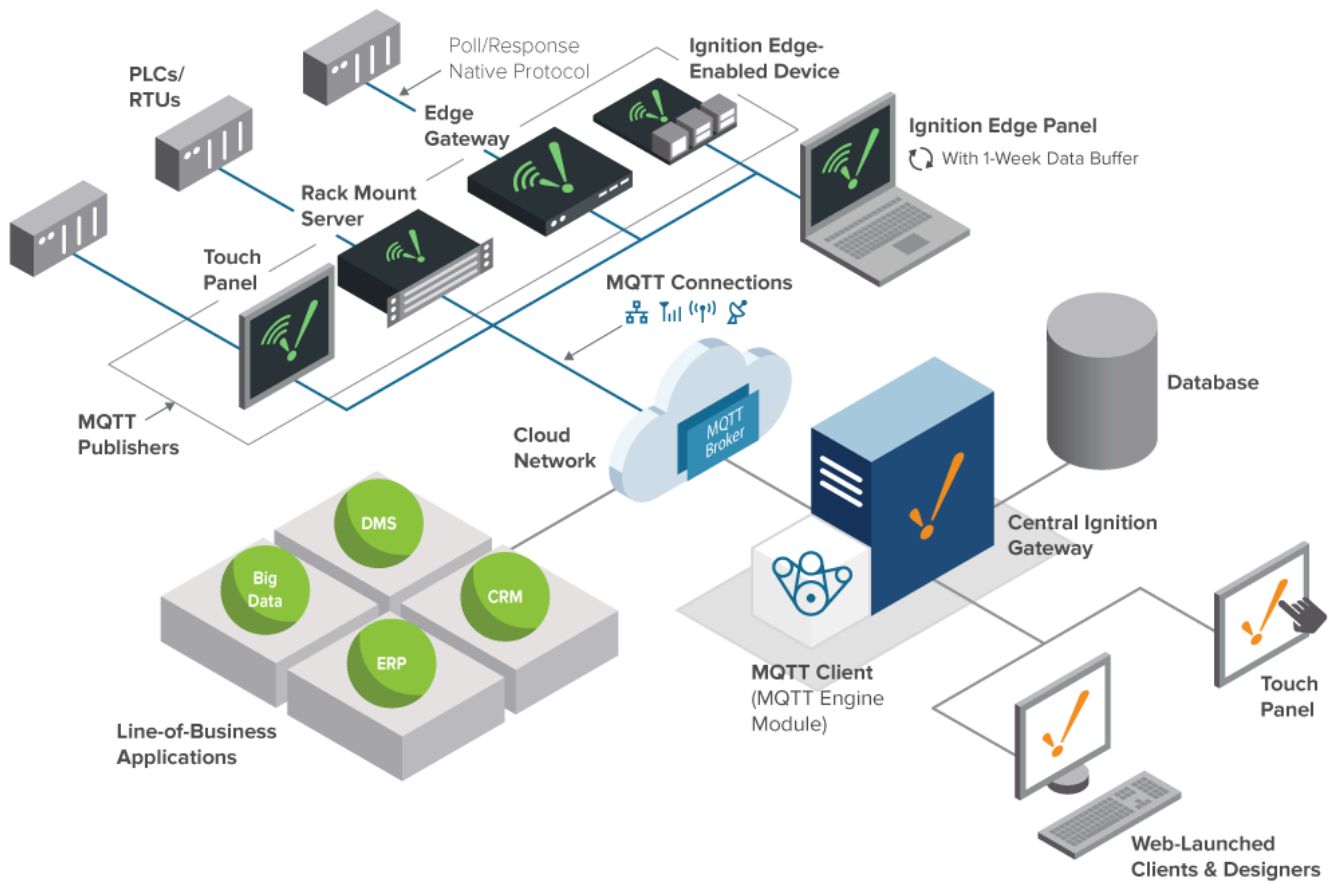
Enterprise Plugin

Edge with the Enterprise plugin provides remote monitoring and control of multiple sites from a central Ignition Gateway. This allows you to pull in Tag and history values to the central site, while the Edge system stores things locally in the event that the network connection is lost.



MQTT Plugin

Edge with the MQTT plugin provides a simple way of setting up an MQTT Publisher next to a PLC, allowing it to connect to a greater IIoT architecture. Here you can see multiple types of devices using Ignition Edge to connect in the network.



This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Redundancy Architecture



INDUCTIVE
UNIVERSITY

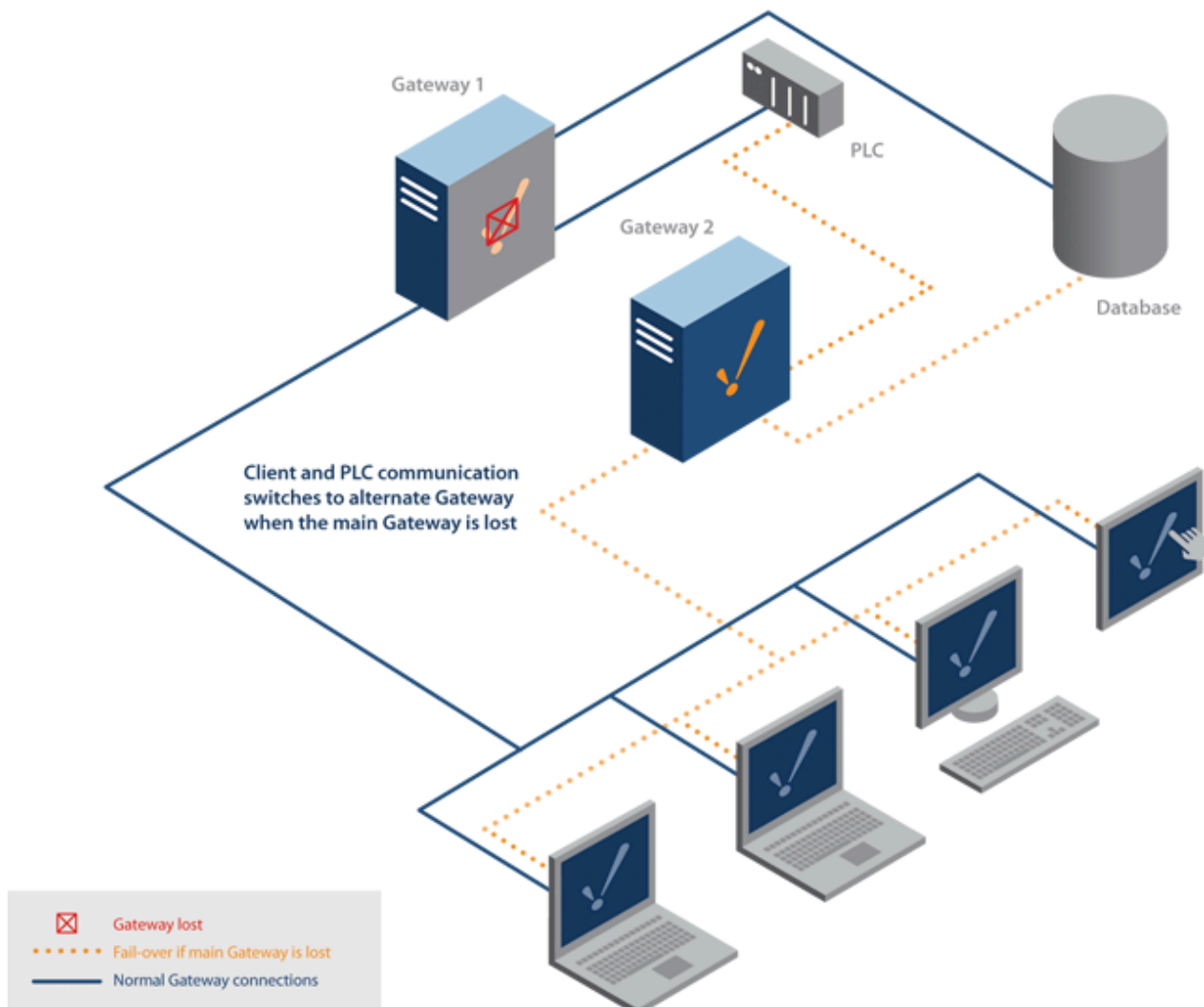
Redundancy Architecture

[Watch the video](#)

Mission Critical - Gateway Redundancy

Using **Redundancy**, two Ignition installations can be linked together, so if one server fails, the other takes over and continues executing. All of the clients connected will be redirected to the backup machine, and historical data will continue to be logged. The transition is seamless, meaning processes will never be prevented from executing due to one server going down.

Any single installation in the System Architectures can be replaced with a redundant pair of Ignition Gateways. If you have a dedicated backup computer, Inductive Automation offers a cheaper Backup-specific license.



Redundancy and Ignition in the Cloud

When Ignition is hosted on in the cloud, most cloud providers offer **built-in redundancy**, meaning a new instance is created in the event the original fails. As a result, you do not need to purchase a backup server for any primary servers that are in the cloud.

Redundancy Architecture Examples

In this section, we will take a look at several other types of architectures that have been enhanced by redundancy.

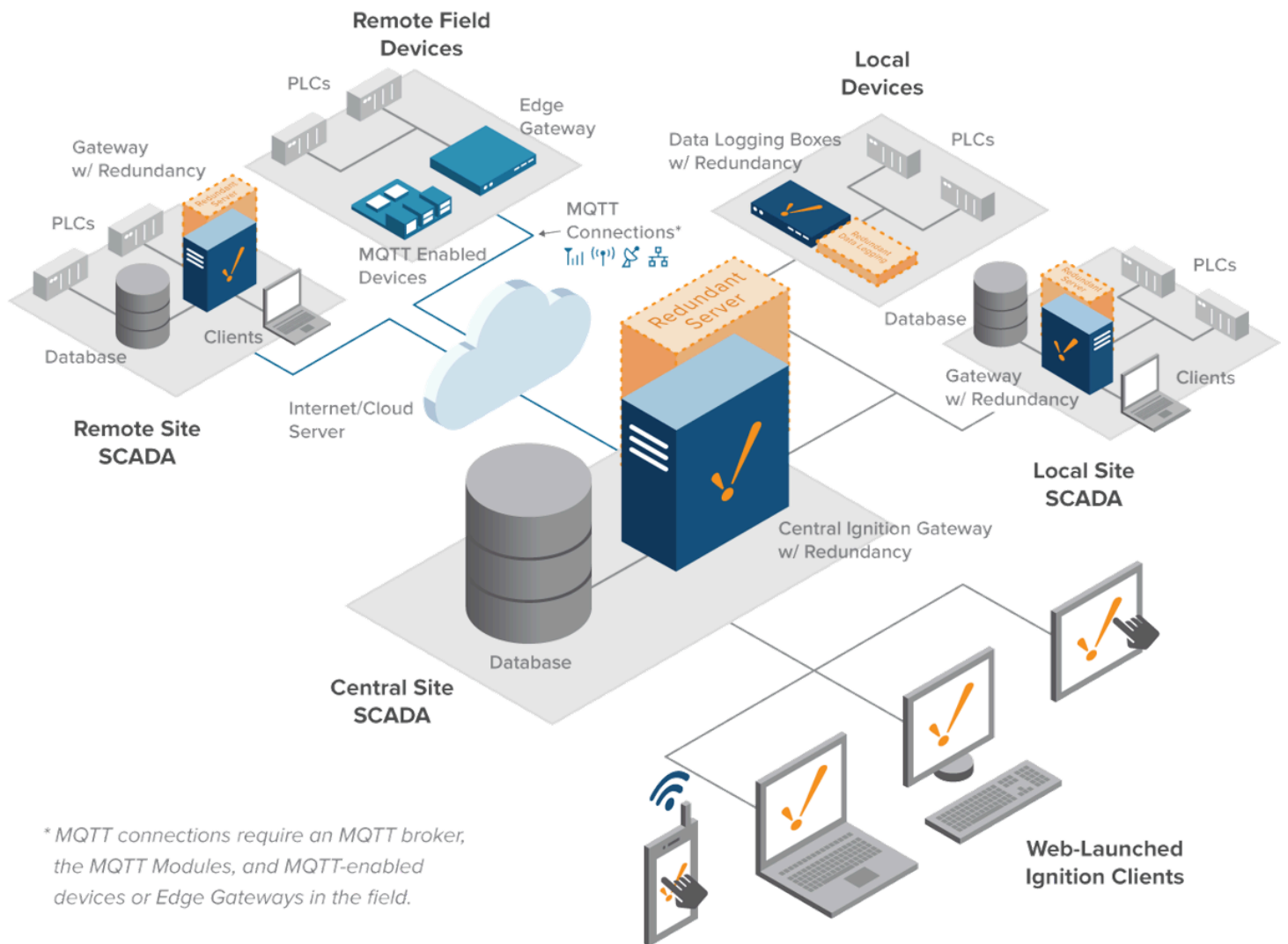
Standard Architecture

Reliability in the **Standard Architecture** can dramatically be improved by adding a redundant server. Should the primary server fail, the redundant server can seamlessly take over, allowing production to continue while the primary is brought back online.



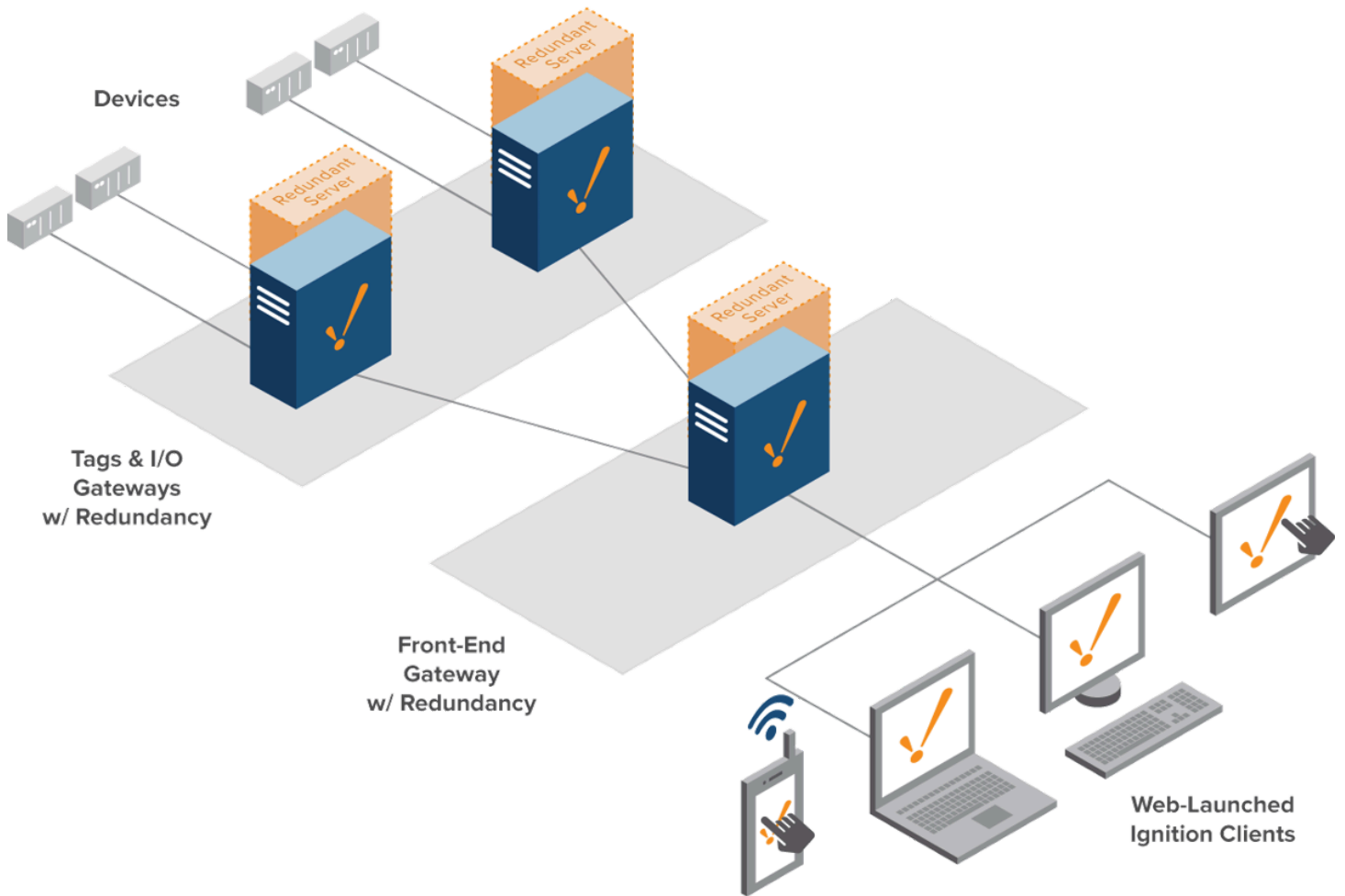
Hub and Spoke

The **Spokes** that contain an Ignition Gateway can be thought of as individual instances of the Standard Architecture. Incorporating a backup server at each of these Spokes will ensure uptime is maintained locally. The Hub can also utilize a backup server, which is especially useful if clients are also launched from it.



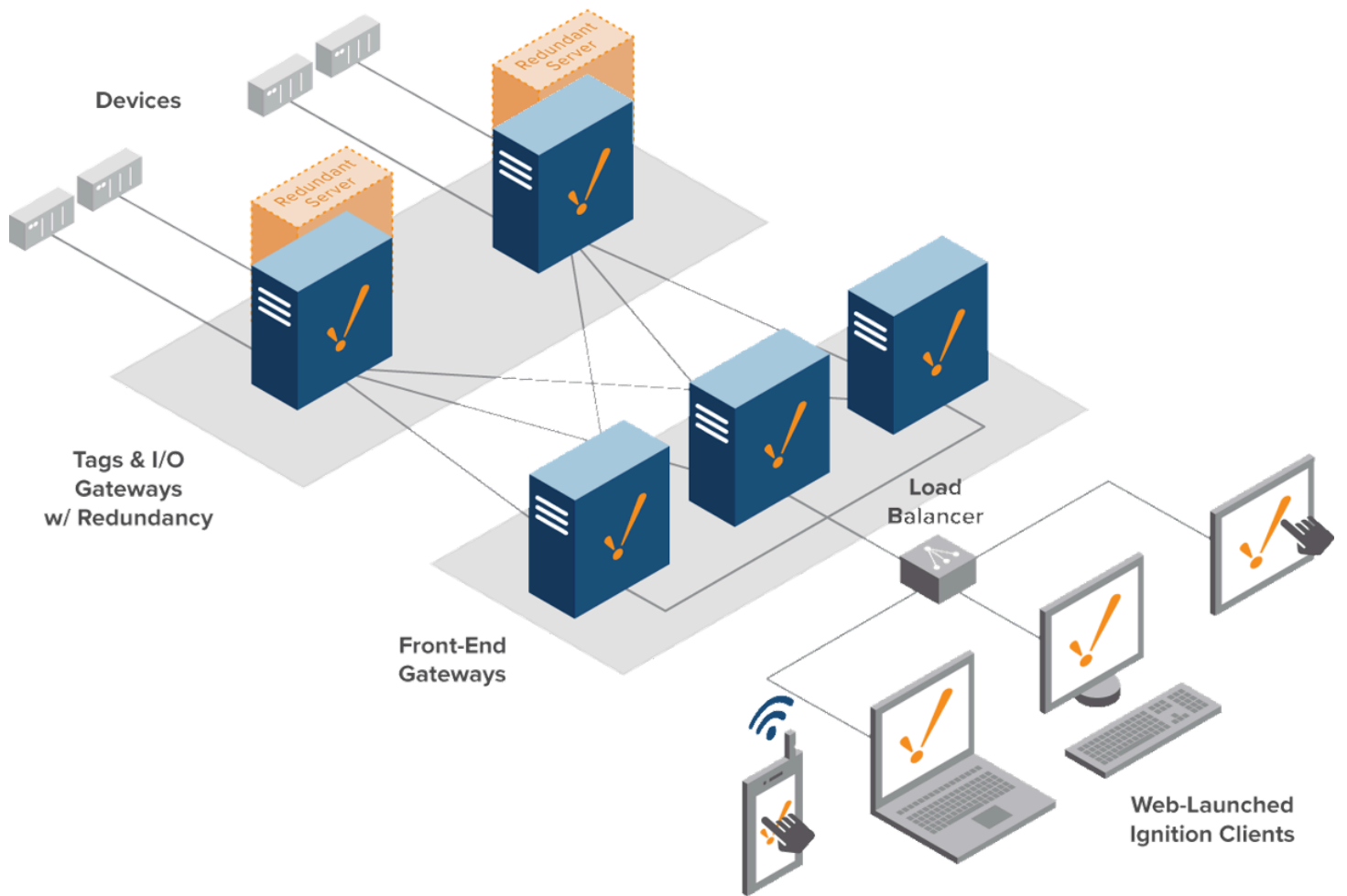
Scale Out Architecture

A server fault using the **Scale Out Architecture**, while not fatal to the whole system, can cripple the system. Without a Load Balancer, a Backup Gateway should be utilized at each Gateway. These means each Gateway has a failsafe ready to take over in the event of a failure.



Scale Out Architecture with a Load Balancer

When utilizing a **Load Balancer**, redundancy should not be applied to the Front-End Gateways: the Load Balancer is responsible for distributing the workload amongst the front-end, so a backup Gateway is not required. However, an extra Front-End Gateway could be incorporated to act as a failover. If you plan on having three Front-End Gateways active at any time, then a fourth Gateway should be available in case one server faults.



This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Cloud Based Architecture



INDUCTIVE
UNIVERSITY

Cloud Based Architecture

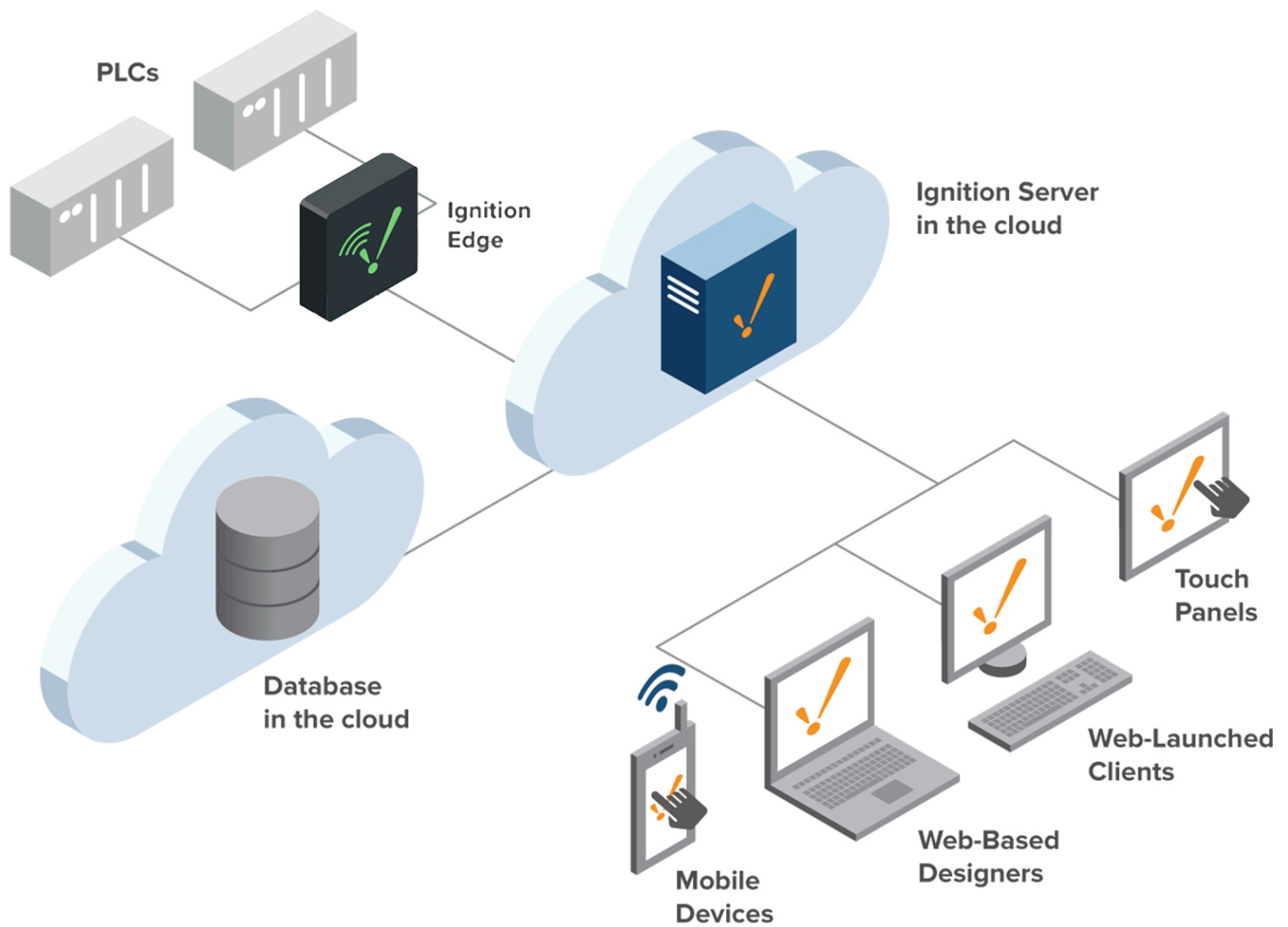
[Watch the video](#)

Hosted Ignition Server

Ignition servers can be hosted in the Cloud. Inductive Automation does not provide cloud services, but there are many great providers out there, such as Amazon EC2, or Windows Azure. The server can connect to the PLCs at the remote or customer sites via Round Robin Poll, Cell Tower, Satellite, or Secure VPN. The clients are connected to the Ignition server via Internet. Liberal use of Ignition's security features protects each customer from accessing another customer's projects or data.

Architecture Example

In this example, we see the [Standard Architecture](#), but with the Ignition Server and Database both running on cloud servers. Note, that the Database and Ignition are on different cloud servers. This is because cloud service providers generally have specialized services that are designed explicitly for hosting SQL databases, as well as more general purpose services such as running Ignition. An installation of Ignition Edge is placed local to the PLCs, and is responsible for forwarding data to the cloud.



Why Use a Cloud Based Architecture?

This type of architecture is ideal in smaller organizations since you don't have to worry about maintaining a server. Additionally, there is no on-site IT staff required to monitor the system.

Does the Database Always Have to be in the Cloud?

If Ignition is hosted in the cloud, then it is very common to have your database also hosted in the cloud. However, many of our customers find that also having a controls database on-site to store temporary or in-progress data useful. This is great when you want to have data stored locally.

Are Cloud Based Servers Secure?

Cloud based servers are increasingly common, but it is important to educate your organization on the risks associated with them. If your organization is interested in a cloud based solution, we recommend following the guidelines in the [Ignition Security Hardening Guide](#), as it addresses many potential security issues for both cloud and non-cloud based servers. Additionally, it offers recommendations on how to better protect your system.

Should I have a Redundant Ignition Server in the Cloud?

If you are using redundancy or are concerned about uptime, you should be using Ignition redundancy regardless of whether your Ignition is in the cloud. While most cloud providers offer built-in redundancy, this is often much slower than Ignition's redundancy failover time. Typically Cloud Server redundancy has a new computer instance spun up (started) in the event the original fails. This will often take a couple minutes where Ignition's redundancy (with default settings) takes around 20 seconds to switch over. There are other options to make a Cloud Server start faster, but currently they all slow down the CPU substantially to make it possible.

What if I Require an On-Site Server with Redundancy?

If you are using Ignition's Redundancy with a Cloud Server, then typically we recommend that both of the servers are run side-by-side. This means either both in the cloud or both on site. One warning: If you have both instances of Ignition in the cloud but if your connection to the internet goes down, then you lose access to both instances of Ignition.

Cloud Service Suggestions

Many customers find the following cloud based service suggestions helpful. This is not an exhaustive list, there are many good options available.

Amazon Web Services

- Ignition Hosting: [AWS EC2 Instance types](#). The M3 and M4 instance types meet most customers needs in regards to hosting Ignition.
- Database Hosting: [AWS Relational Database Service](#).

Microsoft Azure

- Ignition Hosting: [Azure for Discrete Manufacturing](#)
- Database Hosting: [Azure SQL Database](#)

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

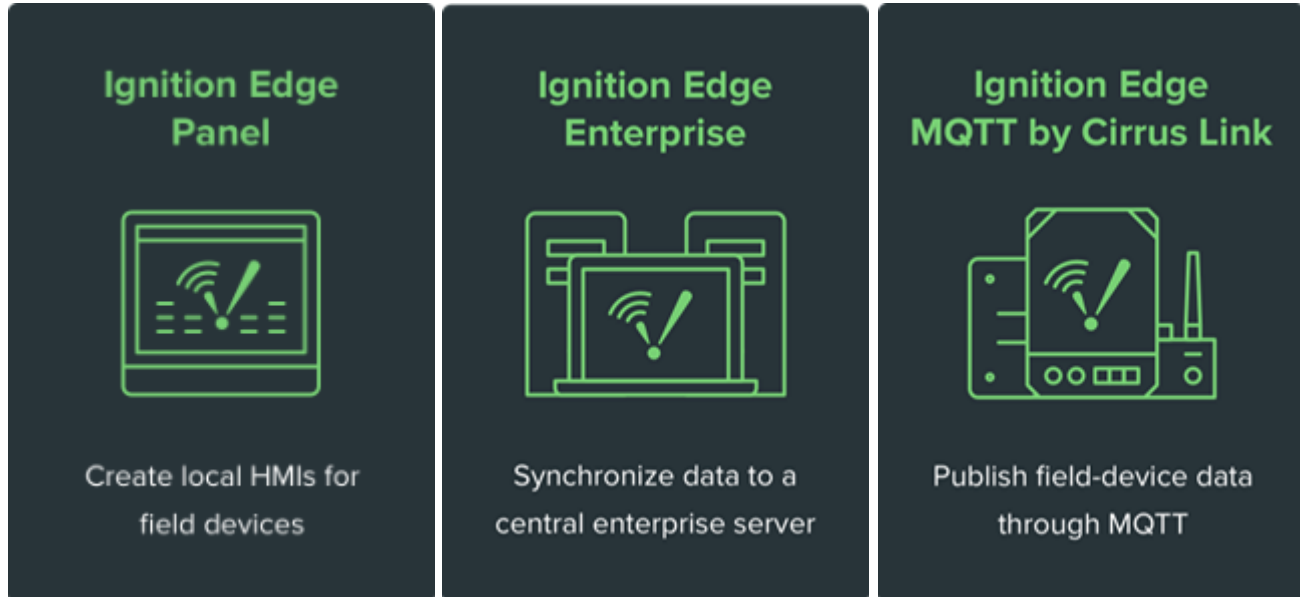
Ignition Edge

New in 7.9.2

Welcome to Edge

Edge is a new, stripped down version of Ignition that is specifically designed for use as an Edge of Network device.

There are currently three Ignition Edge plugins that can be used: **Panel**, **Enterprise**, and **MQTT**. Each offers its own features and you can combine them however you'd like to create the system you need. Often, Panel and Enterprise are used together to create a simple way to keep local fallback terminals up to date.



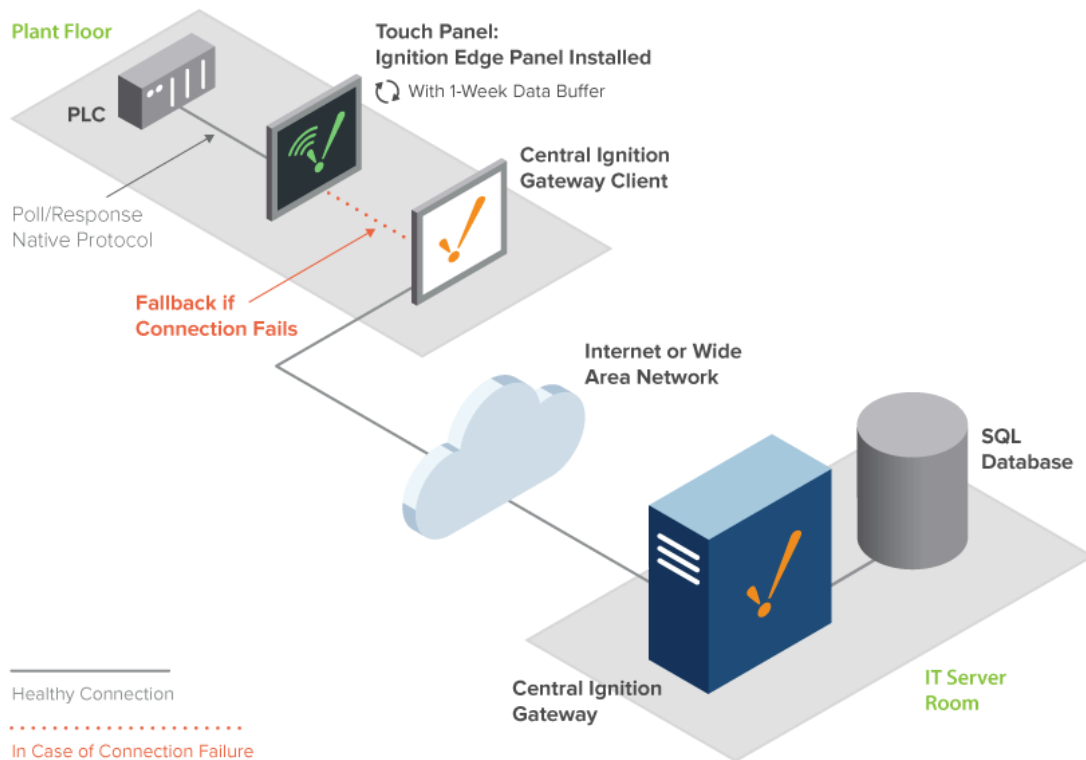
Panel

Ignition Edge Panel enables edge-of-network-HMI functionality. This includes:

- One local client
- One remote client

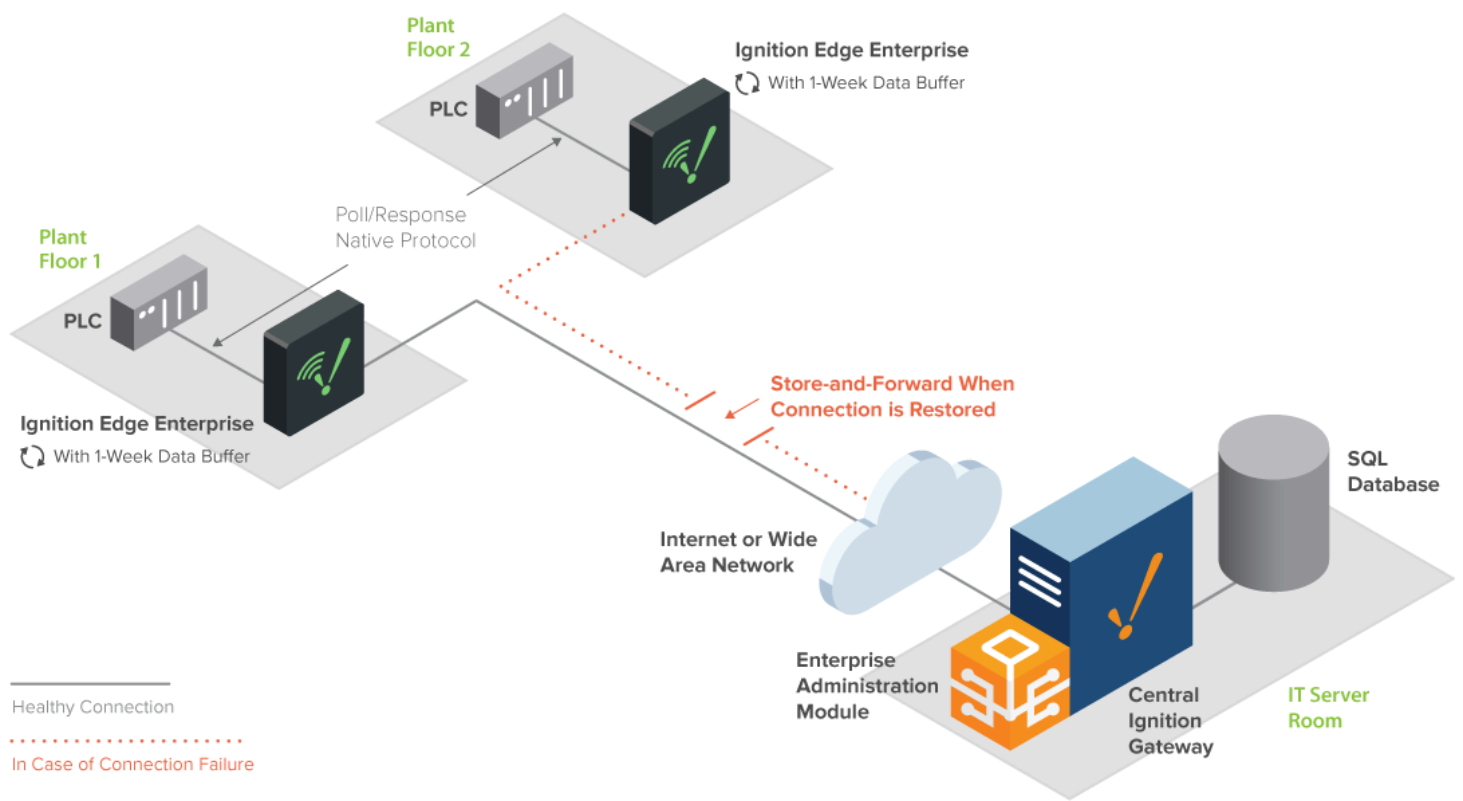
- Alarming features, including one-way email notification

Ignition Edge Panel includes one week of data buffering for trending and local client fallback for mission-critical applications.



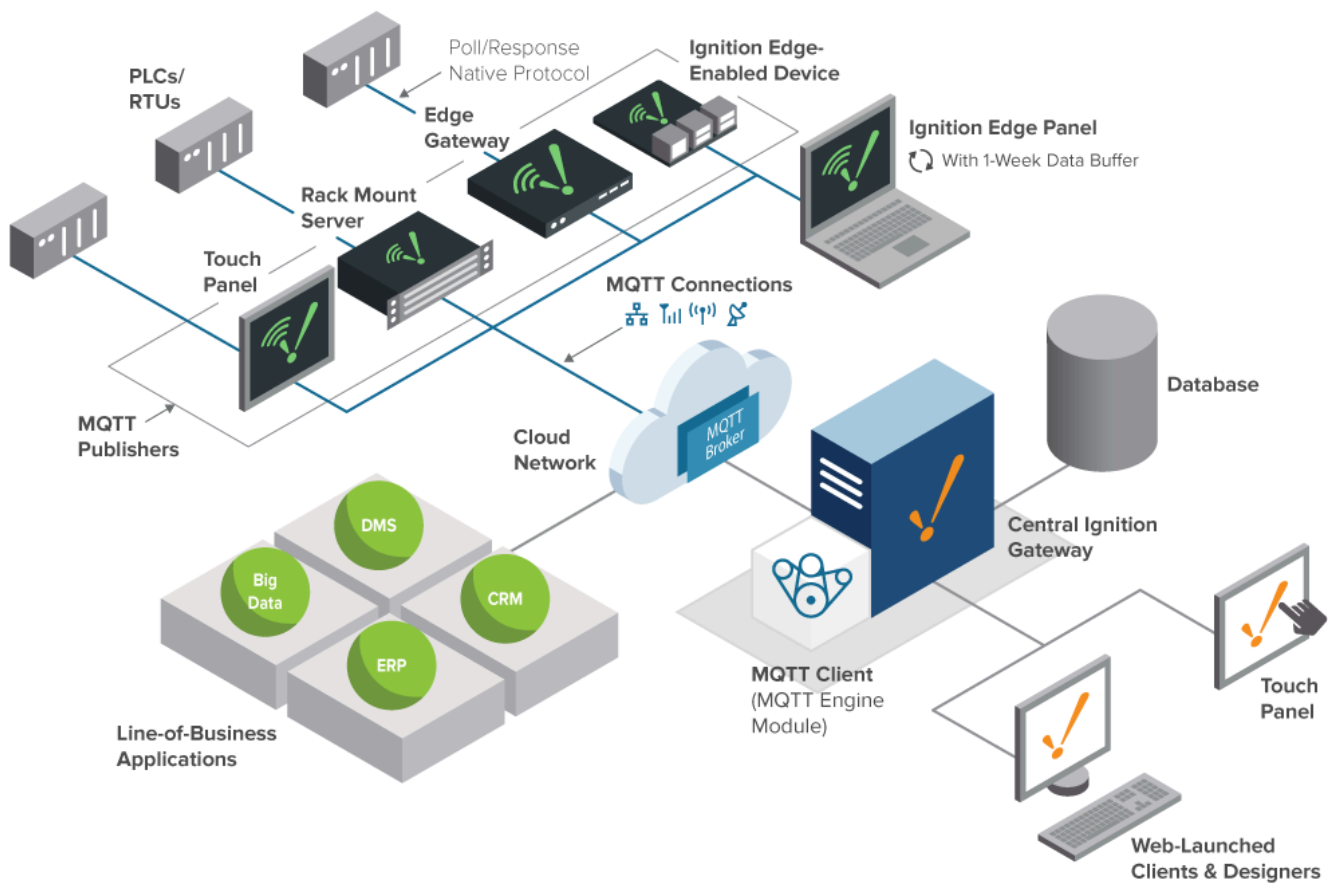
Enterprise

Ignition Edge Enterprise acts as a limited remote server that synchronizes data from the edge of the network to a central Ignition server. It has Distributed Tags, history, and remote alarming built in. Remote backup, restoration management, and centralized monitoring of performance and health metrics are also available if the Ignition Enterprise Administration Module (EAM) is installed on the central Ignition server. Edge Enterprise comes with up to one week of data buffering so it can be used to store-and-forward your data to a central server in case of network failure.



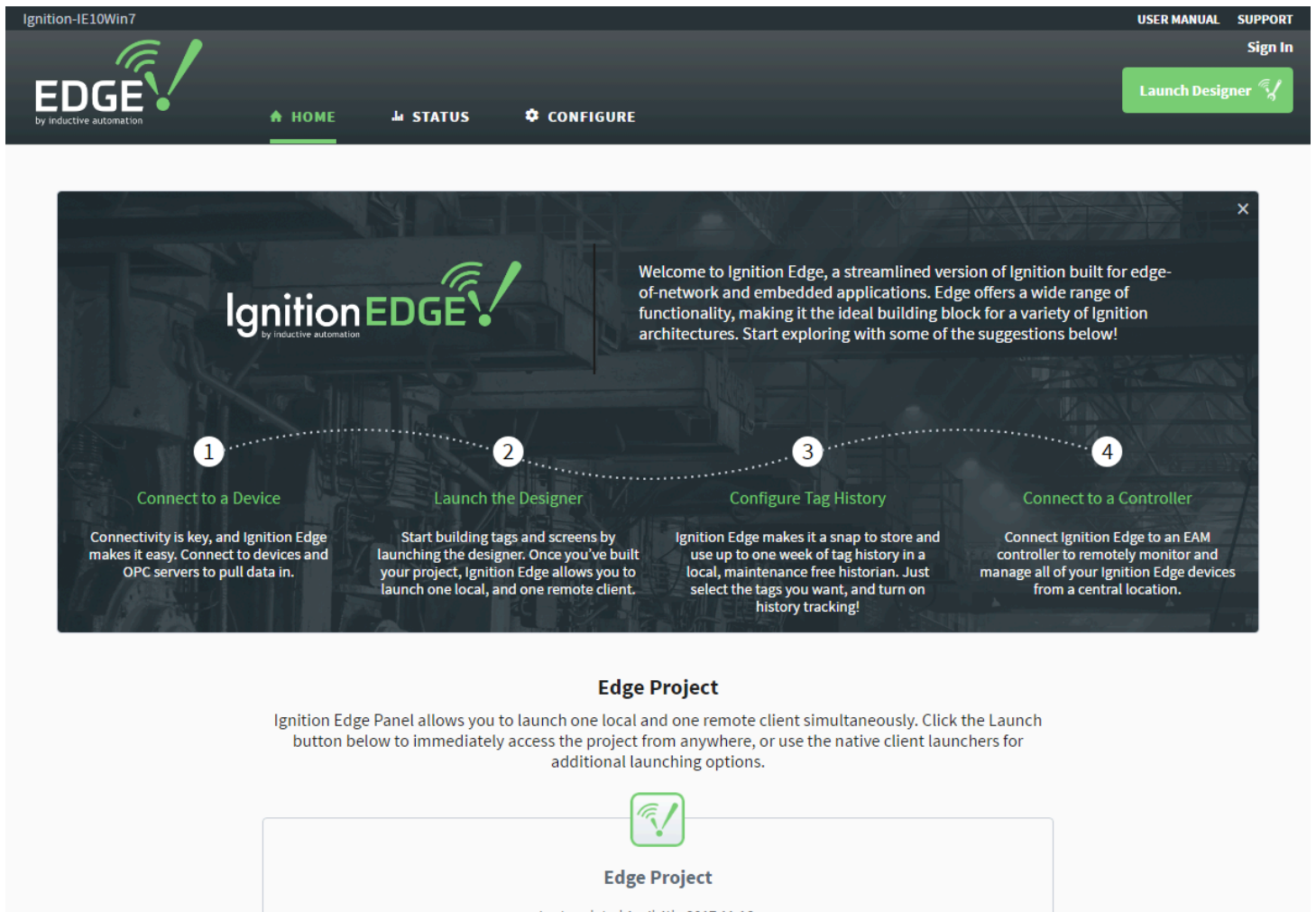
MQTT

Ignition Edge MQTT by Cirrus Link turns virtually any field device such as a touch panel or a client terminal into a lightweight, MQTT-enabled edge gateway that works seamlessly with Ignition IIoT. Ignition Edge MQTT uses the highly efficient MQTT protocol to transmit data to any MQTT broker and supports the Sparkplug data-encoding specification.



What's Different About an Edge Gateway?

All Edge installations are a limited version of the Ignition platform for a significantly reduced cost. This helps keep things simple, and these installations can always work with other full versions of Ignition if you need additional functionality. There are a lot of visual changes to the Gateway Webpage, and many pages have been removed to avoid confusion. You can see below that after installing Edge, the traditional orange Ignition logos are replaced with new green Edge logos so a gateway can be easily identified.



Licensing

The new Edge version of Ignition has a separate license that works in the same way as a traditional Ignition license. The Edge license will not work if you attempt to activate it on an Ignition Gateway as it is only setup to license an Edge Gateway and its modules. However, you can activate a full Ignition license on an Edge Gateway, but this is not recommended due to the limitations of the Edge platform not being able to fully utilize the unlimited nature of an Ignition Gateway. Additionally, once a license has been activated on an Edge Gateway, the trial time for any unlicensed Edge plugins will be removed, and you will not be able to access them.

Projects

Edge installations only ever have one project. It is automatically created if it did not exist, and cannot be removed.

Database Access

Database access is **disabled**. This means that any database queries or bindings will not work, and that the user sources that require a database connection will not be available. The Status and

Configure pages of the Gateway Webpage have been updated to remove these features and reduce confusion. Despite this restriction, some "historical" storage can happen inside of Ignition.

One Week of Internal Storage

- **Tag History** - Up to one week or 10 million data points of history can be stored locally. Pruning will automatically happen, and the historian is automatically enabled and not configurable.
- **Alarm Journal** - The alarm journal will store all alarm events locally for one week of events. There are limited configuration options.
- **Audit Logs** - Like the Alarms, the Audit log is limited to one week of local storage. It can't be configured.

Gateway Scripting

All Gateway level scripting is **disabled**, including Gateway Event Scripts and Tag Event Scripts. This does not affect project or client scoped scripts.

Gateway and Project Backups

Making a Gateway backup in an Edge installation will work perfectly in other Edge installations. Edge Gateway Backups are not compatible with Ignition Gateways, so you may not take an Edge Gateway Backup and restore it on an Ignition Gateway. You may also not take an Ignition Gateway Backup and restore it on an Edge Gateway.

However, you will still be able to export Tags, Windows, and other project resources in the Designer from full Ignition Gateways, and import them into an Edge project, or vice versa.

Tag Limit

There is a strict 500 Tag limit in an Edge Gateway. Any Tags added to the Tag Browser above that limit will not work until others are removed. Folders, UDT Definitions, UDT Instance Tags, and Client Tags are not included in that Tag count.

Changed in 7.9.9 As of Ignition 7.9.9, Edge no longer has a tag limit. Older versions of Edge can simply upgrade to 7.9.9+ to remove the tag limit.

OPC Limitations

The internal OPC-UA server in the Edge Gateway acts only as a client and not as a server. This means Edge **can** connect to other outside OPC servers as a client to fetch data, but **can not** act as an OPC-UA server for outside programs to connect to.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Installing Edge

New in 7.9.2

Downloading and installing Ignition Edge works in much the same way that installing Ignition on any operating system works. Ignition Edge uses the same installation methods that Ignition uses, so you simply need to download and install as normal. While Edge is a great companion for a minimal system such as an **ARM** device, it can be installed on just about any operating system.

NOTE

To install Edge on an ARM processor, you need the Edge-specific zip file named **Ignition-Edge-linux-armhf-x.x.x.zip** where x.x.x is the version number.

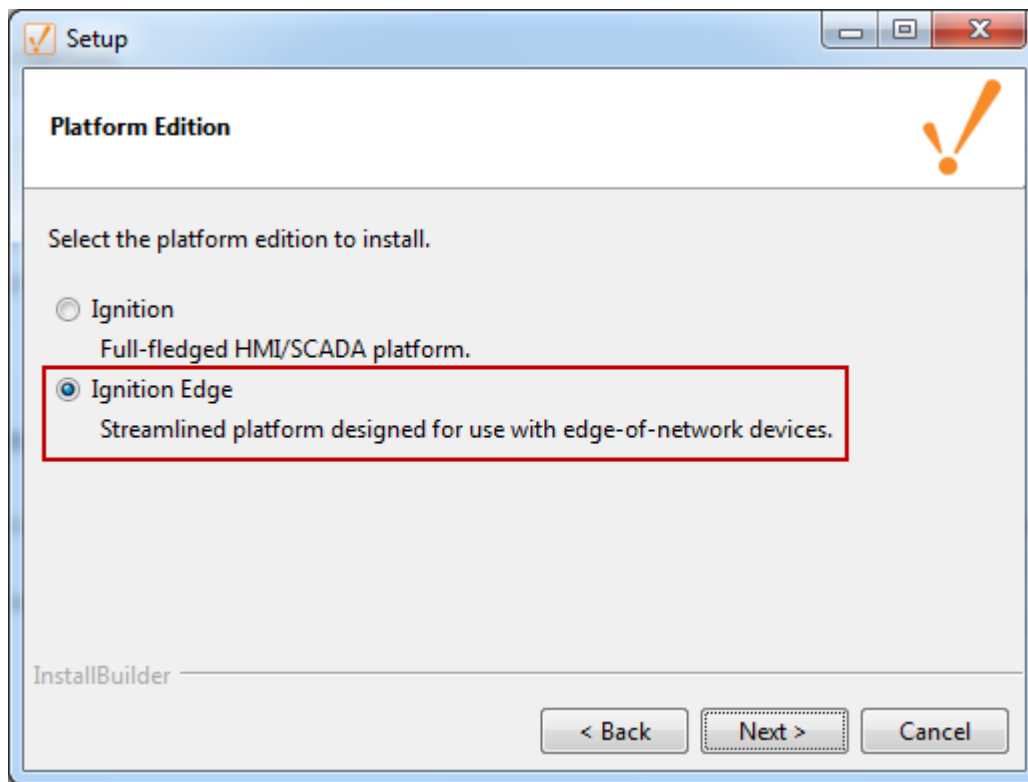
Using the Ignition Installer

On most full OS installations, you can use the same installer for Edge as the full version of Ignition. During the installer setup, the wizard will ask you to select which **Platform Edition** to install. Here you can select between a full Ignition install and an Edge install. We want to select the **Ignition Edge** option.

NOTE

The Installer will only ask for a Platform Edition if it is a fresh install. If Ignition is already installed, the installer will detect that, and upgrade instead. If you wish to install Edge on a computer that already has Ignition, you will need to first [uninstall Ignition](#), and then run the installer and select Edge.

Once you have selected an Edge installation, there is no option for a custom installation that allows you to select modules. When installing Edge, the appropriate modules will be added for you.



Further Install Instructions

You can look at the following install pages for more detailed information regarding installation on a specific platform.

- [Installing Ignition](#)
- [ARM - Install](#)

Gateway Webpage

Once the installation process is done and the Edge Gateway has finished starting, navigating to localhost:8088 in a web browser will take you to the Ignition Edge Gateway Webpage. You will notice the Gateway will be green instead of our usual Orange color, so you instantly know this is an Edge install instead of a full install.

Once Edge is installed, you can start using it. It has the same two hour trial as the full install of Ignition.

**Trial Version** 1:57:39 We're glad you're test driving our software. Have fun.**Activate Ignition**

Welcome to Ignition Edge, a streamlined version of Ignition built for edge-of-network and embedded applications. Edge offers a wide range of functionality, making it the ideal building block for a variety of Ignition architectures. Start exploring with some of the suggestions below!

- 1 Connect to a Device**
Connectivity is key, and Ignition Edge makes it easy. Connect to devices and OPC servers to pull data in.
- 2 Launch the Designer**
Start building tags and screens by launching the designer. Once you've built your project, Ignition Edge allows you to launch one local, and one remote client.
- 3 Configure Tag History**
Ignition Edge makes it a snap to store and use up to one week of tag history in a local, maintenance free historian. Just select the tags you want, and turn on history tracking!
- 4 Connect to a Controller**
Connect Ignition Edge to an EAM controller to remotely monitor and manage all of your Ignition Edge devices from a central location.

Edge Project

Ignition Edge Panel allows you to launch one local and one remote client simultaneously. Click the Launch button below to immediately access the project from anywhere, or use the native client launchers for additional launching options.



This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Edge Plugins

New in 7.9.2

What is a Plugin?

Ignition Edge offers different plugins that each add unique functionality to the Edge Gateway. There are three plugins available:

- Panel
- Enterprise
- MQTT

Each Edge installation must contain at least one plugin, but any combination of plugins can be used as well. Select the ones that you need and don't use the rest, just like the modules in a standard Ignition installation.

Panel

The Panel plugin is to provide a remote terminal for operators, usually to act as a backup in case there is a network failure. Panel provides up to two launchable clients for your Edge project, one local and one remote can be open at the same time. In addition, Panel provides alarm notification in a limited form. This means the Edge Gateway has only one email notification profile and no alarm pipelines.

Configuring Alarm Notification

Because of this alarm limitation, when navigating to the Alarm Notification page in the Gateway, you will be taken immediately to edit the Edge Notification Profile. In the settings, you can select an SMTP profile to use as well as a roster. Both your SMTP profiles and Rosters must be configured first, and the 'From' address of the notification email will be the email address that is used to log in to the SMTP server.

When a Tag goes into alarm, it will automatically go through this notification profile and be sent to all users in the selected roster. This alarm notification is one-way, meaning alarms can't be acknowledged from the notification email.

Ignition-IE10Win7

EDGE
by inductive automation

HOMESTATUSCONFIGURE

USER MANUALSUPPORT
adminSign Out
Launch Designer

Search...

SYSTEM

OverviewBackup/RestoreLicensingModulesProjectRedundancyGateway Settings

NETWORKING

Gateway NetworkEmail Settings

SECURITY

Audit Log ViewerUsers, RolesService SecuritySecurity Zones

ALARMING

JournalNotificationOn-Call RostersSchedules

OPC-UA SERVER

CertificatesDevicesSettings

OPC CONNECTIONS

Trial Version0:38:29We're glad you're test driving our software. Have fun.

Activate Ignition

Edge Notification Profile

Main

Enabled☐ (default: true)

Email Settings

SMTP Profile-None-

On-Call Roster-None-

Alarm Criteria

Minimum PriorityLow

Maximum PriorityCritical

Save Changes

Enterprise

The Enterprise plugin allows you to setup the Edge Gateway as an EAM agent. This allows you to remotely monitor and manage Edge Gateways from a central Ignition Gateway. Enterprise also includes the Gateway Network, bringing in all of the remote distributed services that come with that. However, Edge can only be the target of distributed services, meaning Ignition Gateways will have the ability to setup Remote History Provider connections to Edge, but Edge will not be able to setup a Remote History Provider connection to an Ignition Gateway. Other Ignition Gateways can use the distributed services to query Tag values, Tag history, and alarms from Edge Gateways with the Enterprise Plugin.

The Enterprise plugin introduces Tag History Synchronization. Tag History Synchronization allows you to sync your history data from the Edge Gateway to a full Tag History Provider on an Ignition gateway, allowing you to save past the one week data limit. The Edge Gateway will still only see the one week

of data that it can hold internally, but it will allow you to store the data in a more permanent database that can be viewed from other Ignition systems. This is setup by navigating to the Tag History Sync page link at the bottom of the Edge Gateway Configure page.

Overview

Backup/Restore

Licensing

Modules

Project

Redundancy

Gateway Settings

NETWORKING

Gateway Network

Email Settings

SECURITY

Audit Log Viewer

Users, Roles

Service Security

Security Zones

ALARMING

Journal

Notification

On-Call Rosters

Schedules

OPC-UA SERVER

Certificates

Devices

Settings

OPC CONNECTIONS

Servers

Quick Client

ENTERPRISE ADMINISTRATION

Agent Setup

ENTERPRISE SERVICES

Tag History Sync

Enterprise Tag History Synchronization

Settings

Remote Sync Enabled

☐

(default: false)

Remote Gateway Name

The gateway to target for remote synchronization. Must have the Tag Historian module installed, and allow remote storage.

Remote Provider Name

The remote history provider to sync data to.

Sync Schedule

Sync Frequency

The frequency with which data will be sent to the remote gateway. This setting will be used in conjunction with the sync schedule, if enabled.
(default: 10)

Sync Frequency Units

Seconds

▼

(default: SEC)

Max Batch Size

The maximum number of data points that will be sent per batch to the remote gateway.
(default: 10,000)

Enable Schedule

☐ If enabled, the data will only be synchronized during the times specified by the pattern provided.
(default: false)

Schedule Pattern

A comma separated list of time ranges. Examples:
9:00-15:00
9pm-5am
20.30-04.30

Here, you can setup the history sync to point to a Tag History Provider in another Gateway that is connected through the Gateway Network.

- Remote Sync Enabled - Allows you to turn Tag History Synchronization on or off.
- Remote Gateway Name - Here you specify the name of an Ignition Gateway that is connected through the Gateway Network. The Ignition Gateway must have the Tag Historian module installed and will need at least one Tag History provider. The Ignition Gateway's security settings will also need to be configure to allow remote storage.
- Remote Provider Name - The name of the Tag History Provider that the Edge Gateway's history data will be stored to. The Tag History Provider can be any type of history provider that the Ignition Gateway has.
- Sync Frequency - How often data will be sent to the Ignition Gateway based on the units specified in the Sync Frequency Units.
- Sync Frequency Units - The unit of time that will be used with the Sync Frequency.
- Max Batch Size - The maximum number of records that will be sent in a single batch to the Ignition Gateway.
- Enable Schedule - If enabled, data synchronization will only happen during the schedule specified. This allows you to limit Tag history traffic being sent over the network to a time when

network traffic is a little lighter.

- **Schedule Pattern** - A comma separated list of time ranges that will determine when data will be sent over the network.

Tag History Sync is unique in that if the connection to the remote Ignition Gateway is severed, the Edge Gateway will use its week of storage as a Store and Forward buffer, allowing you to store up to a week of data before data is lost. Once the connection is restored, the Edge Gateway will send data over based on the max batch size and data frequency until all previous data is sent.

MQTT

The MQTT plugin allows this copy of Edge to push data directly into a Broker using the fast and lightweight MQTT data transfer protocol. In effect, this makes your computer (and whatever devices are attached to it) into a remote MQTT Publisher. If you are already rolling out new smart devices that connect to your Broker, this allows you to connect all of your legacy devices into the same system.

The MQTT plugin is unique in that you must also download and install the **Cirrus Link MQTT modules** from our website <http://www.inductiveautomation.com> to use it. These modules are installed in the Edge Gateway in the same way that modules are installed into a full Ignition Gateway, so see the **Installing or Upgrading a Module** page for more details. The MQTT modules function in the same way that they do in an Ignition installation, so you can check out the documentation at the **Cirrus Link website** to see how to set them up and use them.

SYSTEM

Overview

Backup/Restore

Licensing

Modules

Project

Redundancy

Gateway Settings

NETWORKING

Gateway Network

Email Settings

SECURITY

Audit Log Viewer

Users, Roles

Service Security

Security Zones

ALARMING

Journal

Notification

On-Call Rosters

Schedules

OPC-UA SERVER

Certificates

Devices

Settings

MQTT DISTRIBUTOR

Settings

MQTT ENGINE

Settings

MQTT TRANSMISSION

Settings

MQTT Distributor Settings

General

Users

General Settings

Configuration

Port	1883
Websocket Port	8090
Enable TLS	☐ (default: false)
Keystore Password	password
Secure MQTT Port	8883
Secure Websocket Port	9443
Java Keystore File	Choose File No file chosen Java Keystore File to upload for SSL enabled MQTT

Save Changes

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Edge Common Tasks

New in 7.9.2

Overview

This page details some common configurations that may occur on an Edge Gateway.

Remote Service Security

In many cases, an Edge Gateway may need to interact with another Edge Gateway, or even an Ignition Gateway. In these cases, Service Security on the Remote Gateway must be configured in a manner that allows access to the interested subsystem. For more information, see the [Security Zones](#) page.

Connect to an MQTT Server

Edge Gateways can easily pull values from an MQTT server. This requires that Cirrus-Link's **MQTT Engine module** is installed on the Gateway. Additional details can be found in [Cirrus Link's MQTT documentation](#).

Synchronize Edge Tag History Data with a Remote Ignition History Provider

Edge Gateways can forward their local history to a Historical Provider on an Ignition Gateway. This provides an opportunity to take the local weekly data buffer, and store it into a SQL database for long-term storage and retrieval. This configuration has several requirements:

- Historical Tags on the Edge Gateway should already be configured.
- A remote Ignition Gateway connected to a SQL database.
- The [Tag Historian Module](#) must be installed and running on the remote Ignition Gateway.

1. Configuring Remote History Provider Access

Service Security on the remote Ignition Gateway must allow **Query and Storage** access for the Security Zone the Edge Gateway connection is placed in.

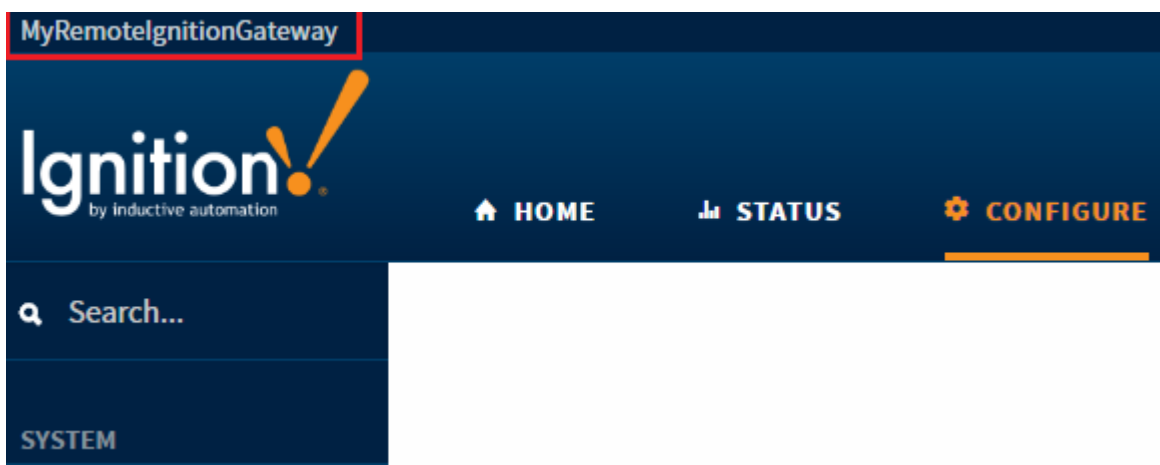
1. On the remote Ignition Gateway, navigate to **Configure > Security > Service Security**.
2. This example will modify the policy on the "Default" Security Zone. Click the Edit button next to the "Default" Security Zone.
3. Under **History Provider Access**, locate the Access Level property that corresponds to the remote Ignition Gateway's History Provider the Edge data should be synchronized with, and set that property to **Query and Storage**. Note that the Default Profile Access could instead be set to **Query and Storage** if multiple Edge Gateways will synchronize with multiple History Providers. In this case, it may be preferable to create a separate Security Zone instead of modifying the "Default" zone.
4. Click the Save button at the bottom of the page to save changes.

2. Gateway Network Connections

Both Gateways must be connected on the Gateway Network. More information may be found on the [Basic Setup section of the Gateway Network](#) page. Once the connection has been established, the History may be synchronized.

When configuring the connection, note the **system name** of the remote Ignition Gateway, as you will have to type it in a later step.

If you're uncertain, you can always find the **Gateway Name** in the upper left corner of the Gateway Web Interface.



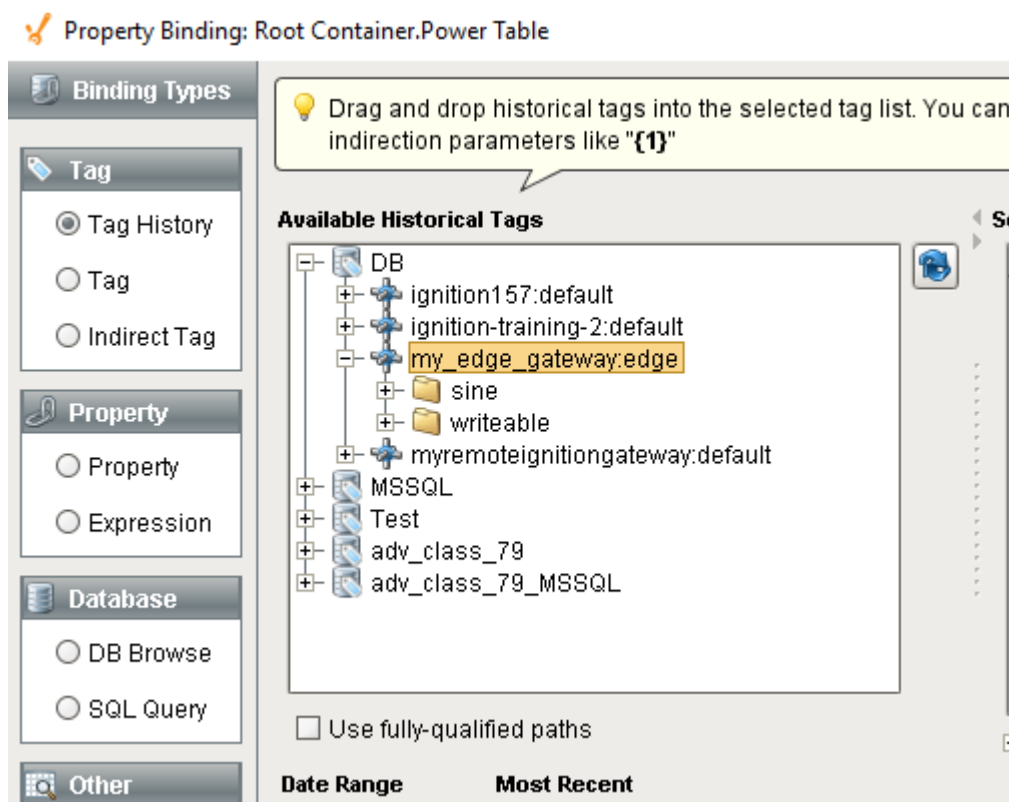
3. Configure History Synchronization

With storage access and a Gateway Network connection, History Synchronization may be enabled on the Edge Gateway.

1. On the Edge Gateway, navigate to **Configure > Enterprise Services > Tag History Sync**.
2. Enable the **Remote Sync Enabled** property to enable synchronization.
3. Specify values for the **Remote Gateway Name** and **Remote Provider Name** properties. The **Remote Gateway Name** is the **system name** of the Ignition Gateway. Check "Gateway Network Connections" above for more information. The **Remote History Provider Name** is simply the name of the Database Connection you wish to store the Edge data in. On the remote Ignition Gateway, navigate to **Status > Connections > Databases** for a full list. Once all properties have been configured, click the **Save Changes** button.

4. View the Synchronized History

With the history synchronized, Tag History Queries on the remote Ignition Gateway will now be able to retrieve history collected on the Edge Gateway. For example, a Power Table with a Tag History Binding on the Data property could display the synchronized data:



Read/Write to Edge OPC Tags From an Ignition Gateway

Ignition Gateways may read and write to Tags on an Edge Gateway. This allows for architectures where Edge Gateways that are simply collecting data that is centralized and maintained from a single Ignition Gateway. An Edge Gateway has a single Tag Provider for OPC tags, named "edge". The following will describe how to provide read and write access to Edge Tags from an Ignition Gateway

1. Service Security on the Edge Gateway

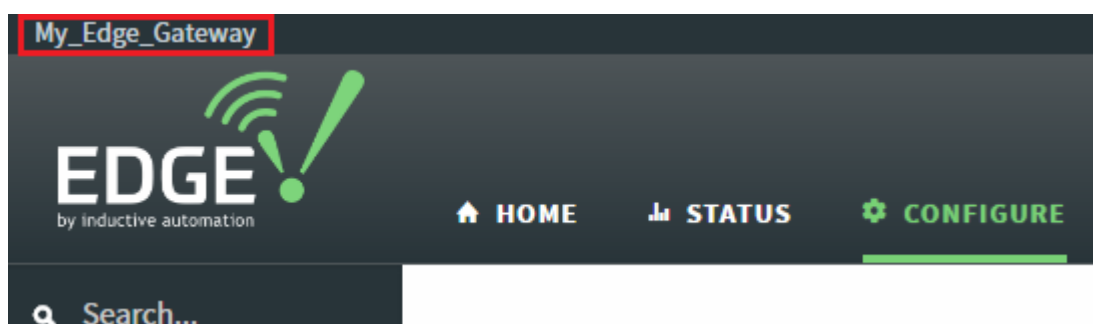
If the Ignition Gateway should only **read** Tags on the Edge Gateway, then this step may be skipped. If **write** access is desired, then the Service Security on the Edge Gateway must be configured to allow writes.

1. On the **Edge Gateway Web Interface**, navigate to **Configure > Security > Service Security**.
2. This example will modify the policy on the "Default" Security Zone. Click the **Edit** button next to the "Default" Security Zone.
3. Under **Tag Access**, setting the **Access Level: 'edge'** to **ReadWrite** will allow remote systems to write to the Tags in the provider, while **ReadWriteEdit** will additionally allow edit access. Note that these same values could be set on the **Default Provider Access Level** property instead, which would set the same access level to all Tag Providers on the Edge Gateway, including Tags exposed by the MQTT Engine module.
4. Once the desired Access Level has been set, click the **Save** button.

2. Gateway Network Connection

Both Gateways must be connected on the Gateway Network. More information may be found on the [Basic Setup section of the Gateway Network](#) page. Once the connection has been establish, the tags on the Edge Gateway will be accessible by other Ignition Gateways.

Once connected, note the system name of the Edge Gateway, as you will need to select it from a list in the next step. If you're uncertain, the system name can always be found in the upper left corner of **Gateway Web Interface**:



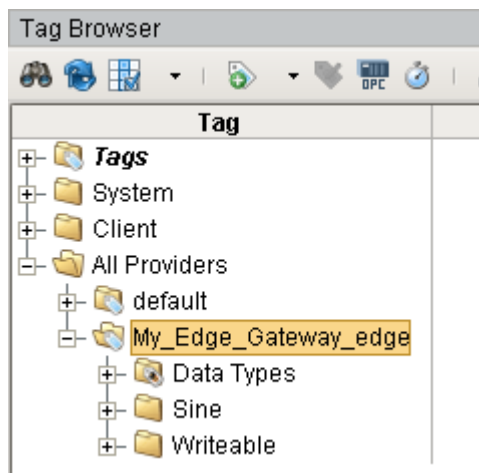
3. Remote Tag Provider from the Ignition Gateway

Now that both Gateways are connected on the Gateway Network, a Remote Tag Provider may be created on an Ignition Gateway.

1. On the **Ignition Gateway Web Interface**, navigate to **Configure > Tags > Realtime**.
2. Below the list of Providers, click the **Create new Realtime Tag Provider** link.
3. Select **Remote Tag Provider (Gateway Network)** and click the **Next** button.
4. From the list of available Gateways, select your Edge Gateway and click **Next**.
5. Select the Tag Provide on the Edge Gateway you wish to access from the Ignition Gateway and click the **Next** button. On Edge Gateways, the provider named "edge" will contain OPC Tags.
6. Review the settings for the Remote Tag Provider. Note that the name of the provider defaults to "GatewayName_ProviderName", but can be changed. When ready, click the **Create New Realtime Tag Provider** button at the bottom of the page. The new remote Tag Provider will appear in the list of Tag Providers.

4. View the Remote Tag Provider

Now that the new tag provider exists, Launch the Designer from the Ignition (non-edge) Gateway, and open any project. In the **Tag Browser**, expanding the **All Providers** folder will show the OPC Tags from the Edge Gateway.



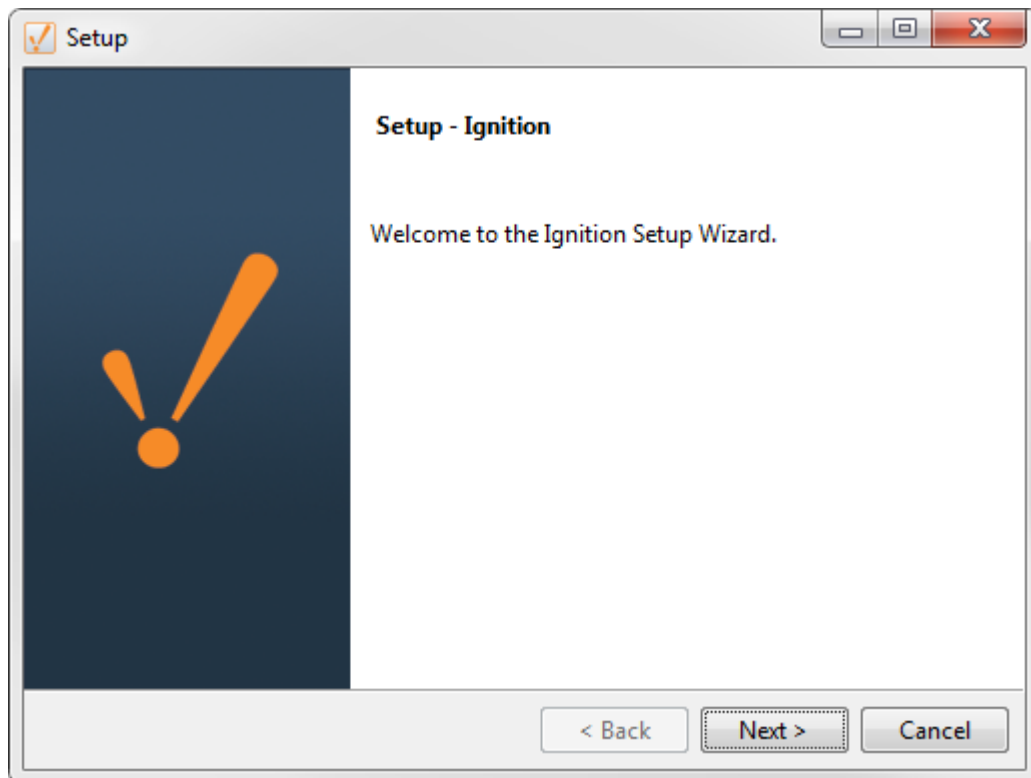
This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Installing and Upgrading

Getting Ignition Running

Getting Ignition up and running on your computer is a very simple process, but it all starts with installation. Fortunately, Ignition only needs to be installed on **one** computer, then any computer on the network can access your projects.



Once you have Ignition installed, you'll be greeted by the Gateway Webpage. That's how you know everything went smoothly. Since you don't need to install clients, check out the [Launching Clients](#) section to see how easy it is.

Simple Installation Steps

You can download and install Ignition in just a few minutes! It's very simple:

1. Download Ignition from your web browser
2. Run the installer on any any Windows, Mac, or Linux operating system that can run Java 8

3. The Gateway Webpage opens automatically after the installer finishes so you can get started immediately!

What Operating Systems Can I Use?

Ignition can be installed on any modern computer that will run Java. This means all of the popular systems will work perfectly, with both 32 and 64 bit systems supported! Inductive Automation has separate installers for each though, make sure you get the right one for your system.

Ignition will install on:

- Windows 32 or 64 bit
- Mac OS X
- Linux 32 or 64 bit
- ARM 32 bit

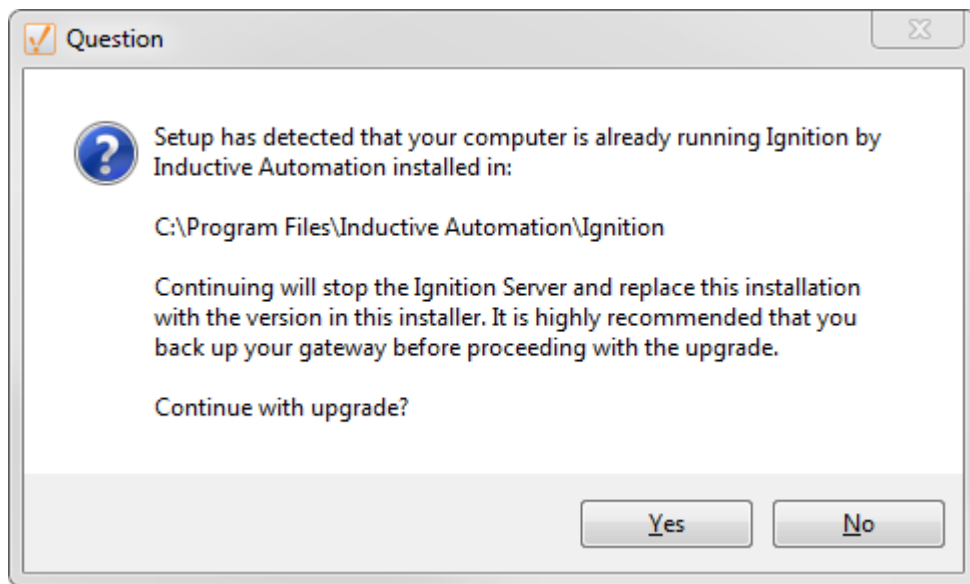
CAUTION

Ensure that you have the correct bit version of Java for the version of Ignition that is being installed.

For step-by-step instructions on installing, see the [Installing Ignition](#) or [ARM - Install](#) pages. Ignition has the ability to use a normal graphical installer where you get to see and control all the steps, as well as a command line installer that can be run automatically.

What about Upgrading?

Upgrading is a snap, too! Just download the installer for your system from our website and run it. That's it! It will figure out if you already have Ignition installed and then change from an installer to an Upgrader automatically.



Simple Upgrading Steps

Regardless of which Operating System you are using, upgrading has the same simple set of steps:

1. Download Ignition within your web browser.
2. Run the installer on the computer that has an older version of Ignition (don't forget to backup your work!)
3. The installer automatically upgrades all of your projects and opens the Gateway Webpage!

For a complete guide to protect your projects when upgrading, see the [Best Practices when Upgrading](#) page.

Installing Individual Modules

Ignition is Modular, so how does that work with the installer? If you use the 'Typical' option when installing Ignition, it will only add our default modules. You can always run the installer again and add other modules with the 'custom' options, or download individual modules and [add them through the Gateway Webpage](#).

Uninstalling Ignition

We hate to see you go, but we understand Ignition is not for every computer. If you need to uninstall, just run the uninstall.exe file from your install directory (typically C:\Program Files\Inductive Automation\Ignition) or use the operating system to remove it. Ignition is not made up of multiple programs so there's nothing special to do.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Installing Ignition

Choose your operating system for detailed instructions on installing and uninstalling Ignition.

Windows

Mac OS X

Linux



INDUCTIVE
UNIVERSITY

Installing Ignition on
Windows

[Watch the video](#)

BEFORE YOU INSTALL IGNITION, MAKE SURE YOU MEET THE MINIMUM SYSTEM REQUIREMENTS

- Java 8+ (server), Java 6+ (client)
- Dual-Core Processor (32 or 64 bit)
- 4GB RAM
- 10GB free HD space

Also, ensure you have Administrator privileges to run the Ignition installer.

1. Go to the Inductive Automation website at www.inductiveautomation.com
2. Select **Download Ignition** in the upper right corner of the page.
3. On the Ignition Downloads page, scroll down to **System Installers** and select the **ignition-X.X.X-windows-x64-installer.exe** installer.

If your computer runs on a 32-bit Windows, click the **ignition-X.X.X-windows-32-installer.exe** installer.

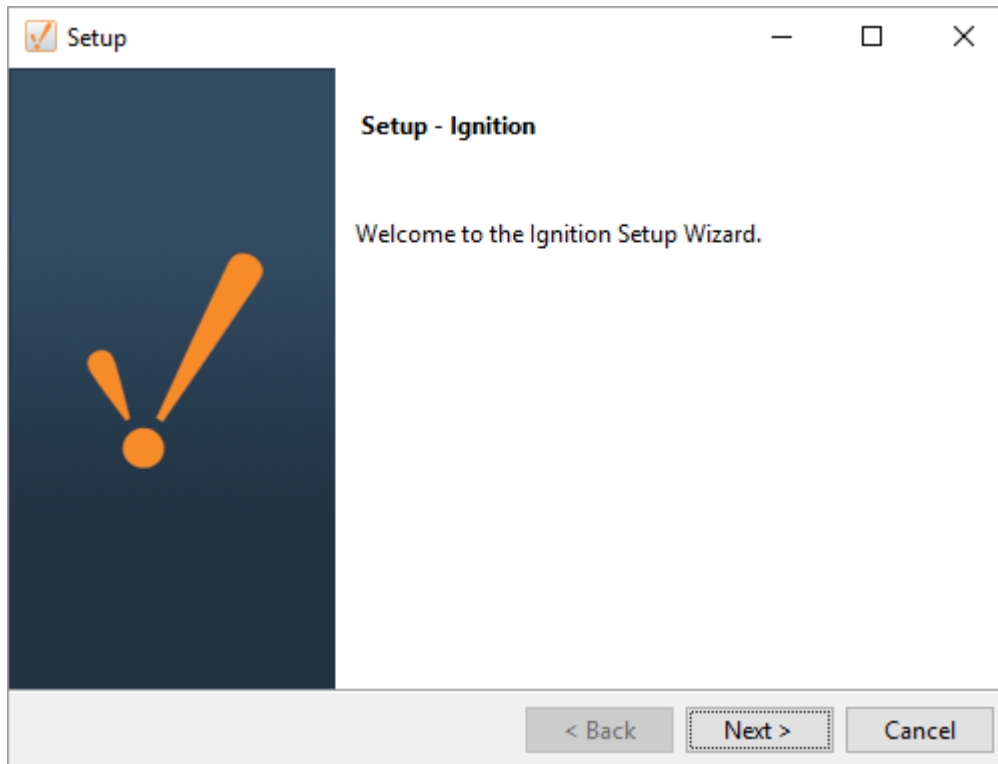
NOTE

An error window is displayed if you try to install 64-bit Ignition on a 32-bit Windows system. You are allowed to install 32-bit Ignition on a 64-bit system (however, this is not

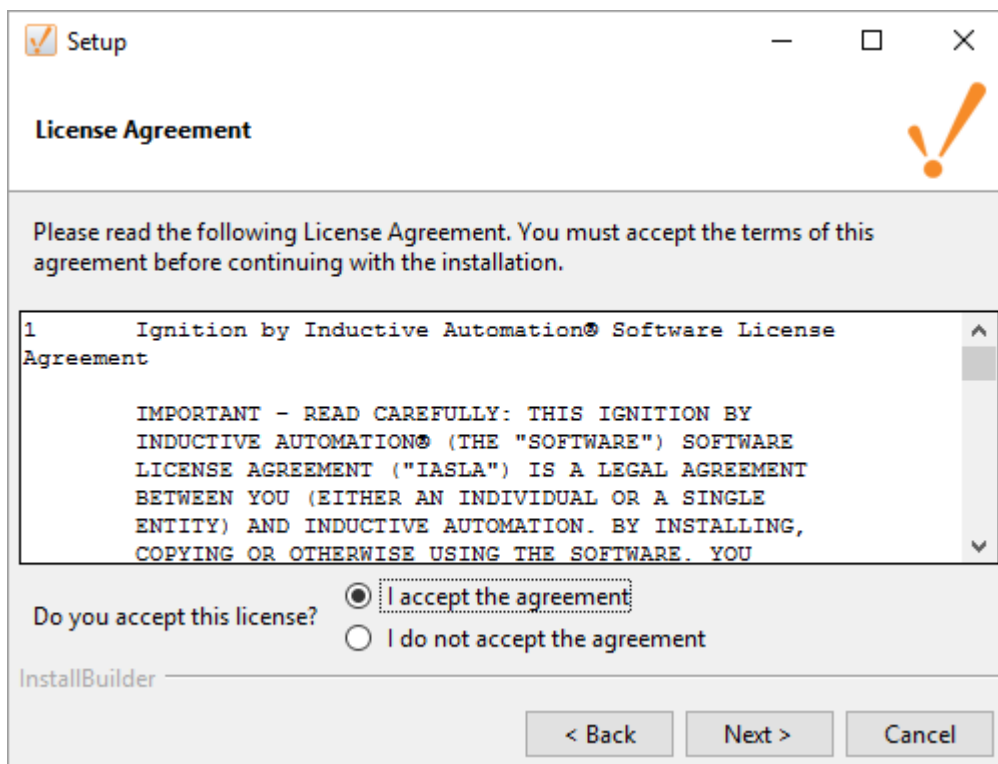
recommended).

4. After the installer downloads, double-click on the **.exe** executable file to start installation.

The Setup window welcomes you to the Ignition Setup Wizard.

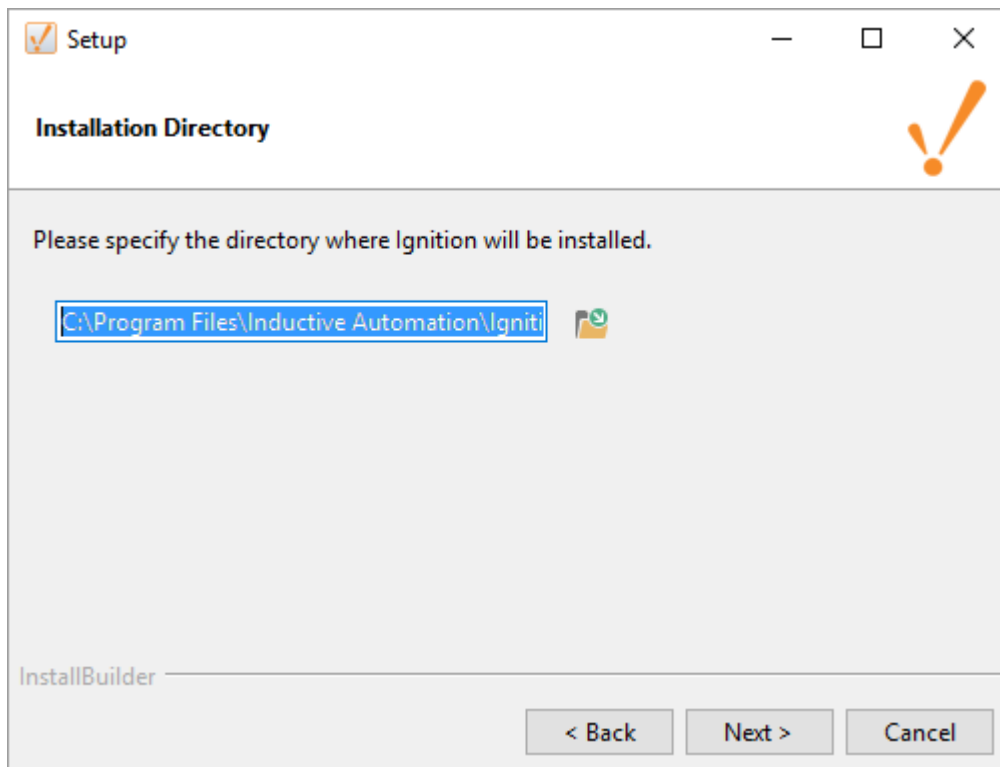


5. Click **Next**.
6. Click the **I accept the agreement** button and click **Next**.



The Installation Directory window is displayed. By default, Ignition is installed under your Program Files directory. This is usually a good choice.

7. Make a note of the default installation directory, **Program Files/Inductive Automation/Ignition**, accept the default Installation Directory, and click **Next**.



The **Installation Mode** window is displayed.

8. Select the platform you wish to install. The rest of this guide will use **Ignition** as the platform.

New in 7.9.2

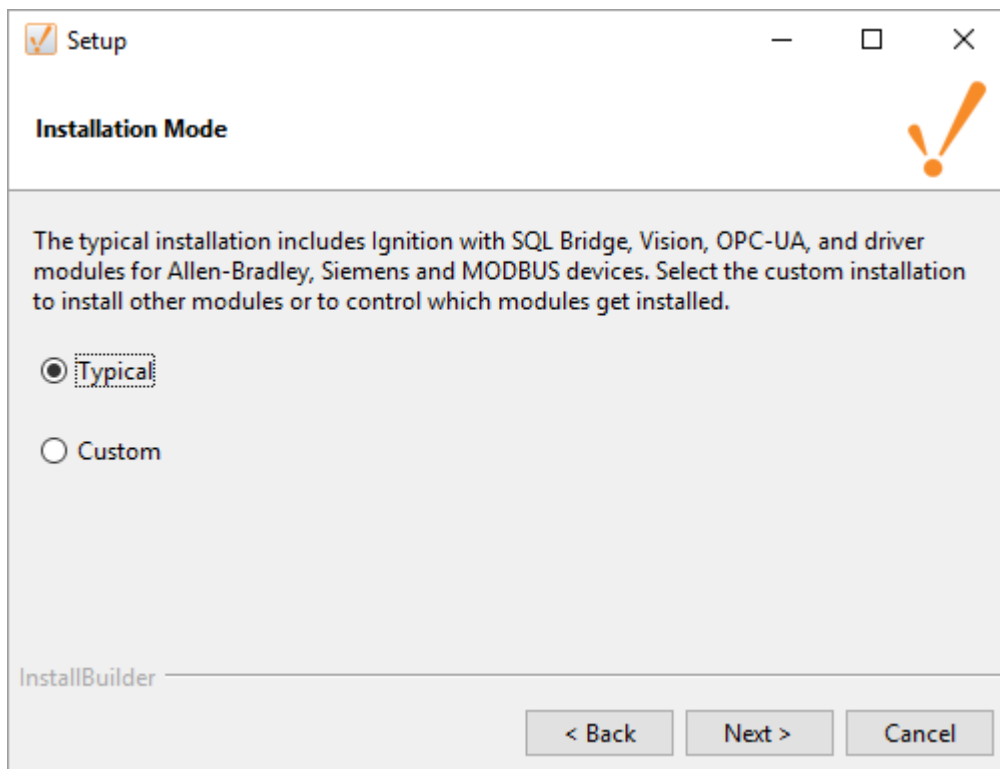
For Ignition Edge installations, select the **Ignition Edge** radio button then skip to step 11. More information on Edge can be found on the [Ignition Edge](#) page.



9. Select either the **Typical** or **Custom** mode.

The **Typical** installation includes Ignition with all of the HMI/SCADA modules. The only modules that are not installed by default are the MES modules.

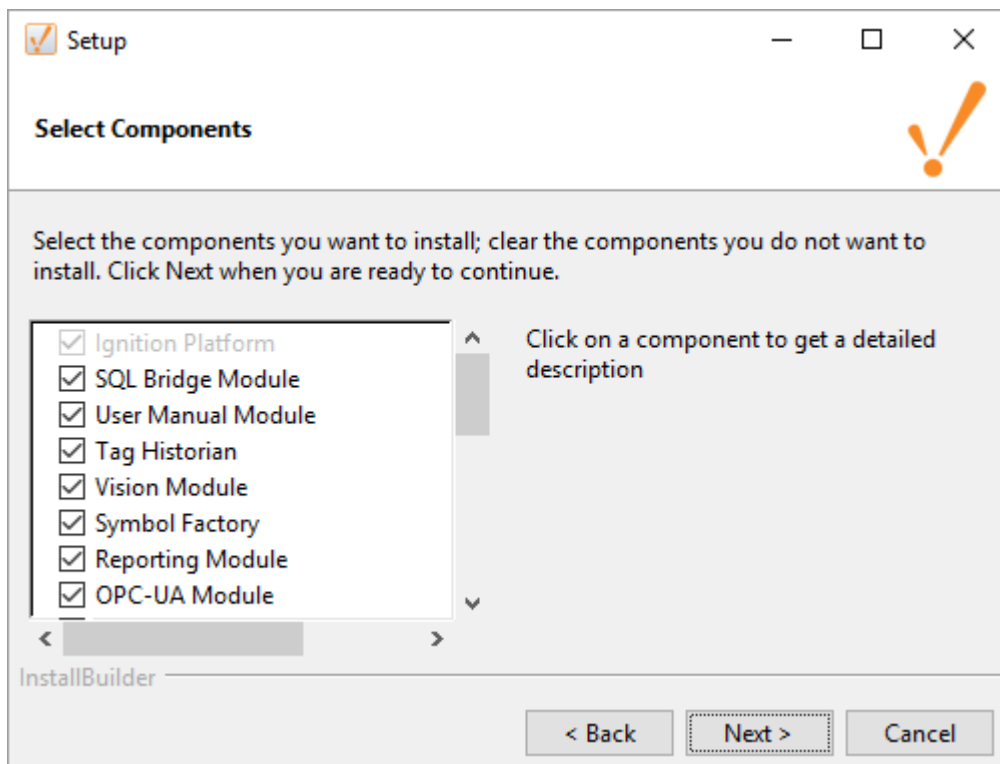
The **Custom** installation lets you control and individually select the modules that you want installed.



10. Click **Next**.

The Setup window is displayed.

If you selected Custom mode, in the Setup window you can select the check box next to the modules you want to install as part of the Ignition installation. To see a brief description of the module, click on the module name.



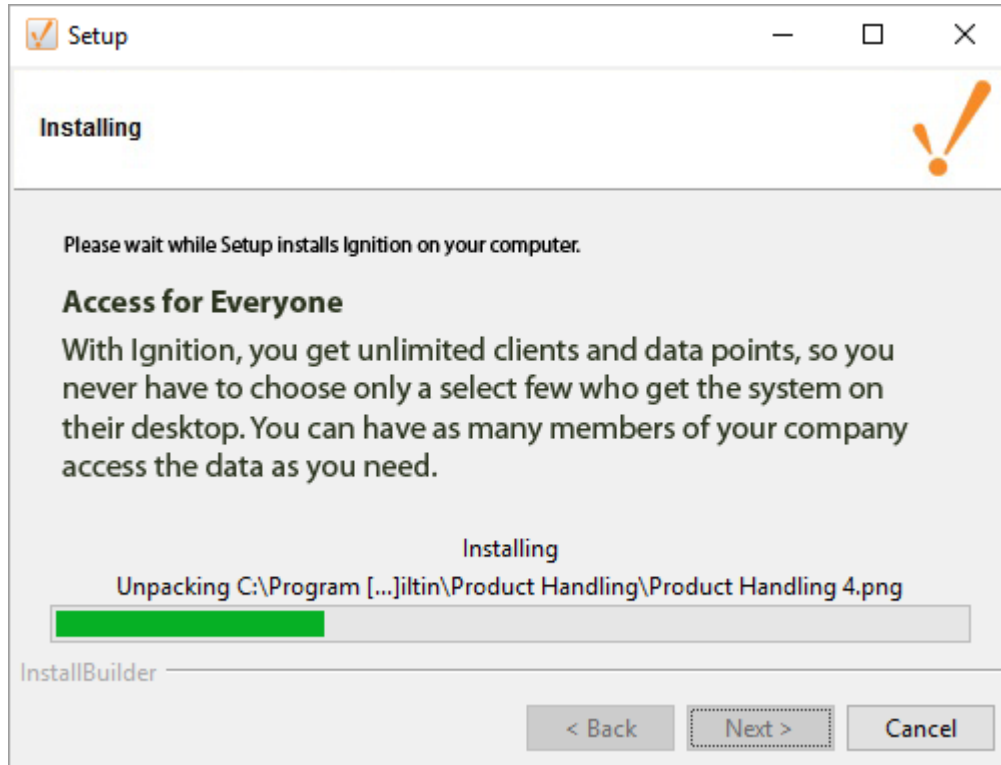
11. Click **Next**.

The Setup Wizard shows Ignition is now ready to install.

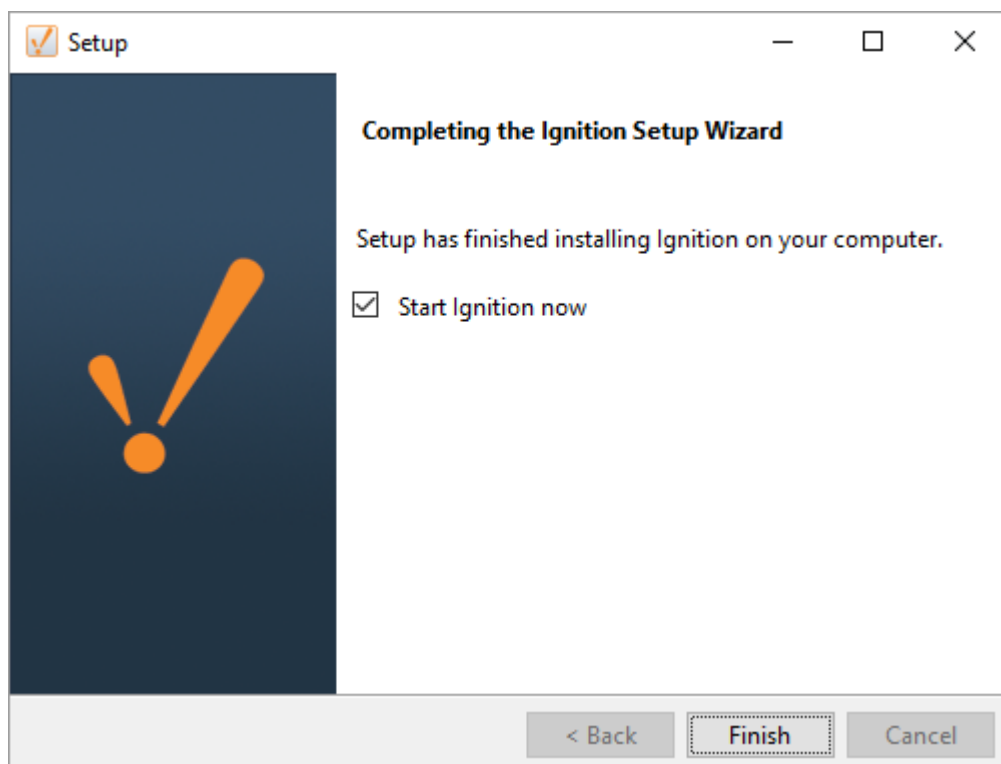
If you decide to change your selections, click **Back** to go back and make your changes.

12. Click **Next**.

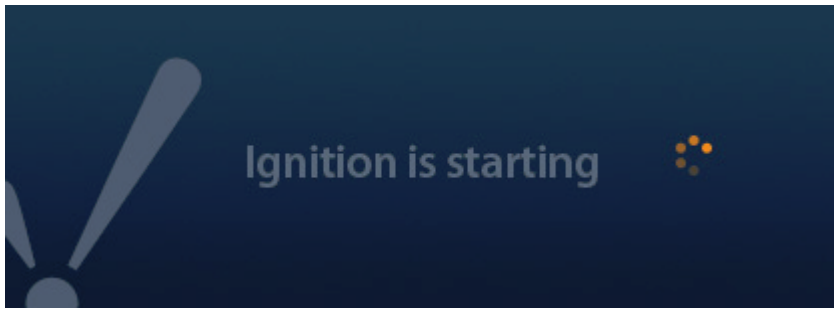
Ignition will install the selected modules.



13. After the Setup Wizard shows Ignition installation is finished, make sure that the check box for **Start Ignition now** is selected, then click **Finish**.

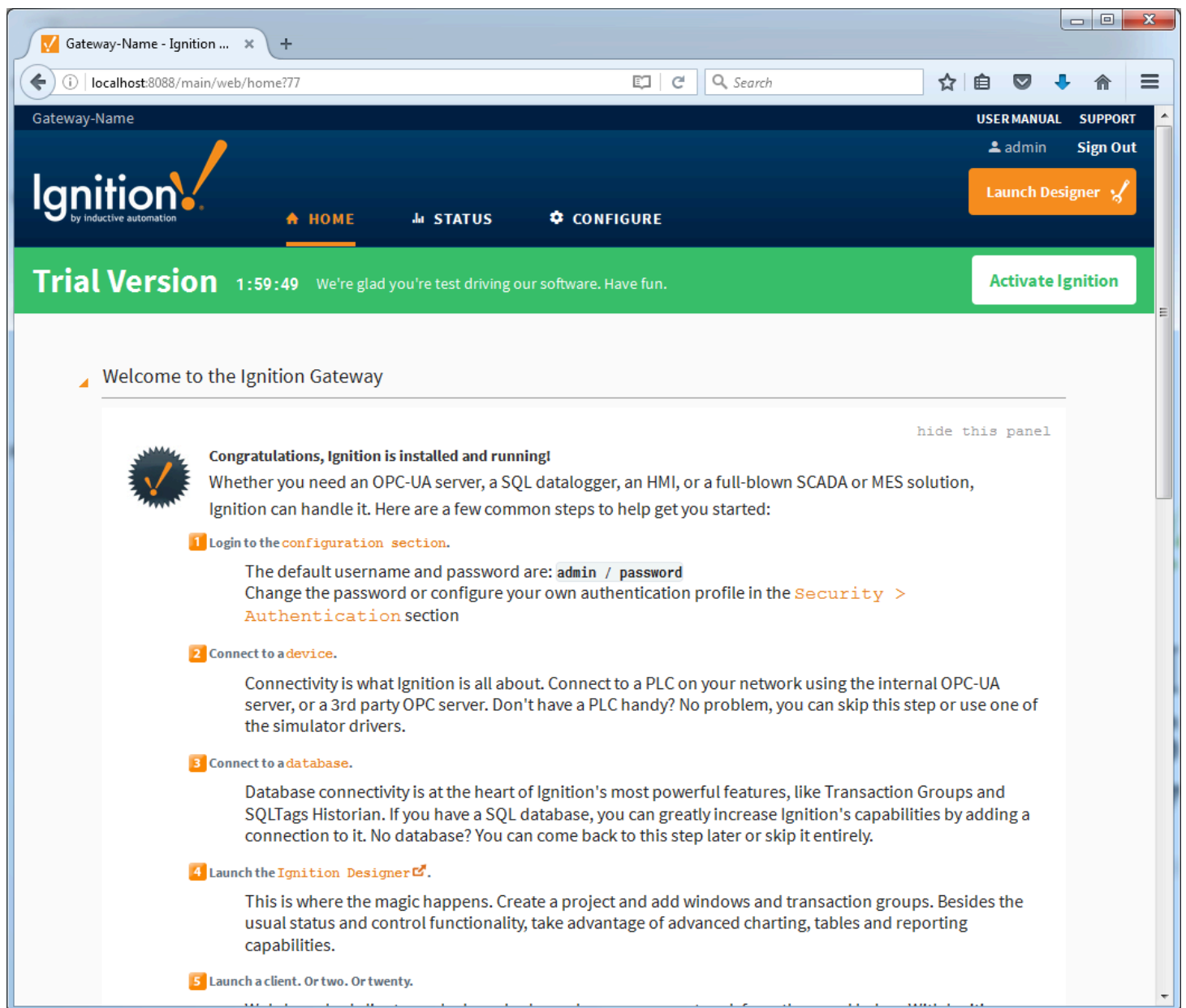


A splash screen informs you that Ignition is starting. This may take a few minutes to finish.



That's it! Ignition Installation is complete and the Gateway is then launched.

Your web browser opens the Gateway home page at **http://localhost:8088**



Ignition installs itself as a Windows Service (**Start > Control Panel > Administrative Tools > Services > Ignition Gateway**), so it starts automatically when your computer starts up.

Automated Installation on Windows

You can automate Ignition installation across different machines using scripts. Ignition is then installed silently from a command shell without showing any user prompts.

Keep in mind that the installer cannot automatically start the Gateway after a silent installation. You can use the following command to start the Gateway: `net start ignition`

Command line example

```
Ignition-7.x.x-windows-x64-installer.exe --mode unattended --prefix "C:\some folder" --unattendedmodeui none

net start ignition
```

Flags

- `-- mode unattended` (ensures that no prompts appear during installation)
- `-- prefix "C:\some folder"` (optional flag; if a value is set, then Ignition is installed in the specified folder, otherwise Ignition is installed in a default location under C:\Program Files)
- `-- unattendedmodeui none | minimal` (the none flag does not display any sort of graphic during installation; the minimal flag displays a small progress bar and nothing else)

Uninstalling Ignition on Windows

BEFORE YOU UNINSTALL IGNITION...

Be sure to back up your Gateway and unactivate your Gateway license.

During a new Ignition installation, an uninstaller executable is created.

When Ignition is uninstalled, the settings database and folder in `/data` is backed up to `/data_<current date>`, and modules and user-supplied JDBC jars in `/user-lib` are backed up to `/user-lib_<current date>`. Also, the Ignition service is removed from Windows installations automatically.

To run the uninstaller for Windows:

1. Open the Start menu and navigate to **All Programs > Inductive Automation > Ignition**.
2. Select **Uninstall Ignition**. The uninstaller wizard guides you through the uninstallation process.

You can also uninstall Ignition from the Windows **Control Panel > Programs and Features** section.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Upgrading Ignition



TIP

Upgrading from 7.6 or earlier?

Please read the [v7.7 Upgrade Guide](#) below before you begin.

Choose your operating system for detailed instructions on upgrading Ignition.

Windows

Mac OS X

Linux

BEFORE YOU UPGRADE IGNITION...

You should backup your Gateway, see [Gateway Backups](#).



INDUCTIVE
UNIVERSITY

**Upgrading Ignition on
Windows**

[Watch the video](#)

The Ignition installer also doubles as an upgrader, thus upgrading Ignition is very similar to downloading a new version and installing from scratch. You can see your current Ignition version by looking in the **Status** page of the Gateway.

To Upgrade Ignition on Windows

1. Simply download the Windows executable installer from our website, and double-click on the executable installer.

A window is displayed asking you **Continue with upgrade?**

NOTE

Make sure you download the 32-bit installer for a 32-bit Windows installation, and the 64-bit installer for a 64-bit Windows installation. If you attempt to upgrade a 32-bit installation with a 64-bit installer, or vice versa, you will get an error message.

2. Click **Yes** to start the upgrade process.
3. Accept the license agreement. The **Upgrade Mode** window is displayed.
4. You choose either:
No, just upgrade currently installed modules

The Ignition base installation and the existing modules are upgraded.

Or you can choose:

Yes, let me choose which modules to add or remove

The Select Components window is displayed and you can then add a module that was not part of the original installation or uninstall/remove a currently installed module. You can view a brief description of each module by clicking on the module name. Selecting the check box next to a module installs the module as part of the upgrade. Clearing the check box next to a module uninstalls the module during the upgrade.

5. Click **Next**. The **Ready to Install** window is displayed.
Ignition is ready to be upgraded. At this point, you may click **Back** to change your selections.
6. Click **Next** to finish the upgrade. Once the upgrade process starts, it may take a few minutes to finish. At the end of the upgrade process, the Setup window is displayed letting you know that **Setup has finished installing Ignition on your computer**.
7. Select the check box for **Start Ignition now** and click **Finish**.

A splash screen displays informing you that the Ignition service is starting. Once the Ignition Gateway starts up, your web browser opens and brings you to the Gateway Homepage.

You can go back and check the **Status** page of the Gateway to see that a new version of Ignition is now installed.

Automated Upgrading

You can silently upgrade Ignition from a command shell without showing any user prompts.

Remember to make a Gateway backup before performing any type of upgrade. Also, the installer cannot automatically start the Gateway after a silent upgrade.

Use the `net start ignition` command as shown below.

Command line example

```
gwcmd -b C:\backups\mybackup.gwbk
```

```
Ignition-7.x.x-windows-x64-installer.exe --mode unattended --unattendedmodeui none  
--autoupgrade true
```

```
net start ignition
```

Flags

- `-- mode unattended` (ensures that no prompts appear during installation)
- `-- unattendedmodeui none | minimal` (the none flag does not display any sort of graphic during installation; the minimal flag displays a small progress bar and nothing else)
- `-- autoupgrade true` (disables any Are you sure? prompts when running the upgrader so that a user can run an automated upgrade without any user interaction)

v7.7 Upgrade Guide (Upgrading from 7.6 or Earlier)

NOTE

The following applies specifically when upgrading a legacy system from **Ignition 7.6 or earlier to 7.7**. It was originally published as a standalone tip sheet and is preserved here for reference.

Thinking of upgrading your existing Ignition system to version 7.7? Can't wait to get your hands on all those new features? Great! But first, read this short upgrade guide to be prepared.

1. Make a Backup First

Before any major upgrade, it is important to make a backup of your existing system, and write down what version you were using. That way, if you encounter any unexpected turbulence, you can revert back quickly.

2. Report Issues

If you find anything out of place, let us know! Visit the Ignition 7.7 forum to report any anomalies you find.

3. Call Your Sales Rep

Ignition 7.7 is a major upgrade to the Ignition platform and all of its modules. When you upgrade, all of your modules will revert back to trial mode until you update your license.

- If you have **Ignition Total Care Upgrade Protection**, you don't need to pay anything additional to upgrade your license key, but you will need to update the license that is installed on your server(s). If you have internet access on your Gateway, this is as easy as clicking the **Update License** link on the license page of the Gateway's configuration web interface. If you don't have internet access, you'll need to do an offline activation.
- If you **don't have Total Care Upgrade Protection**, you'll need to contact your sales rep and arrange to purchase the upgrade.

4. Upgrade Java

Ignition 7.7 requires Java SE 8 for the Gateway. Clients and Designers still support Java 6, 7, and 8.

5. Custom Tomcat Settings

If you added items like MIME types to the `tomcat/conf/web.xml` folder by hand, you'll need to apply those changes to `webserver/webdefault.xml` after upgrading.

6. Scripting Changes

Ignition 7.7 has a number of new scripting features. If you're a heavy script user, you should be aware of what these changes mean.

- Going forward, `app.*` is being replaced by `project.*`. However, if you upgrade an existing installation with `app.*` scripts, they will remain indefinitely. Whether you move them over to the `project.*` space is up to you.
- New scripting locations use new scoping rules. Ignition's scripting semantics have changed somewhat for new scripts. We now correctly use Python's module scoping rules. This has two major effects:
 - Variables are now directly accessible from within functions. For example:

Old Scoping

```
x = 5
def myFunction(x = x):
    print x
```

New Scoping

```
x = 5
def myFunction():
    print x
```

- The `global` keyword now works differently. Instead of making the variable truly "global", it declares it global within the scope of the current module. See the Python Programming FAQ for more details. In order to make new scripts compatible with older scripts that used global variables, there is a new function, `system.util.getGlobals()`, which provides access to the old global space.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

ARM - Install

What is ARM?

Without getting too technical, an **ARM** (or **A**dvanced **R**ISC **M**achine) usually refers to a very small and usually cheap device that is often created for a very specific task. These machines are so called because they use an ARM processor (a small processor with less speed/capabilities than a traditional computer processor). You often see these processors in small devices like cellphones, watches, or others that have a limited amount of space and/or battery power.

In the context of Ignition, you might take a small device (like a Raspberry Pi or similar) and install Ignition to create an edge of network device for simple data logging. You could also install a slimmed down operating system like Linux and use the device as a client for an existing Ignition installation. This page details installing Ignition on the device and is focused on our edge of network example rather than using the device as a client.

ARM Processor Installation

Installing Ignition on an ARM processor cannot be done with the standard installation executable, but is otherwise very simple. To install on ARM, follow these steps:

NOTE

The zip will also contain a README file that will contain some useful information on the installation process.

1. Go to the Inductive Automation website at www.inductiveautomation.com
2. Select **Download > Current Release**.
3. On the Ignition Downloads page, scroll down to the **ARM** section of the zip files.

New in 7.9.2 There is a separate ARM installer file for Ignition Edge named **Ignition-Edge-linux-armhf-x.x.x.zip**.

4. Download the appropriate zip file (**Ignition-Linux-armhf-x.x.x.zip**) using the download link.

i NOTE

Depending on your computer, you may need to run all the commands as root. Prefix everything with `sudo` or first run `sudo su`.

5. Download and install Java 8 if it is not already installed. Go to <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
6. Unzip the Ignition download file and install Ignition into `/usr/local/ignition` by running the following command:

```
unzip <<ignition-Linux-armhf-x.x.x.zip>> -d /usr/local/ignition
```

where `<<ignition-Linux-armhf-x.x.x.zip>>` is the path to the Ignition ARM distribution you downloaded.

7. Change directories to the Ignition install directory.

```
cd /usr/local/ignition
```

8. Make the files executable by executing the following three `chmod` commands:

```
chmod +x ignition.sh
chmod +x ignition-gateway
chmod +x gcu.sh
```

9. To start Ignition now, type `./ignition.sh start`

You can now open a web browser and go to the Gateway home page at <http://localhost:8088>.

Stop and Start Ignition

After installation, you can start and stop Ignition with the following commands:

```
./ignition.sh start  
./ignition.sh stop
```

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Installing or Upgrading a Module



INDUCTIVE
UNIVERSITY

**Installing or Upgrading a
Module**

[Watch the video](#)

All module configuration is done from Gateway in the **Configuration > Modules** section. From here you can add, remove, and restart modules. Modules integrate their settings into the Gateway configuration tree, and therefore do not offer settings in this section.

Controller

USER MANUAL
SUPPORT

admin
Sign Out

Launch Designer

HOME
STATUS
CONFIGURE

Search...

License Incomplete ? 0:00:00
Reset Trial
View Modules

SYSTEM

Overview
Backup/Restore
Licensing
Modules
Projects
Redundancy
Gateway Settings

NETWORKING

Gateway Network
Email Settings

SECURITY

Auditing
Users, Roles
Service Security
Security Zones

DATABASES

Connections
Drivers
Store and Forward

ALARMING

Legacy Notification
Legacy Storage
General
Journal
Notification
On-Call Rosters
Schedules

MAP PANEL

Tile Provider Proxy

Module Configuration

Inductive Automation
View Certificate

Name	Version	Description	License	State		
Alarm Notification	4.9.0 (b2016101208)	Provides alarm notifications via email	Activated	Running	More ▾	restart
Allen-Bradley Driver	4.9.0 (b2016101215)	Allen-Bradley driver suite for the OPC-UA module.	Activated	Running	More ▾	restart
DNP3 Driver	2.9.0 (b2016101215)	A driver supporting DNP3 (Distributed Network Protocol) device.	Activated	Running	More ▾	restart
Enterprise Administration	2.9.0 (b2016101215)	A remote Gateway administration system, allowing you to manage Gateways and automate tasks from a single controller.	Activated	Running	More ▾	restart
SQL Bridge	8.9.0 (b2016101208)	An OPC-to-SQL data logger and transaction manager.	Activated	Running	More ▾	restart
Symbol Factory	5.9.0 (b2016101215)	Vector graphics clipart library for the Vision module.	Activated	Loaded	More ▾	restart
Tag Historian	2.9.0 (b2016101208)	Turns any database into a powerful historian that can store and drive data in Ignition.	Activated	Running	More ▾	restart
UDP and TCP Drivers	4.9.0 (b2016101215)	Drivers for receiving and parsing UDP or TCP packets.	Activated	Running	More ▾	restart
User Manual	4.9.0 (b2016101208)	Provides an offline version of http://docs.inductiveautomation.com.	Free	Running	More ▾	restart
Vision	9.9.0 (b2016101215)	A module that provides web-launched HMI/SCADA clients.	Activated	Running	More ▾	restart

→ Install or Upgrade a Module...

To Install or Update a Module

- On the **Configuration > Modules** page of the Gateway, go to the bottom on the list, find the blue arrow, and click the **Install or Upgrade a Module** link.
- Click **Choose File** and select a `.mod1` file that you have previously downloaded.

! DON'T HAVE AN IGNITION MODULE ALREADY DOWNLOADED?

Go to the Inductive Automation [downloads page](#). Search for the module you want and save it to your Downloads folder on your computer.

Third Party Modules

All third party Ignition modules require the Ignition platform to be installed.

To install third party modules:

- Install Ignition: See [Installing and Upgrading Ignition](#)
- Once Ignition is installed, download the module and install it in the Ignition Gateway:
See [Installing or Upgrading a Module](#)

SepaSoft, Inc. MES Modules for Ignition		
OEEDT Installer Module	OEEDT-Installer-module.modl (8MB)	Version: 1.9.0.1610111600
SPC-Installer-module	SPC-Installer-module.modl (10MB)	Version: 1.9.0.1610111606
Instrument Interface-module	Instrument-Interface-module.modl (1MB)	Version: 1.9.0.1610111601
Recipe-Installer	Recipe-Installer-module.modl (8MB)	Version: 1.9.0.1610111609
Productionsimulator-Installer	Production-Simulator-module.modl (58KB)	Version: 1.9.0.1610111559
Trace-Installer	Trace-Installer-module.modl (7MB)	Version: 1.9.0.1610111555
Web Service-module	Web-Service-module.modl (2MB)	Version: 1.9.0.1610111611
Barcode Scanner-module	Barcode-Scanner-module.modl (61KB)	Version: 1.9.0.1610111610

Note: If you are using Internet Explorer, IE changes the file extension to a '.zip' file. Rename '.zip' to '.modl' (I.e., Production Simulator-module.modl).

Go back to the **Install or Upgrade Module** screen and browse your Downloads folder for your module, select it, and click **Open**. Ignition will copy the path to the module to the Browse field and all that's left for you to do is click **Install**.

3. Click **Install**. When the page reloads you can now see the module you installed in the list of modules.

To Uninstall a Module

1. On the **Configuration > Modules** page of the Gateway, go to the module you want to uninstall.
2. Under the **More** button, click the **uninstall** link on the far-right next to the module entry in the table. Another page opens asking you to confirm.
3. Click **Confirm**. The module is no longer installed and is removed from the list.

Modules are hot-swappable, this means these actions can be performed while the system is running. Furthermore, the isolated nature of modules ensures that performing one of these actions only affects that particular module, and any modules which depend on it. For example, uninstalling the SQL Bridge module will not affect any running Vision module clients.

Restarting a Module

Modules can be restarted by clicking the **restart** button next to their entries on the **Configuration > Modules** page. Because of the isolated nature of modules, the other modules are not affected by the **restart** (unless they depend on that particular module).

Module Status

The installed module list on the **Configuration > Modules** page also provides some basic information about the **State** of the module. The **Version**, **License**, and **State** columns are all displayed in the list. Module licensing is performed centrally in **System > Licensing**, so the values here are only for information purposes.

The module **State** can be either **Running** or **Loaded**. Running means the module programs are actively performing tasks, for example, the Alarm Notification modules shows as Running because it is continually checking for condition changes that need to set the alarm on or off. Loaded means the module program or resources are available but nothing is running, for example, the Symbol Factory module makes all the symbol images available for your use.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Third Party Module - Signing Options

Third Party Modules - Signing Options

Several third party modules add even more possibilities and functionality to help you design a project. To use third party modules with Ignition, module authors have several different options to choose from when it comes to obtaining signed certificates. They can use a signed certificate signed by a trusted certificate authority (CA), or self sign their own module. New in Ignition version 7.9 (and retroactively added in versions 7.7.8 and 7.8.3), there are two things Ignition will ask you to verify before using a new module:

End-User License Agreement (EULA)

The first time a third party module is installed, the user will be prompted to accept the end-user license agreement (**EULA**) establishing the user's right to use the software. After the EULA is accepted once, you will not be prompted again for any identical licenses, even if it is a different module. Any changes to the license will cause the module to be quarantined until the new license is accepted.

Accept Certificate

The process for accepting **trusted** or **self-signed** certificates work the same way as the process for the accepting a EULA. These two types of certificates are digital signatures verifying the author's identity and telling Ignition that it is a trusted module. The first time a module is installed, the user will be prompted to accept the certificate. After the certificate is accepted once, you will not be prompted again for any identical certificates, even if it is from a different module.

❗ USING A SINGLE CERTIFICATE FOR MULTIPLE MODULES

If module authors use a single certificate for all their modules, their users will have a more pleasant install experience because once a certificate is accepted by the user, it is valid for any modules that use it.

Module Configuration Page

You can check on all the installed and uninstalled modules from the Gateway webpage on the **Configure > Modules** page. The Module Configuration page is grouped by categories and you can see at a glance what modules are installed, which modules have certificates, and which modules are unsigned.

Inductive Automation modules are displayed first, followed by SepaSoft modules. Modules from other authors are grouped by the trust level of their modules certificate. Authors who use multiple certificates will have multiple groups. On the **Module Configuration** page, you'll notice all certificates will display a link to the general details about the certificate. Modules in the Quarantined section are not installed, and typically need some action (i.e., accept the certificate and/or license, or install the module). Unsigned modules when not running in Developer mode will also be in the Quarantine section, and the Install button will be grayed-out so the module cannot be installed until the Gateway is restarted in Developer mode

Controller

by inductive automation

USER MANUAL

SUPPORT

admin

Sign Out

Launch Designer

HOME

STATUS

CONFIGURE

Search...

SYSTEM

Overview

Backup/Restore

Licensing

Modules

Projects

Redundancy

Gateway Settings

NETWORKING

Gateway Network

Email Settings

SECURITY

Auditing

Users, Roles

Service Security

Security Zones

DATABASES

Connections

Drivers

Store and Forward

ALARMING

Legacy Notification

Legacy Storage

General

Journal

Notification

On-Call Rosters

License Incomplete ? 0:00:00

Reset Trial

View Modules

Module Configuration

Inductive Automation

View Certificate

Name	Version	Description	License	State	
Alarm Notification	4.9.0 (b2016101208)	Provides alarm notifications via email	Activated	Running	More ▼ restart
Allen-Bradley Driver	4.9.0 (b2016101215)	Allen-Bradley driver suite for the OPC-UA module.	Activated	Running	More ▼ restart
DNP3 Driver	2.9.0 (b2016101215)	A driver supporting DNP3 (Distributed Network Protocol) device.	Activated	Running	More ▼ restart

Unsigned

Name	Version	Description	License	State	
Acme Module	1.0.0	Module description	Trial	Running	More ▼ restart

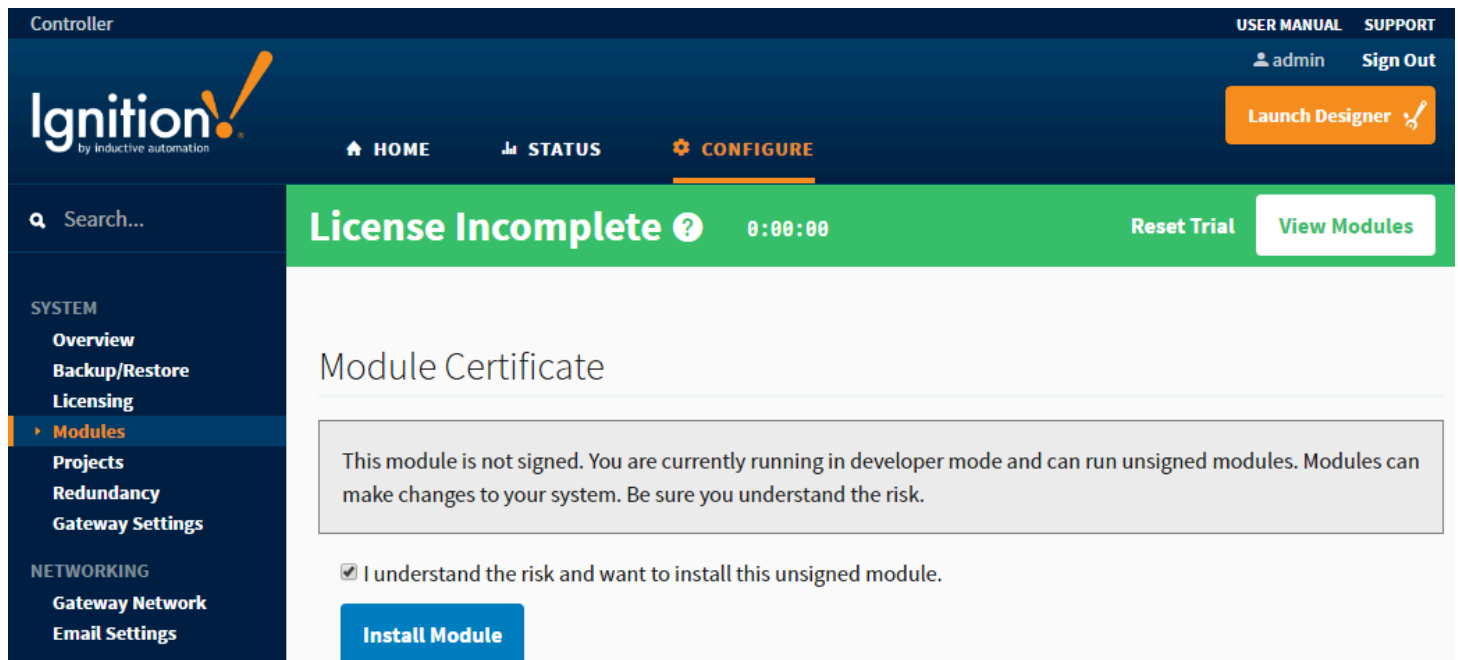
Quarantined Modules

Name	Version	Description	Quarantine Reason	
Ace Module	1.3.8	Module description	Certificate has not been accepted.	delete install

Installing Third Party Modules

Installing Modules via the Gateway

For Inductive Automation and other third party modules that have already been signed, nothing is changing about the installation. For new third party modules with either no certificate, or signed the new way with a certificate that the user has not previously accepted, there is one extra step in the installation process. Go to the **Configure > Modules** page to accept the end-user license agreement. The user will then be directed to a page where they can view the module details and accept the certificate if there is one, or a message will be displayed that there is no certificate. The user will be asked to accept any risks of running unsigned modules as shown in the example below. An unsigned module will *not* install unless the user is in Developer mode. As a result, the user will not be able to continue with the module installation if Developer mode is not enabled.



Installing Modules via a Modules Folder

For Inductive Automation and previous third party modules, nothing is changing about the installation. For third party modules that have a license or a certificate, the Gateway will examine the module at startup and determine if the user already accepted the license and certificate. If not, a warning will be logged and the module will not start. To accept the license and the certificate, go to the **Module Configuration** page of the Gateway webpage and click the **Install** link to the right of the module. You will be prompted to accept the terms of the license and/or the certificate.

Module Upgrades

Third party module upgrades should go seamlessly unless the module licenses or certificates changed. (This does not affect Inductive Automation modules). If third party modules changed or the user is upgrading from an Ignition version without this signing system, modules will be moved to the Quarantine section until the new licenses and certificates are accepted.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Best Practices When Upgrading

Steps to Upgrade

The Ignition installer also doubles as an upgrader. Simply download the executable installer from our website for your operating system. See the upgrade pages for instructions specific to Windows, Linux, and Mac OS X operating systems.

Regardless of which Operating System you are using, you should follow these steps prior to running the Ignition installer:

1. Make a Gateway Backup (most important)
2. Check your versions
3. Update your License (if needed)
4. Update on a Test Server
5. Upgrade the Production Server

For more information about the changes in each version, see the [New in this Version](#) page.

Make a Gateway Backup

Upgrading Ignition is very easy, but you should always create a gateway backup before upgrading. If you need to rollback to the previous version of Ignition, then having a gateway backup from before the upgrade will allow you to easily move back to your prior version. You can rename your backups to include a version number to make them easy to track.

Go to the **Configure** section of the Gateway Webpage, select **System > Backup/Restore** and make a backup before running the installer/upgrader.

Controller

USER MANUAL SUPPORT

admin Sign Out

Launch Designer

HOME STATUS **CONFIGURE**

Search...

License Incomplete ? 0:00:00 Reset Trial View Modules

SYSTEM

Overview

Backup/Restore

Licensing

Modules

Projects

Redundancy

Gateway Settings

NETWORKING

Gateway Network

Email Settings

System Backup / Restore

Backup Restore Legacy Upgrade

Download Backup ↓

Backs up the *user data* inside this Ignition Gateway server. This includes all projects, drivers, images, and configuration, but not the modules.

Check Versions

Is this is a Major version update or a Minor version update? You can check the version of your Ignition in several ways. The easiest way is to log into the **Status** section of the Gateway Webpage, and select **Systems > Overview**. The version is listed on the **Overview** page under **Architecture**.

Controller

USER MANUAL SUPPORT

admin Sign Out

Launch Designer

HOME **STATUS** CONFIGURE

Search...

License Incomplete ? 0:00:00 Reset Trial View Modules

SYSTEMS

Overview

Performance

Alarm Pipelines

Gateway Scripts

Modules

Redundancy

Reports

SFCs

Voice Alarming

Tags

EAM Tasks

Transaction Groups

CONNECTIONS

EAM Agents

Databases

Designers

Devices

Gateway Network

Store & Forward

OPC Connections

OPC-UA Server

Vision Clients

DIAGNOSTICS

Execution

Logs

Threads

Systems Overview

Architecture

Gateway | Controller

Version: 7.9.0-rc2 (b2016092312)

License: partial

Uptime: 3 days

2% cpu

565 mb

No Redundancy

Add a redundant backup gateway to protect your system from downtime caused by failures.

No Gateway Network

Multiply the power of your Ignition Gateways by combining them into an enterprise network. Streamline the administration, monitoring, deployment, and commissioning process into one central location.

Connections

Designer Sessions

1 open

Databases

1 / 1 connected

Gateway Network Connections

0 / 0 connections

OPC Connections

1 / 1 connected

Environment

Process Id	24120
Operating System	Windows 7 amd64
Java Version	1.8.0_91-b15
Local Time	4:16:28 PM
Available Disk Space	149gb / 298gb
Detected NICs	10.20.8.46

Systems

Performance	2% CPU 563mb
Redundancy	Not configured
Modules	21 installed
Tags	973 tags
Reports	1 scheduled
Alarm Pipelines	0 / 5 active
SFCs	0 running
EAM Role	Controller
EAM Tasks	0 running

NOTE

A **minor version** upgrade is signified if the last version number changes. (i.e. **7.9.1** to **7.9.3**)

Minor version changes are mostly bug fixes. These are minor changes to Ignition and will fall under your existing license.

You **do not** have to update your license.

A **major version** upgrade is signified if one of the first two version numbers change. (i.e. **7.8** to **7.9** or **7.9** to **8.0**)

Major version changes are very large changes to Ignition. These will add functionality or change the way some of the systems in Ignition work.

You **do** have to update your license.

Update your License

Let Us Know

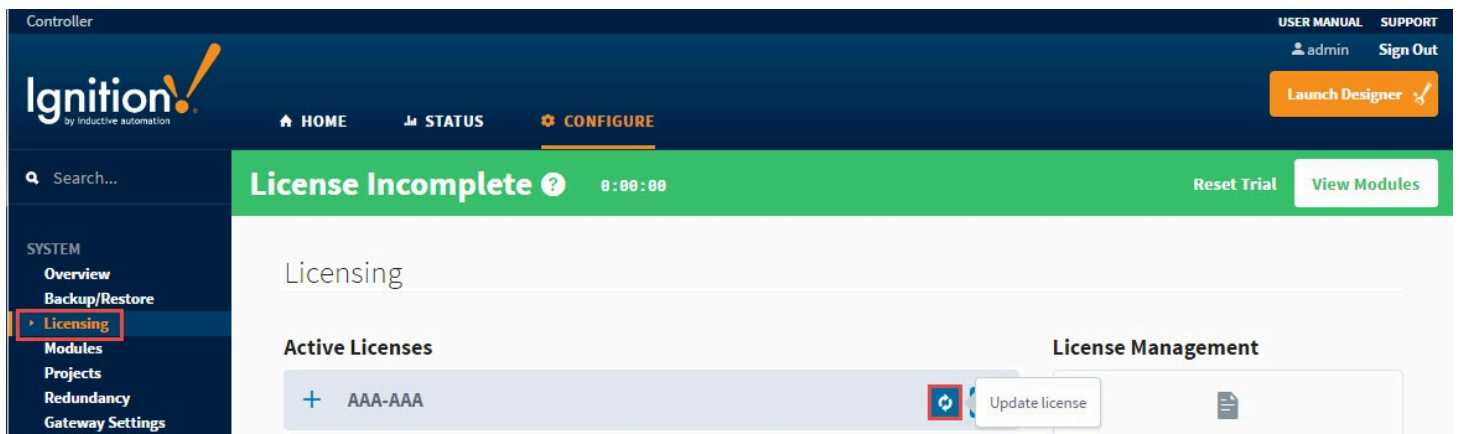
You can reach Inductive Automation by phone at **(800) 266-7798**.

Some major upgrades include new modules or have modified the functionality of preexisting modules. If your license needs to be updated, call us and we can guide you through it. If you are updating from one version to the next, or from 7.0.0 all the way to the latest version, we can talk you through what is different and what new functionality is available. See the [New in this Version](#)

Reload Your License

Once your license is updated in our system, go to the **Configure** section of the Gateway Webpage and click **System > Licensing**. Click on the **Update License** icon .

See our [Updating a License](#) page for a detailed guide. If you are coming from an older version, the gateway may look different. Don't worry, all this functionality is still there, but you may have to unactivate the reactivate instead if the **Update License** icon is missing.



Testing on a Development Server

One of the great things about Ignition is that you can run any Gateway in Trial mode (without a license). That means all you need to do is install on another computer and load your backup. More information on development servers can be found on the [Development Server Best Practices](#) page.

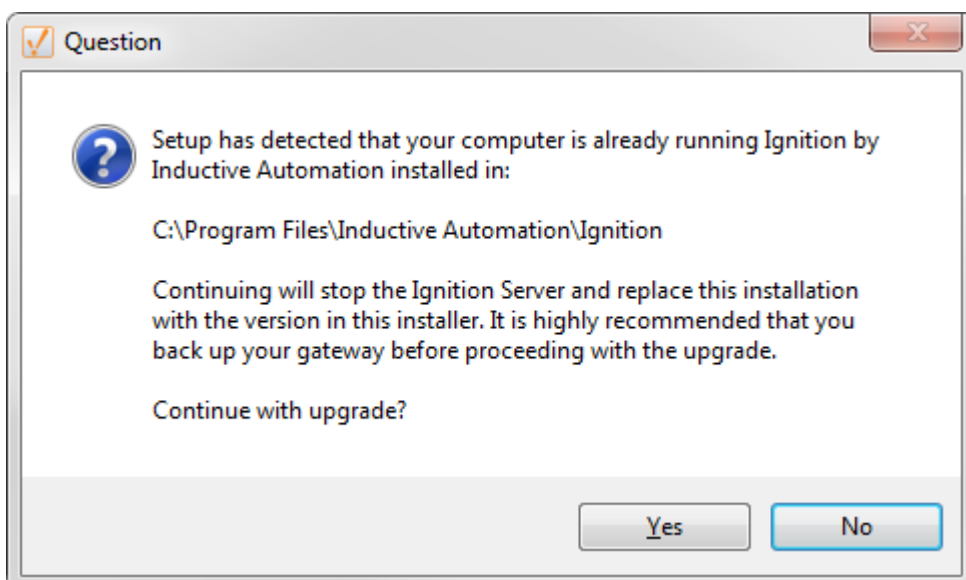
You can download any version of Ignition from the Inductive Automation [downloads archive](#) page.

Details on restoring from a gateway backup can be found on the [Gateway Restore](#) page.

Once you have a development server, you can freely test the new version without having an affect on your production server.

Upgrade the Production Server

Once you've made a backup and you're ready to upgrade the proudction server, simply run the Ignition installer. The Ignition installer doubles as an upgrader and will automatically check for an existing installation and upgrade any installed modules.



Upgrading a Redundant Pair

There are a few extra things to consider when upgrading a redundant pair of Gateways. When performing the upgrade locally, we recommend upgrading the Master first. This allows the Backup to take over when the Master is down for an upgrade, and the Master can resume operations when it comes back online. At this point, the Backup can be upgraded without interruption. The same considerations mentioned above should also be applied to the redundant pair, but you only need a Gateway Backup file from the Master Ignition Gateway.

If the upgrade will be performed remotely using the Enterprise Administration Module, then the Backup Gateway should be upgraded first. This allows the EAM controller to manually tell clients to communicate with the newly upgraded Backup Gateway, even though the Master will be a different version, allowing time to upgrade the Master.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Development Server Best Practices

Overview

Many users find a Development Server useful so changes can be tested in a safe environment before being pushed to production. This page details several recommendations and issues to consider when utilizing a Development Server.

Device Considerations

- The Development Server should have its own PLCs. This prevents the Development Server from accidentally writing to a Tag that the Production Server is connected to.
- Device connection names should be the same on both the Development and Production Servers. **OPC Tags** typically reference the name of the Device Connection in their OPC Item Path, so using the same name on both servers means that Tags can easily be migrated from one server to the other.
- Using PLC simulators is an alternative to the Development Server having its own PLCs. Generally this means finding simulators for your specific devices outside of Ignition.

Licensing Considerations

- The Development Server does not require a licensed Ignition Gateway, but one may be necessary depending on your testing requirements. All modules can run on the trial for a two-hour testing window, and may be started an unlimited number of times. If you want a more flexible test system for your Development Server, you can add a license to it.

Database Considerations

- The Development Server should have its own database, separate from the production database. This is especially useful when designing screens that manually modify a database table. For more information, see our [Database Connections](#) page.
- The names of the database connections used by the Development Ignition Gateway should match the names of the connections on the Production Server. Many system functions that interact directly with the database, such as `system.db.runPrepUpdate`, take the name of the database as a parameter, so using similar names for connections on both servers means you will not have to modify the script once it has been moved to the Production Server.

Network Considerations

- Some users prefer to place their Development Server on a separate network from the Production Server. This way there can be no confusion as to which server is being accessed. Keep in mind that Clients can only be launched on panels/computers that have network access to the host Gateway, so how this development network is configured will impact where testing clients can be launched. For more information, see our [Gateway Network](#) page.
- If you are restoring a copy of your Production Server on your Development Server, you will need to be cautious: two copies of the same gateway, on the same network, will both have access to all PLCs and Databases. In this case it is possible for duplicate historical records to be generated, amongst other potential problems. To prevent this, we recommend keeping the Development Server on a separate network. Alternatively, you could utilize the **Restore Disabled** checkbox when restoring on the Development Server. Once the disabled Development Gateway is up and running, you can edit each connection and point them to testing equipment.

Transferring Resources from Development

- When you are ready to move resources from the Development Server to the Production Server, Gateway Backups should not be utilized as they will include connections to the development resources. Instead, use the Project Exports and Tag Exports functionality to transfer resources.
- In cases where there are multiple Ignition Servers (Production or Testing), the [Enterprise Administration Module](#) can make the resource migration easier.

Cloning a VM

Sometimes it may be useful to clone a VM running an Ignition Gateway to keep all the configurations the same in the new system. However, there are a few things to keep in mind when doing this.

- The cloned VM should be isolated from the network at first. Especially when you have a Gateway Network setup with multiple Gateways communicating with each other or Tag History being stored to a database, having two Gateways with the same name and IP address can cause problems for a system which won't be able to differentiate between the two.
- The new VM should be given a different IP address. If you have a static IP address setup for the VM, the clone should get its own unique IP address.
- The Gateway System Name needs to be changed. This is done by going into the Gateway Webpage, navigating to the Gateway Settings page in the Configure section, and changing the System Name to be different. Note, that this name should be different from all Gateways in the network, and is important when using Tag History, as the Gateway name helps the historian system differentiate multiple Gateways storing to the same database.
- Within the data folder located in the Install Root, there is a file titled **".uuid"**. This file needs to be deleted. Once it has been removed, the Gateway can be restarted, which will automatically generate a new file.
- After all of the points above have been handled, the VM can then be hooked back up into the network and used like a new Gateway.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

Quick Start Guide

Are you new to Ignition? Here are a few simple steps to get going right away! In very little time, you can get a sample project up and running.

1. Download and Install Ignition

Installation is simple and can be done in under 3 minutes. Let's get started and install now.

2. Launch the Gateway

The Ignition Gateway is a web server. When it is running, you access it through a web browser in just 2 steps.

3. Connect to a Device

Ignition can connect to many different devices natively with the built-in OPC-UA Server.

4. Connect to a Database

After you install your database, you can connect it to Ignition. To get detailed descriptions for a number of differ...

5. Open the Designer

Now that we have set up a device and database connection, let's open the Designer and Create a project.

6. Create Tags

With the Designer open, a great first step is to create some Tags. You'll use these tags for realtime status and con...

7. Add Components to a Window

You can drag and drop components onto your project windows. Components can do everything from displaying t...

8. Launch Clients

Now that we've created a project, let's launch a Client to see it in action.

9. Add History to Tags

Storing history for your Tags is simple, and once that's done we can display it on screen in a variety of ways. Let'...

10. Create a Graph

You can see the Tag History data we just created by adding a graph to a window. Your graph can include subplots...

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

1. Download and Install Ignition

Installation is simple and can be done in under 3 minutes. Let's get started and install now.

Windows

Mac OS X

Linux

BEFORE YOU INSTALL IGNITION, MAKE SURE YOU MEET THE MINIMUM SYSTEM REQUIREMENTS

- Java 8+ (server), Java 6+ (client)
- Dual-Core Processor (32 or 64 bit)
- 4GB RAM
- 10GB free HD space

Also, ensure you have Administrator privileges to run the Ignition installer.

1. Go to the Inductive Automation website at www.inductiveautomation.com
2. Select **Download Ignition** in the upper right corner of the page.
3. On the Ignition Downloads page, scroll down to **System Installers** and select the **ignition-X.X.X-windows-x64-installer.exe** installer.

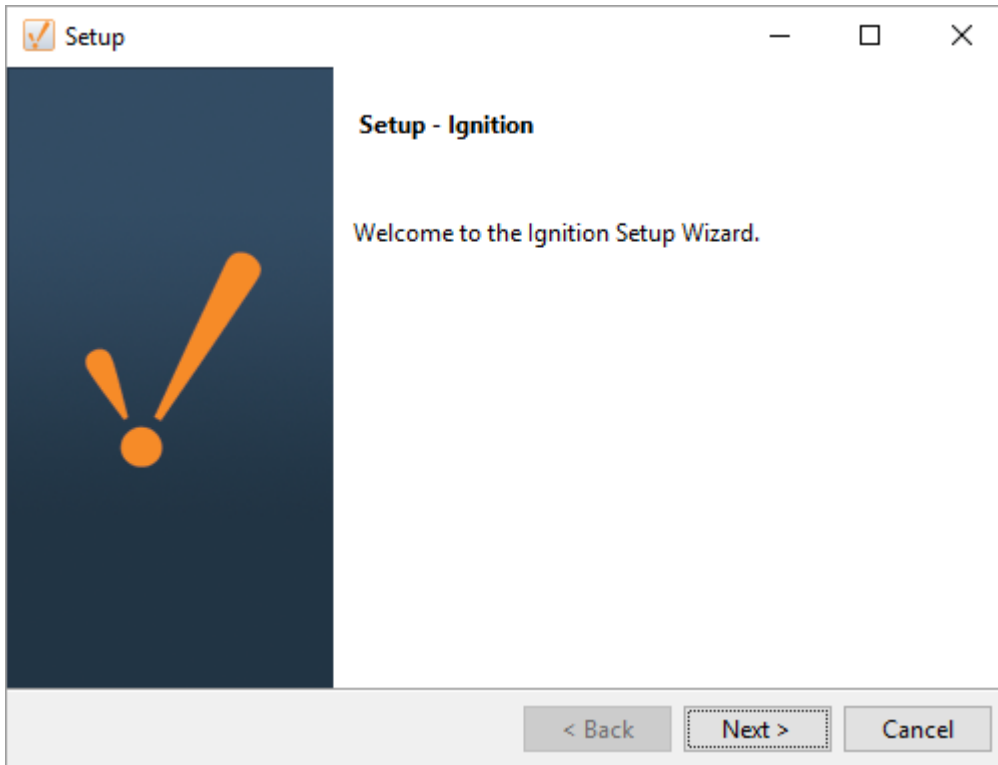
If your computer runs on a 32-bit Windows, click the **ignition-X.X.X-windows-32-installer.exe** installer.

NOTE

An error window is displayed if you try to install 64-bit Ignition on a 32-bit Windows system. You are allowed to install 32-bit Ignition on a 64-bit system (however, this is not recommended).

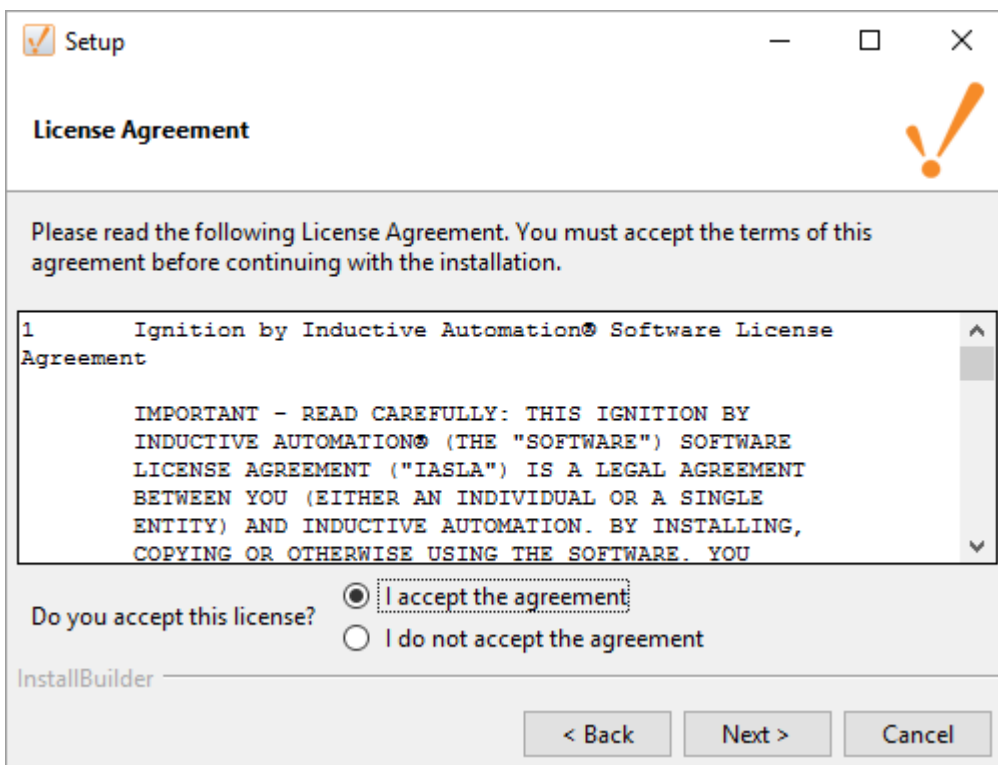
4. After the installer downloads, double-click on the **.exe** executable file to start installation.

The Setup window welcomes you to the Ignition Setup Wizard.



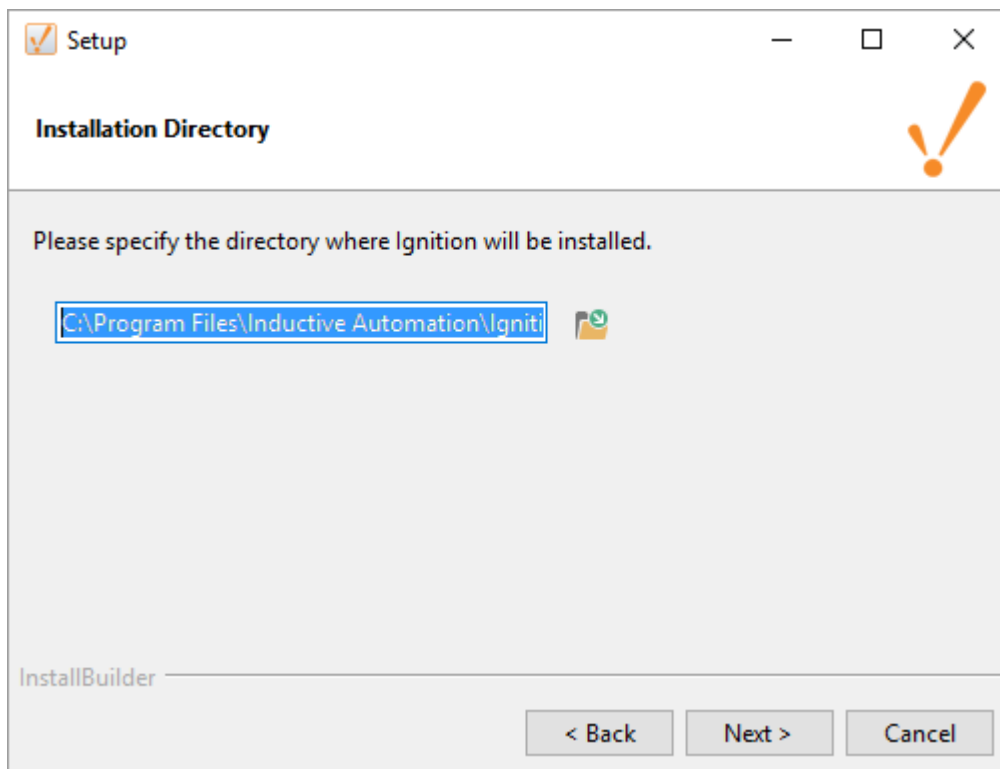
5. Click **Next**.

6. Click the **I accept the agreement** button and click **Next**.



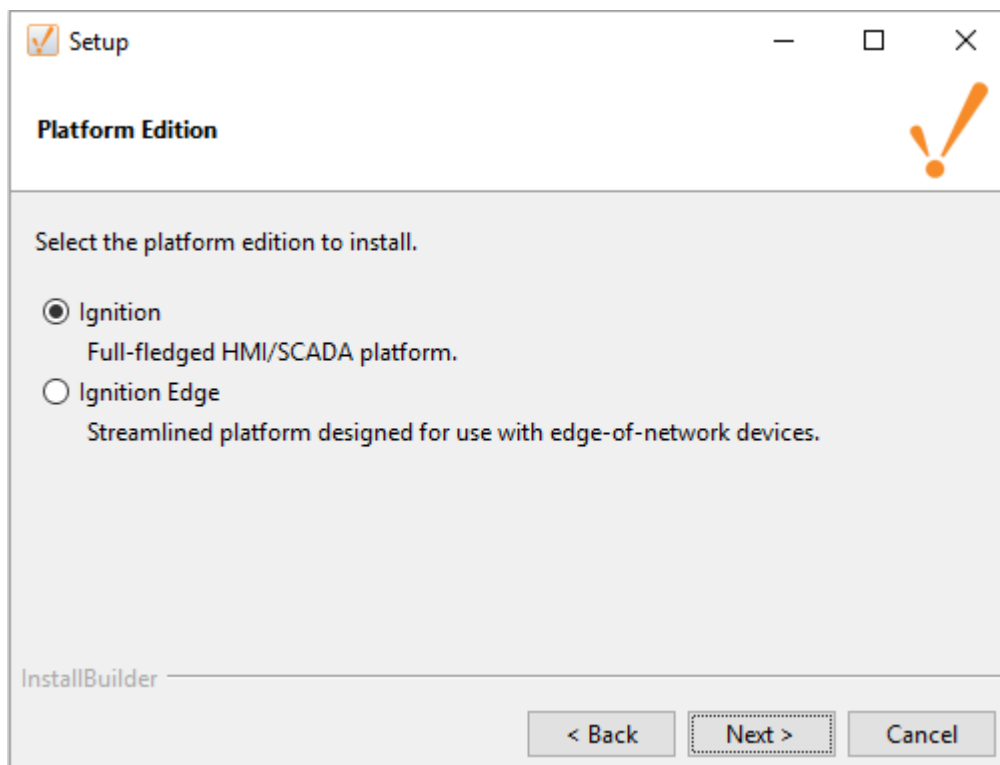
The Installation Directory window is displayed. By default, Ignition is installed under your Program Files directory. This is usually a good choice.

7. Make a note of the default installation directory, **Program Files/Inductive Automation/Ignition**, accept the default Installation Directory, and click **Next**.



The **Installation Mode** window is displayed.

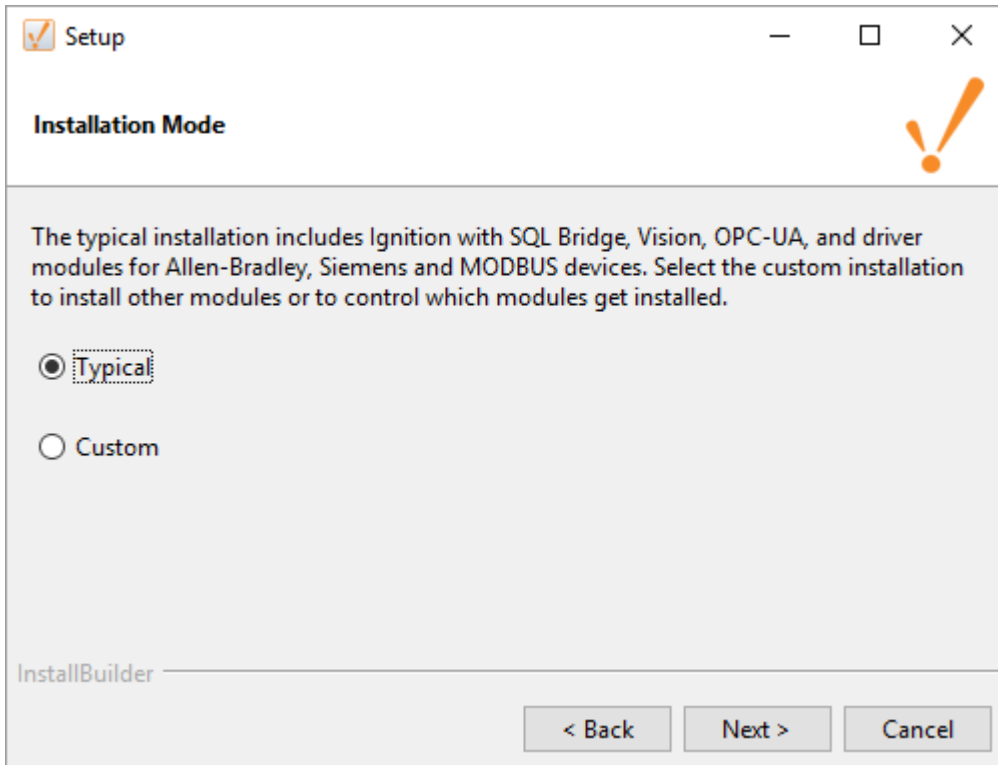
8. Select the platform you wish to install. The rest of this guide will use **Ignition** as the platform.



9. Select either the **Typical** or **Custom** mode.

The **Typical** installation includes Ignition with all of the HMI/SCADA modules. The only modules that are not installed by default are the MES modules.

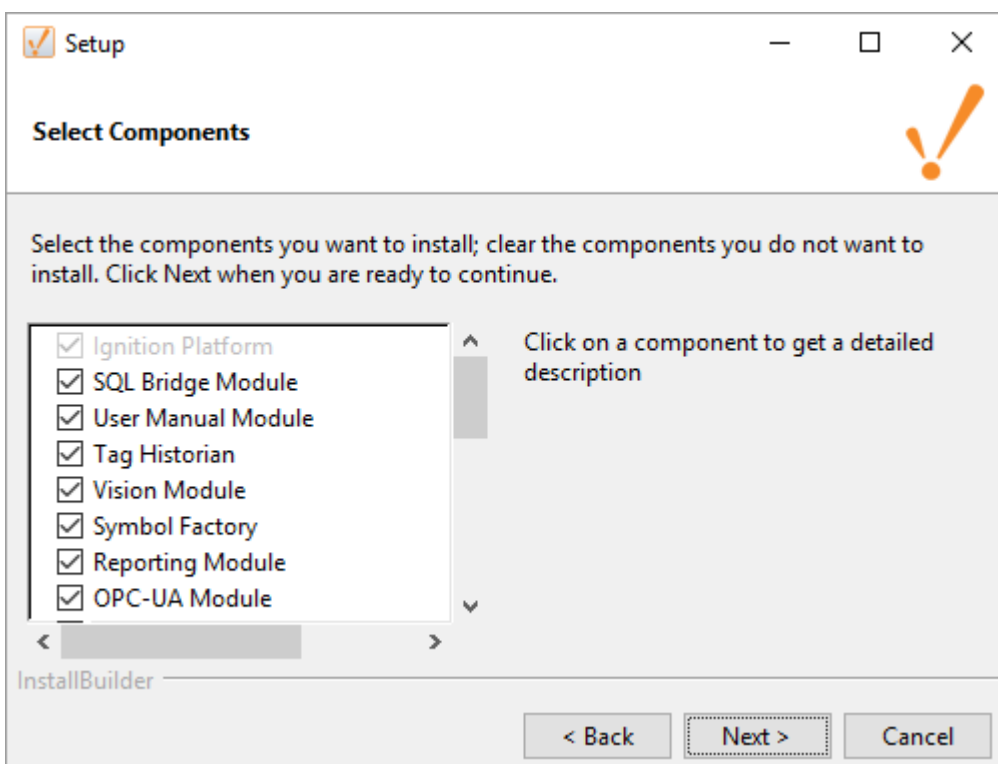
The **Custom** installation lets you control and individually select the modules that you want installed.



10. Click **Next**.

The Setup window is displayed.

If you selected Custom mode, in the Setup window you can select the check box next to the modules you want to install as part of the Ignition installation. To see a brief description of the module, click on the module name.



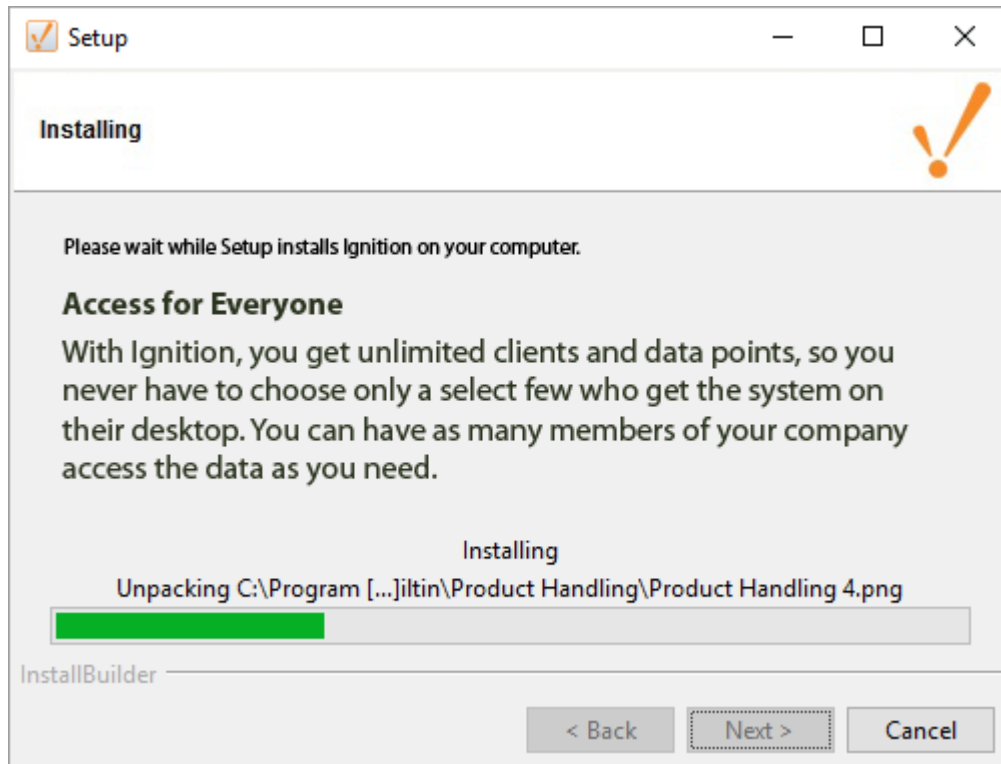
11. Click **Next**.

The Setup Wizard shows Ignition is now ready to install.

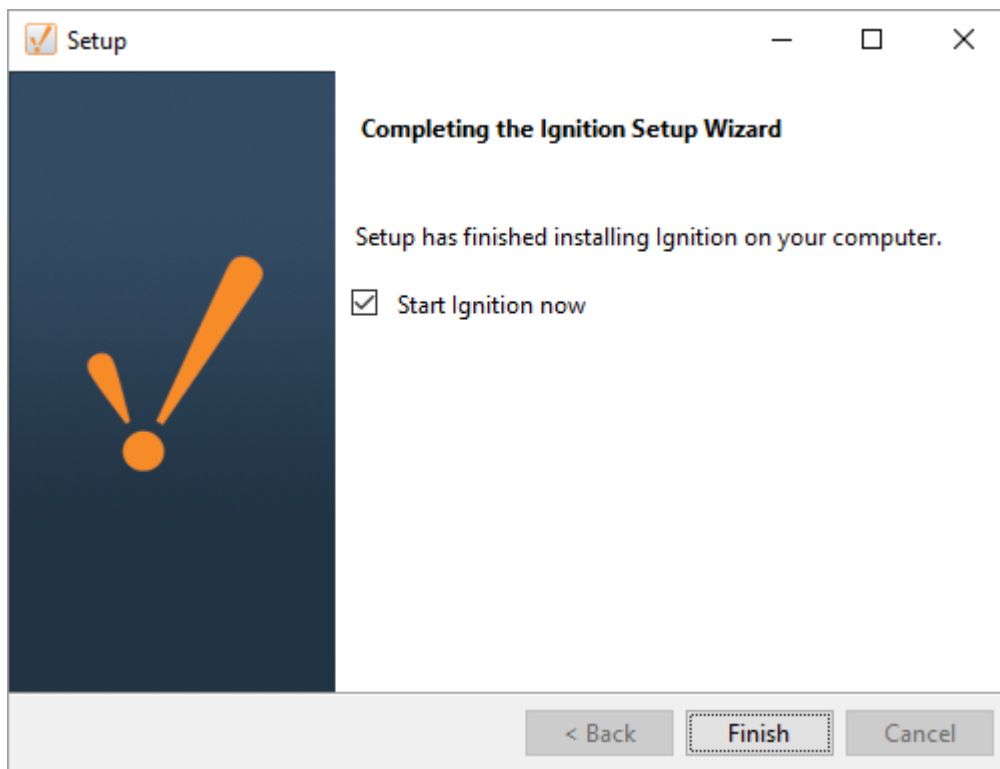
If you decide to change your selections, click **Back** to go back and make your changes.

12. Click **Next**.

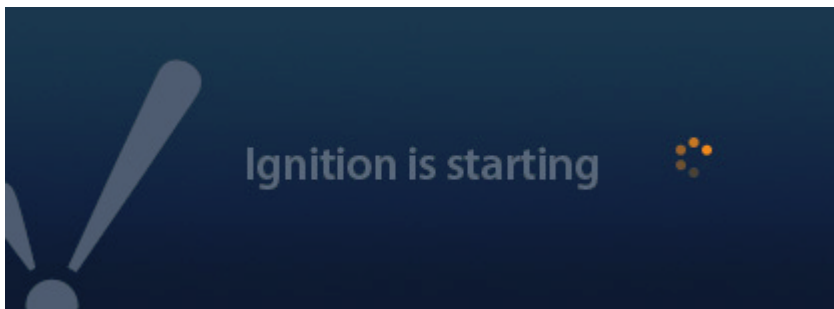
Ignition will install the selected modules.



13. After the Setup Wizard shows Ignition installation is finished, make sure that the check box for **Start Ignition now** is selected, then click **Finish**.

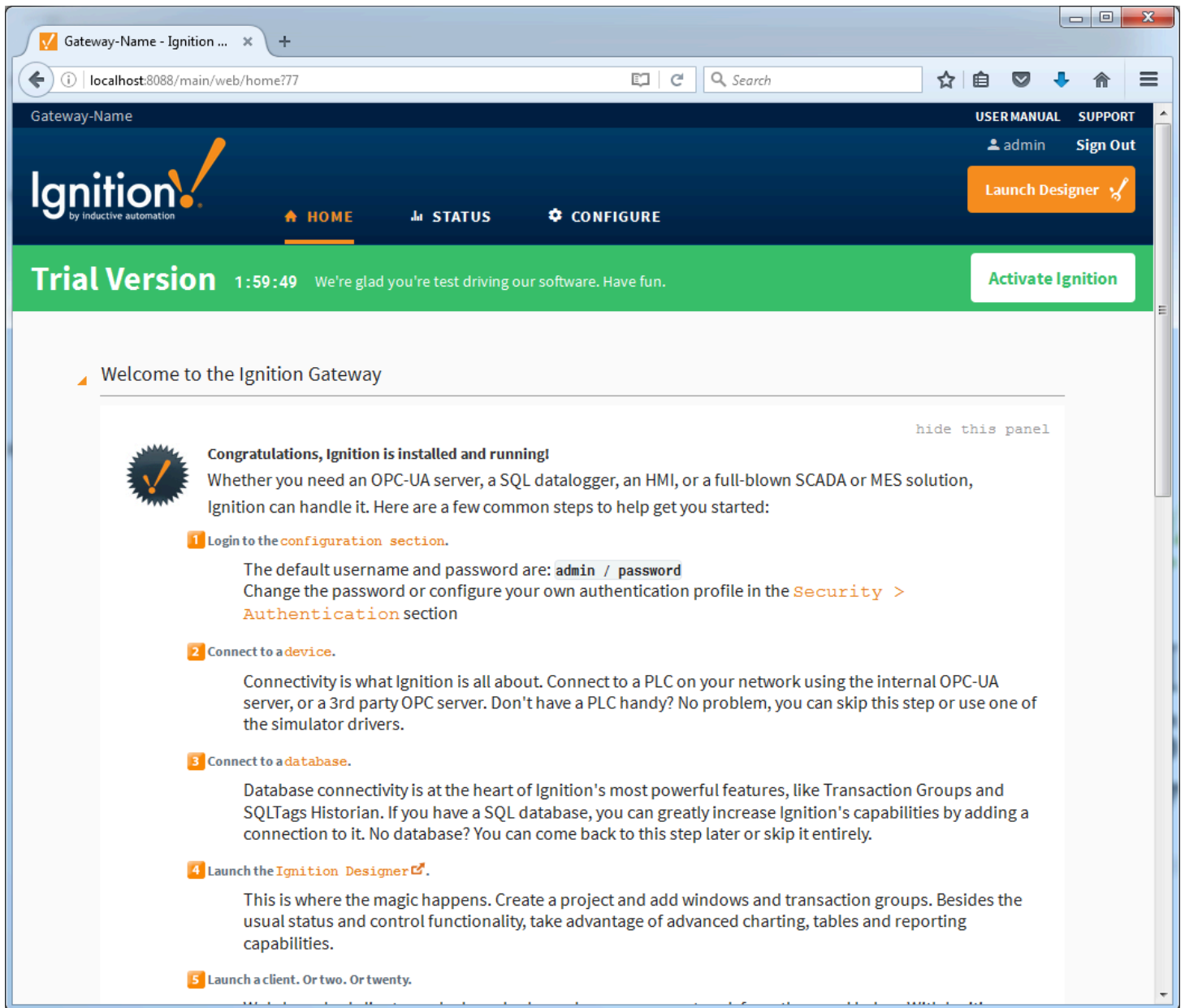


A splash screen informs you that Ignition is starting. This may take a few minutes to finish.



That's it! Ignition Installation is complete and the Gateway is then launched.

Your web browser opens the Gateway home page at **http: //localhost:8088**



Ignition installs itself as a Windows Service (**Start > Control Panel > Administrative Tools > Services > Ignition Gateway**), so it starts automatically when your computer starts up.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

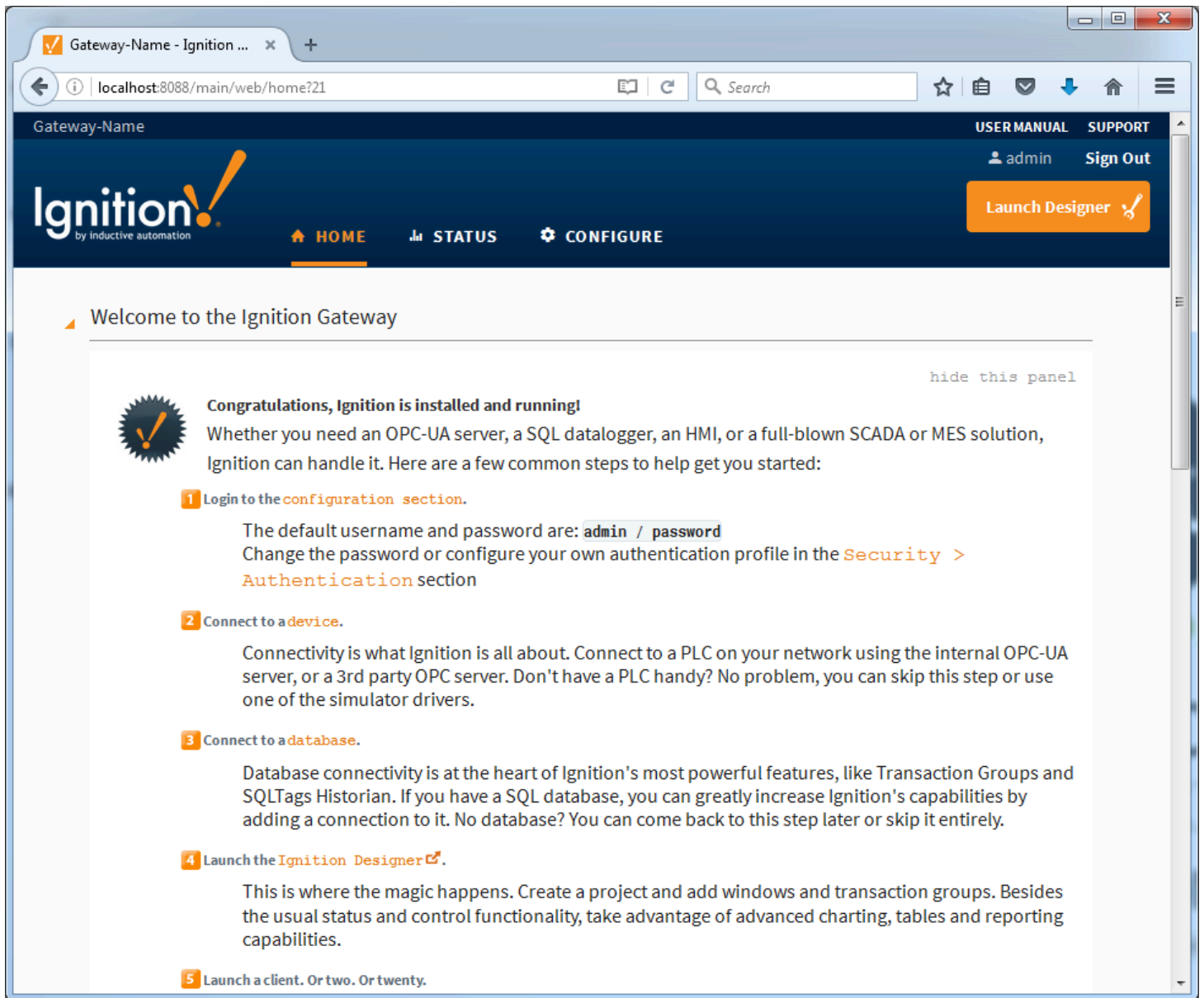
For documentation on the latest stable release, see the [latest version](#) (8.3).

2. Launch the Gateway

The Ignition Gateway is a web server. When it is running, you access it through a web browser in just 2 steps.

The default login credentials for the Gateway are: *username:* **admin** *password:* **password**

1. Figure out the IP address of the computer you installed Ignition on. If this computer is the one Ignition is installed on, then you can use **localhost** instead of the IP.
2. Open up a web browser and go to this address: <http://ipaddress:8088> and it will bring up the Gateway Homepage. If you are using localhost, it will look like this: <http://localhost:8088>



The first time you go to the Gateway Homepage, it shows you 5 common steps to help you get started. You can follow along with these steps and/or with this step-by-step walk through, they follow the same basic workflow. [Learn more about the Gateway Webpage here.](#)

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

3. Connect to a Device

Ignition can connect to many different devices natively with the built-in OPC-UA Server.

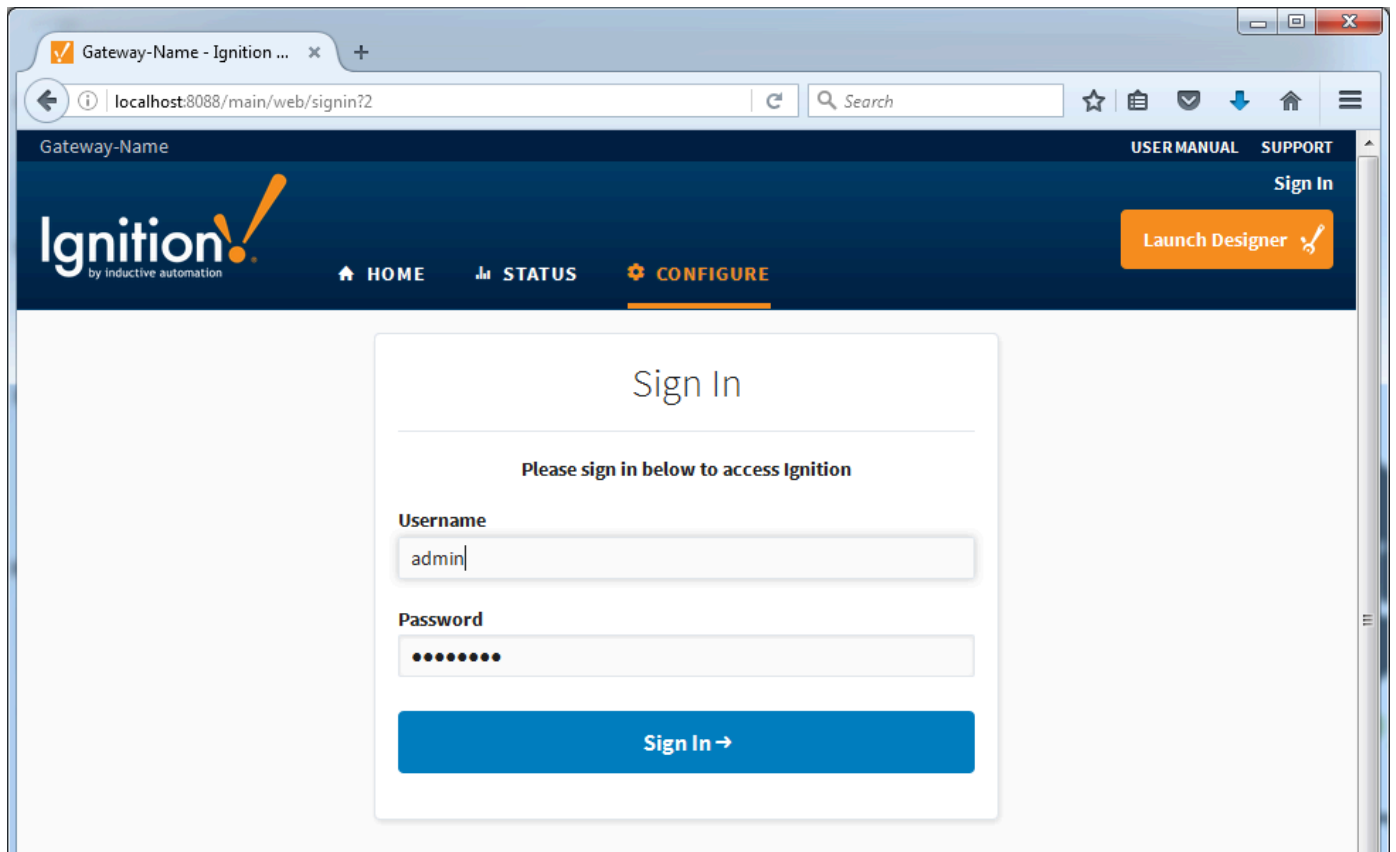
- [Allen Bradley Ethernet](#)
- [Modbus](#)
- [Siemens](#)
- [UDP and TCP Driver](#)
- [DNP3](#)
- [Omron](#)
- [Simulators](#)
- [Third Party OPC Servers](#)

Connect to the Generic Simulator

1. From the Gateway Webpage, navigate to the Configure section.
2. Enter the default username and password for the Designer and click **Login**.

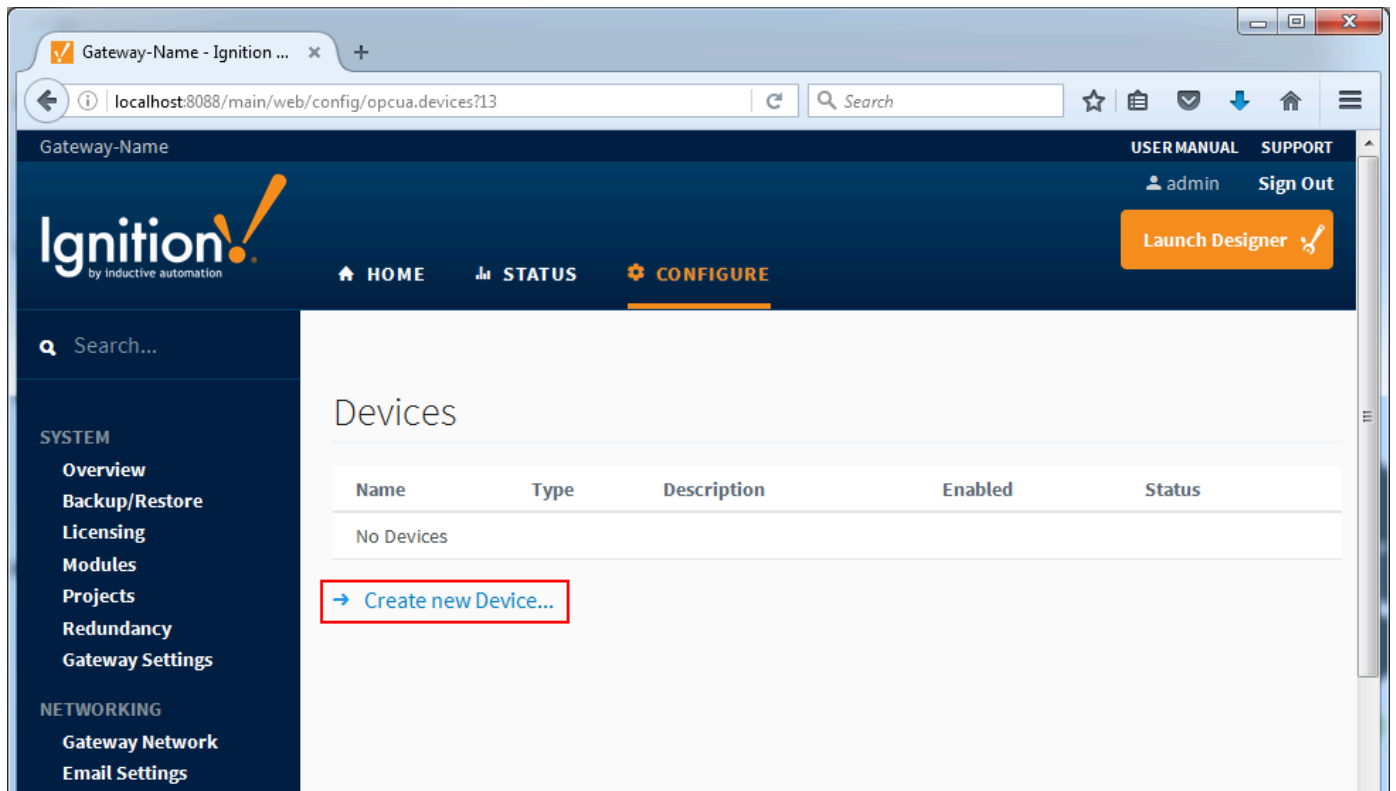
Username: **admin**

Password: **password**



3. Scroll down to the **OPC-UA** section on the left and click the **Devices** link.

4. To add a new Device Connection, click the **Create new Device...** link.



5. Select **Simulators Generic Simulator** and click **Next**.

6. Give the new connection a name and click **Create New Device**.

Devices

Name	Type	Description	Enabled	Status		
Generic	Simulators	Generic Simulator	true	Connected	delete	edit

[→ Create new Device...](#)

Learn more about Devices

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

4. Connect to a Database

After you [install your database](#), you can connect it to Ignition. To get detailed descriptions for a number of different database connections, see [Connecting to Databases](#).

Database Connections

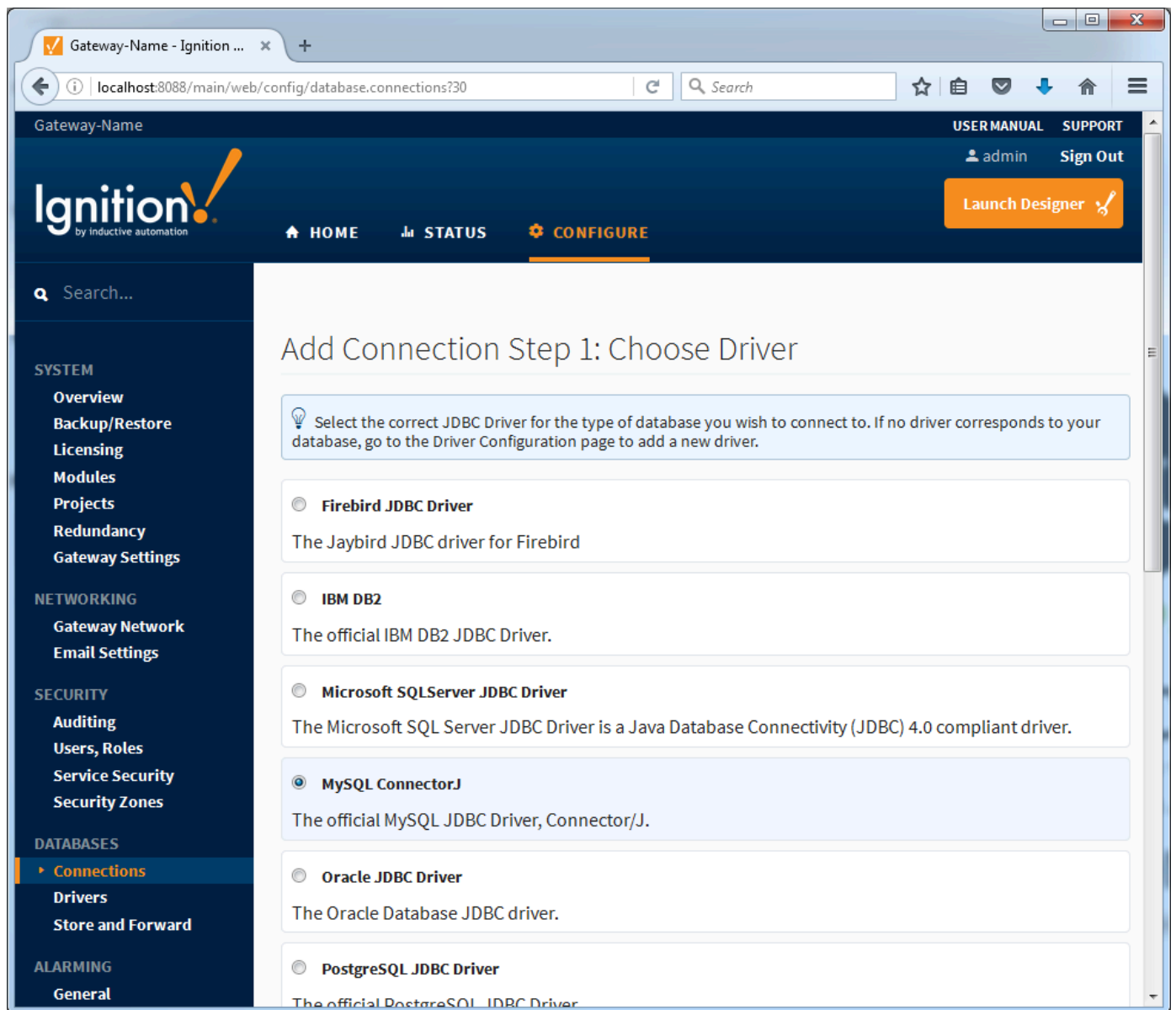
Name	Description	JDBC Driver	Translator	Status	
MySQL		MySQL ConnectorJ	MYSQL	Valid	delete edit

[→ Create new Database Connection...](#)

Note: For details about a connection's status, see the [Database Connection Status](#) page.

Connect to MySQL

1. Log into the Configure section of the Gateway Webpage just like in the previous sections.
2. Navigate to the Databases **Connections** page.
3. Click the **Create new Database Connection...** link.
4. Select the **MySQL ConnectorJ** option and click the **Next >** button.



5. Enter a Name, Connect URL, username, and password. If you just installed MySQL on the same machine as Ignition you will be able to use the following credentials:

Name: **MySQL**

Connect URL: **jdbc:mysql://localhost:3306/test**

Username: **root**

Password: [whatever password you entered when installing MySQL]

Scroll to the bottom and click the **Create new Database Connection** button.

Gateway-Name - Ignition ...

localhost:8088/main/web/config/database.connections?31

Search

☆📁🔒⬇️🏠☰

Search...

SYSTEM

Overview

Backup/Restore

Licensing

Modules

Projects

Redundancy

Gateway Settings

NETWORKING

Gateway Network

Email Settings

SECURITY

Auditing

Users, Roles

Service Security

Security Zones

DATABASES

Connections

Drivers

Store and Forward

ALARMING

General

Journal

Notification

On-Call Rosters

Schedules

TAGS

History

Realtime

OPC-UA SERVER

Certificates

Devices

Settings

OPC CONNECTIONS

Servers

New Database Connection

Main Properties

Name	MySQL Choose a name for this database connection.
Description	
JDBC Driver	MySQL ConnectorJ The JDBC driver dictates the type of database that this connection can connect to. It cannot be changed once created.
Connect URL	jdbc:mysql://localhost:3306/test The Connect URL is JDBC-driver specific. It usually contains the address of the machine that the database is running on. The format of the MySQL connect URL is: <code>jdbc:mysql://host:port/database</code> With the three parameters (in bold) host: The host name or IP address of the database server. port: The port that the database server is running on. MySQL default port is 3306. database: The name of the logical database that you are connecting to on the MySQL server.
Username	root
Password	•••••
Password	••••• Re-type password for verification.
Extra Connection Properties	zeroDateTimeBehavior=convertToNull;connectTimeout=120000;socketTimeout=120000; There is an extensive list of extra connection properties available for MySQL Connector/J. See the documentation for a table describing all connection properties.

Learn more about Database Connections

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

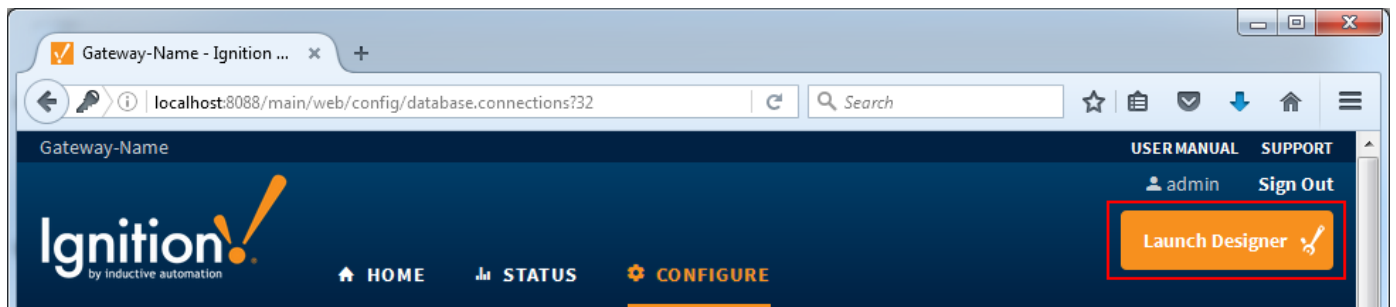
For documentation on the latest stable release, see the [latest version](#) (8.3).

5. Open the Designer

Now that we have set up a device and database connection, let's open the Designer and Create a project.

Launch the Designer

1. On the Gateway Webpage, click the **Launch Designer** button in the upper-right of the page.



Depending on your web browser, it may ask you if you want to Keep or Discard a **designer.jnlp** file. If it does, tell it to keep the file.

If it doesn't start automatically, run the **designer.jnlp** file to start the Designer.

TIP

If you are using Chrome, you can do this by clicking the file in the lower-left of the window or go to your downloads folder and double-click the jnlp file to run it.

The Designer starts up and the login window is displayed.

2. Enter the default username and password for the Designer and click **Login**. It is the same as for the Gateway Configuration section:

Username: **admin**

Password: **password**

NOTE

The default Username is set in the Configure section of the Gateway Webpage, under **Security > Users, Roles**.

Create a Project

The screenshot shows the 'Open/Create Project' window in Ignition. The window title is 'Open/Create Project'. The main area is titled 'New Project Setup'. It contains the following fields:

- Project Name:
- Project Title:
- Authentication Profile:
- Default Database:
- Default Tags Provider:
- Project Template:
- Description:

A 'Create New Project' button is located at the bottom right of the main area. The sidebar on the left has a 'Create New' section with a 'New Project' button highlighted by an orange border, and an 'Open Recent' section below it. The Ignition logo is in the top left, and the Inductive Automation logo is in the bottom right.

After logging into the Designer, the Open/Create Project window gives you two options, to create a new project or open an existing one. As we don't have any existing projects yet, we will create a new project.

1. In **Project Name**, enter a name for your new project (no spaces!).
2. If you **created a Database Connection**, you will see the Default Database field is already populated. This will be important when we get to storing history.

3. For Project Template, select a template from the drop-down list, for example, you can select the **Single-Tier Nav, West & North** template.

Rather than starting from a blank project, the templates let you start from one of the eight skeleton projects that already have navigation windows setup. These projects are completely customizable and are designed to help you get started faster.



4. Click on **Create New Project**.

The Designer will now launch and you can work on your project.

[Learn more about the Designer](#)

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

6. Create Tags

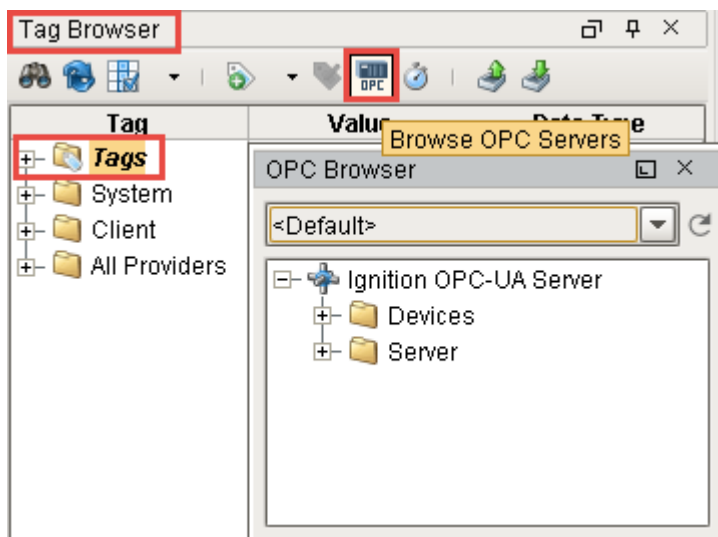
With the Designer open, a great first step is to create some Tags. You'll use these tags for realtime status and control and to store history with the Tags Historian. You can use Drag and Drop to bring Tags into Ignition with the Designer. Once there, Tags are all configured in the Tag Browser panel.

Click and Drag Tags into the Designer

If you have [created a device](#) in the earlier step, the easiest way to create some Tags is by dragging the tags from the OPC Browser.

On the left side of the Designer you will see 3 panels, the **Project Browser**, the **Tag Browser**, and the **Property Editor**.

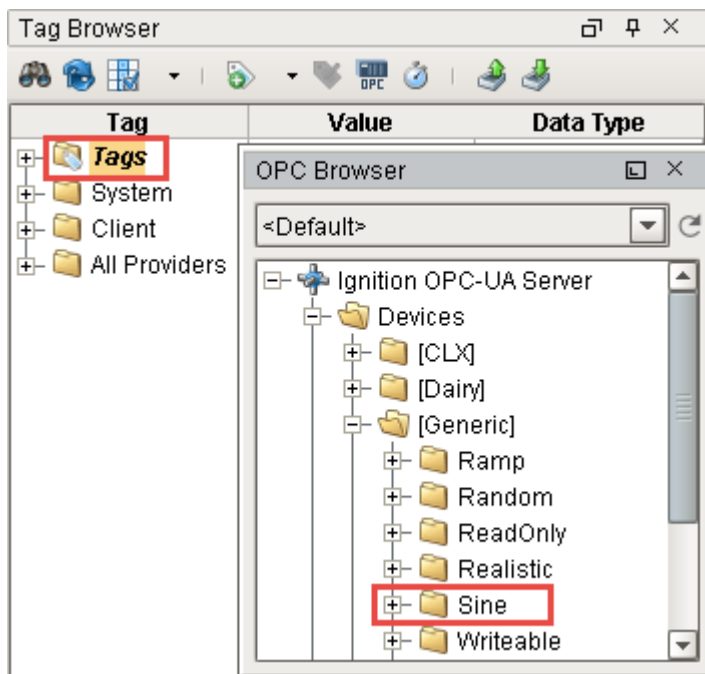
1. In the **Tag Browser**, select the **Tags** folder then click the **Browse OPC Servers** icon.



The OPC Browser is displayed and you can browse all of your OPC connections.

By default, you've got a connection to the internal **Ignition OPC-UA Server**, which has any devices in it that you created earlier. Expand the Devices folder and find some tags that you're interested in.

2. From the **OPC Browser**, highlight the Tags you want (for example, as in the picture below, the entire **Sine** folder is selected) and drag them to the **Tags** folder in the **Tag Browser**.



You can drag individual Tags or whole folders. When you drag folders, Ignition keeps the same hierarchy as the PLC.

3. That's it! You now have some Tags. Expand the **Tags** folder and the new **Sine** folder to see what you have added. You can see their values come in and start updating automatically. By default, they update at a 1 second rate.

Tag	Value	Data Type
Tags		
Data Types		
Sine		
Sine0	-86.11	Float8
Sine1	-0.59	Float8
Sine2	3.86	Float8
Sine3	8.52	Float8
Sine4	-1.48	Float8
Sine5	-86.11	Float8
Sine6	-0.59	Float8
Sine7	-16.14	Float8

[Learn more about Tags](#)

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

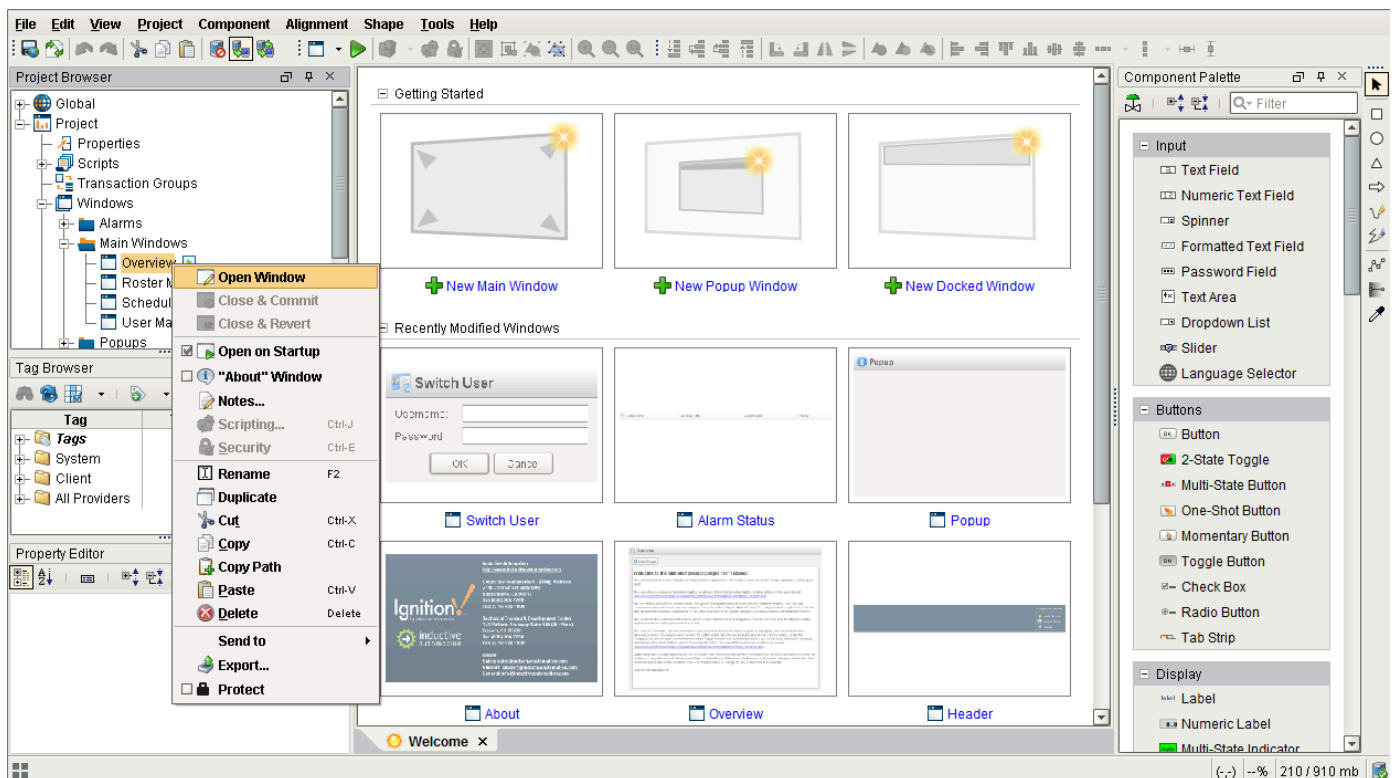
For documentation on the latest stable release, see the [latest version](#) (8.3).

7. Add Components to a Window

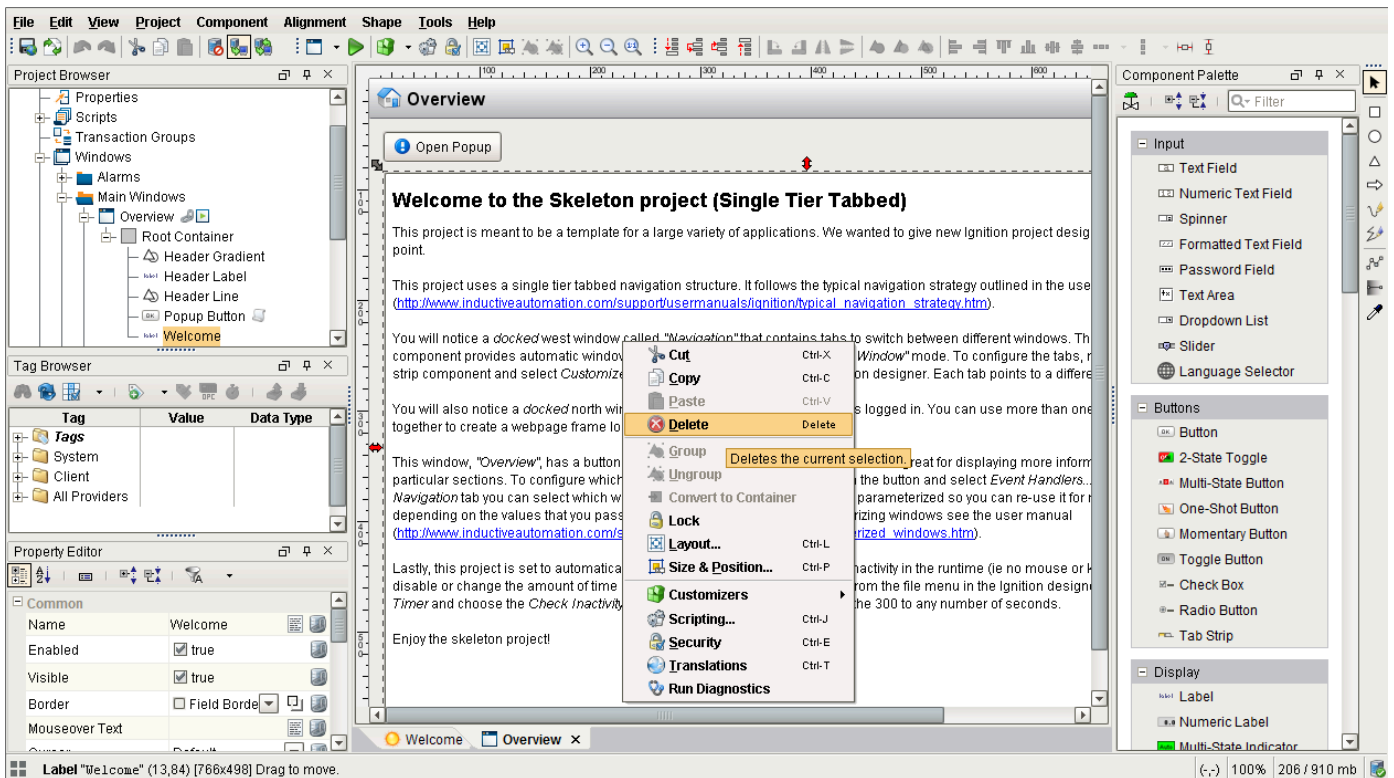
You can drag and drop components onto your project windows. Components can do everything from displaying text, to showing Tag values, to displaying reports and more. You can bind components to each other, to Tags, to values in the database, or any combination. Let's create a few components and show live values in the Client.

Add Components to Your Project

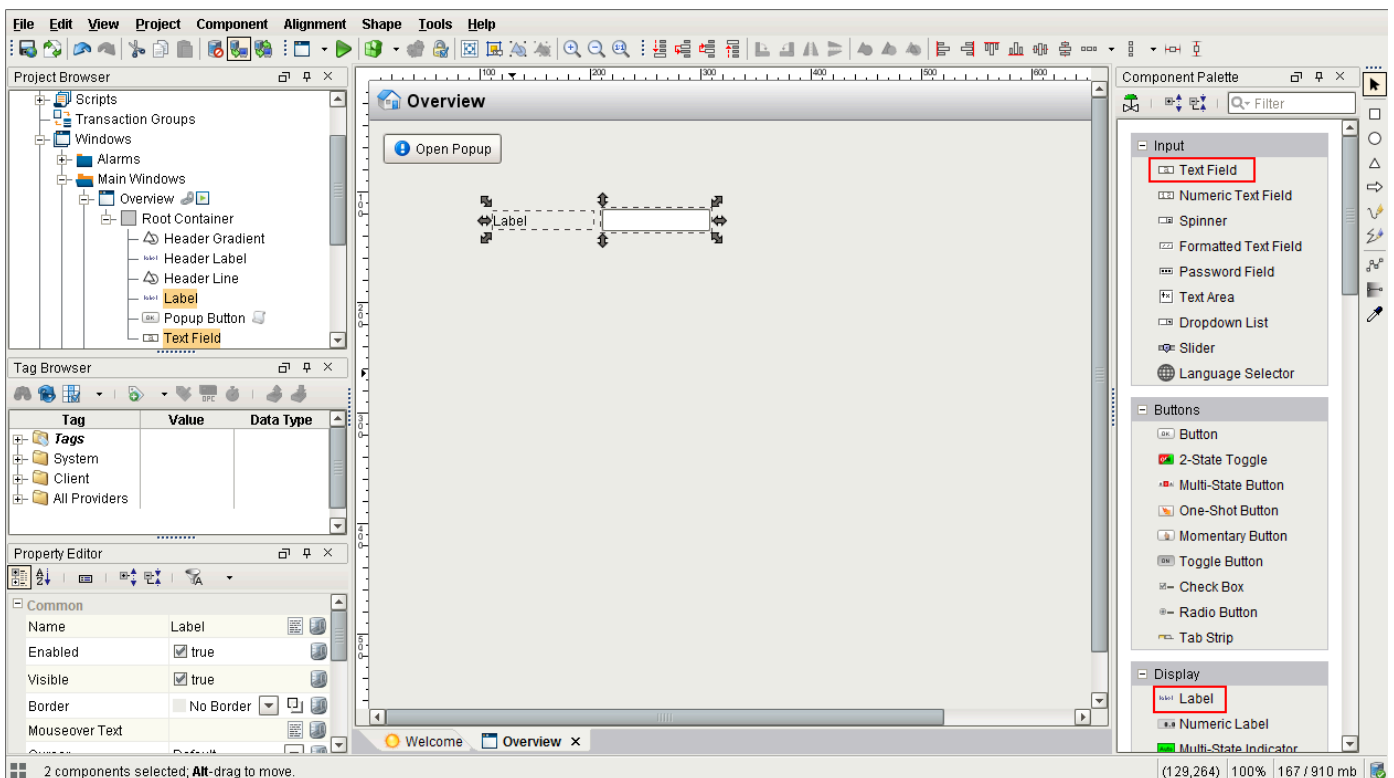
1. From the Designer, go to the **Project Browser** and right-click to open the **Main Windows -> Overview** window.



2. You can see there are already components on the window, let's make a little space. Select the large text Label and right-click to delete it.

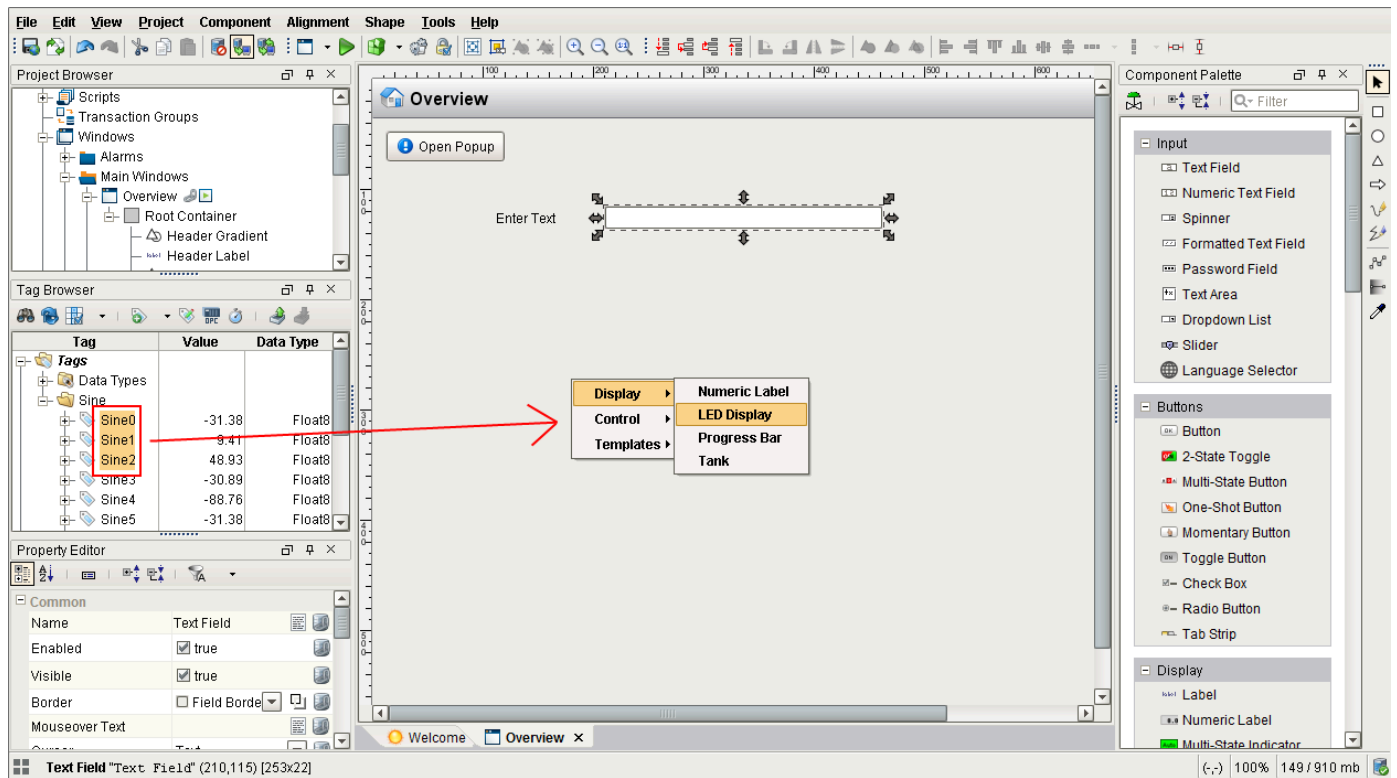


- Now we have some space to work with. You can start by clicking and dragging components from the **Component Palette** on the right onto the window. Let's add a **Text Field** and a **Label** component. Users can type in the text field.



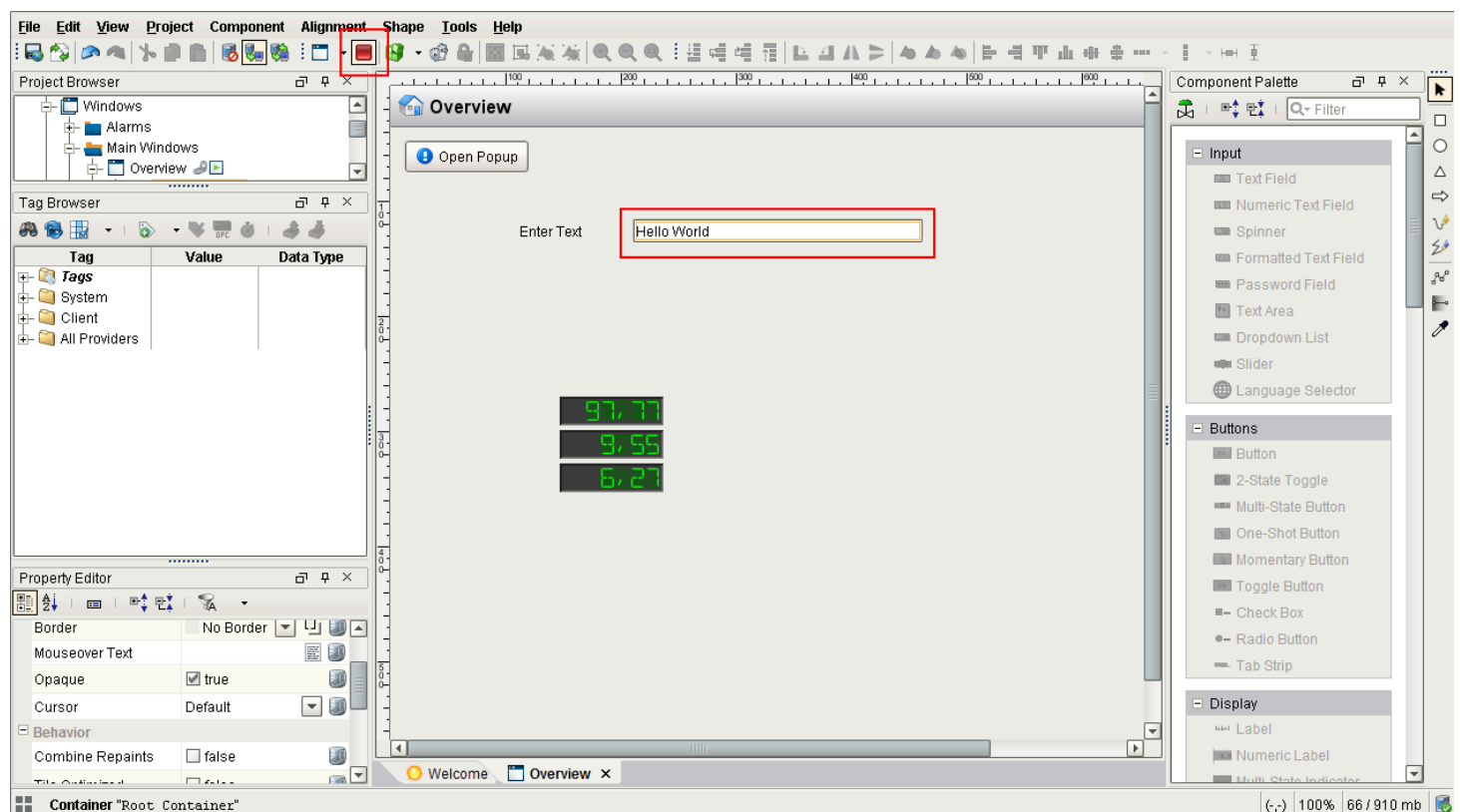
- Double-click on the **Label** component to type directly into it. I've renamed mine to say "Enter Text." You can also select a component and click-and-drag the arrow handles to resize them.
- Now let's add some Tag values to the window. The simplest way is to go into the **Tag Browser** and drill down to the Tag(s) you want and drag them onto the window. Select a few Tags and drag

them into the window. You will see a popup asking you what type of components you want to use to show these Tag values. Select **LED Display**.



6. Now you can see live values on your window! Don't forget to **save your project**.

You can Preview your project in the Designer by clicking the **Preview Mode** button. This allows you to interact with your windows as if you were using a Client. The Preview Mode button changes between a green triangle and a red square depending on which mode you are currently in. Look for the green triangle to go into preview mode.



Learn more about [Components](#) and [Property Binding Types](#)

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

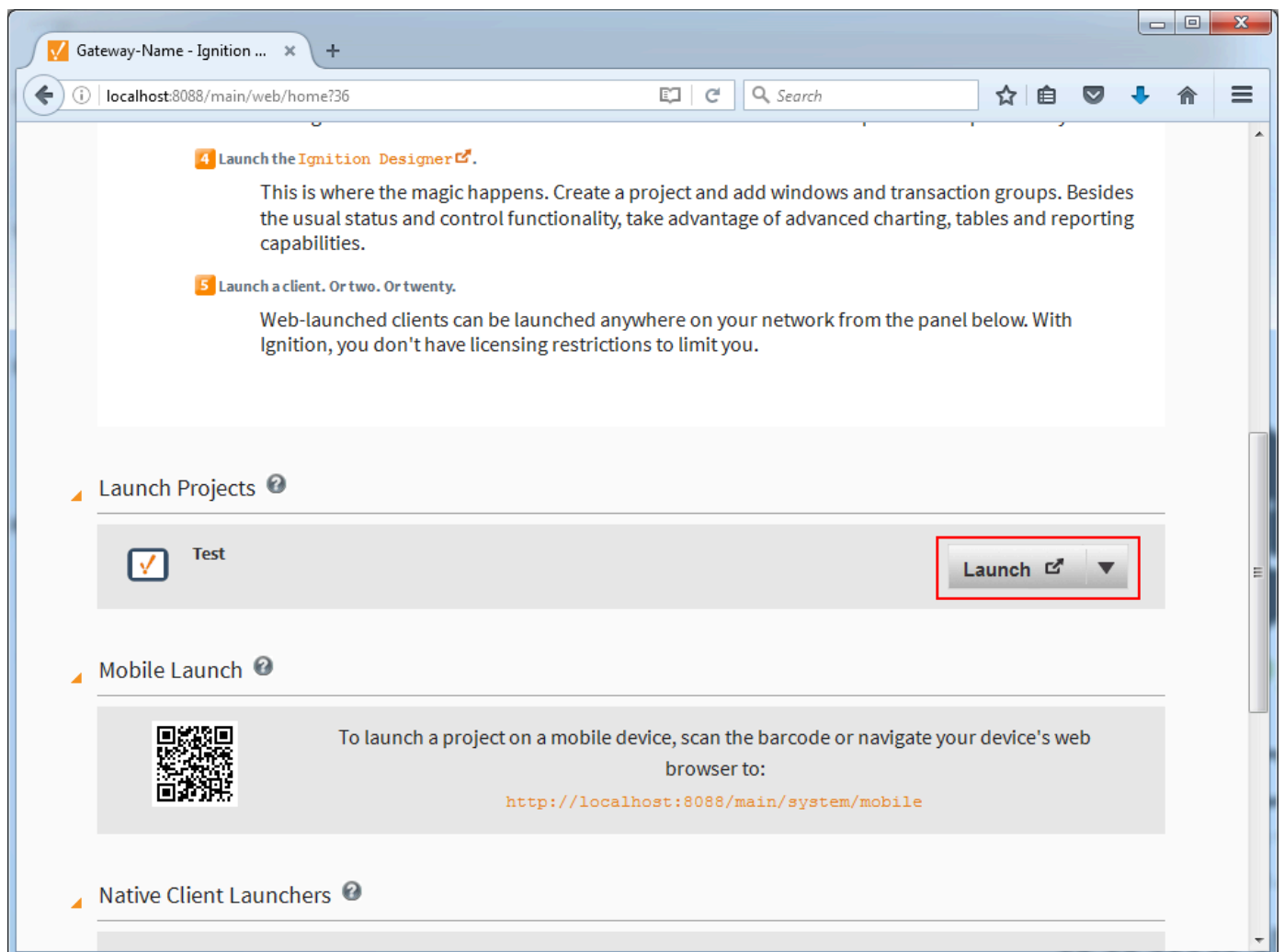
8. Launch Clients

Now that we've created a project, let's launch a Client to see it in action.

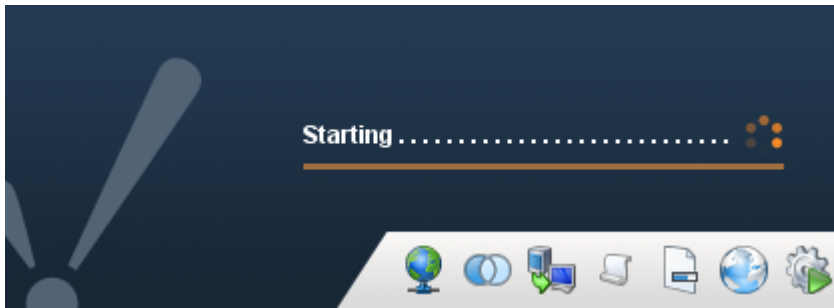
Launch a Client

1. If you haven't already, make sure you have a [project created](#).
2. Go back to the Gateway **Home** web page.

Your project will appear in the **Launch Projects** panel with a big **Launch** button to its right. Scroll down until you see it and Click on the **Launch** button to start up a Client.



A loading screen will pop up briefly before the Client opens.



3. Login to the Client using:

Username: **admin**

Password: **password**

By default, a new project uses the same user source as the Gateway, so the admin/password credentials will work.

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

For documentation on the latest stable release, see the [latest version](#) (8.3).

9. Add History to Tags

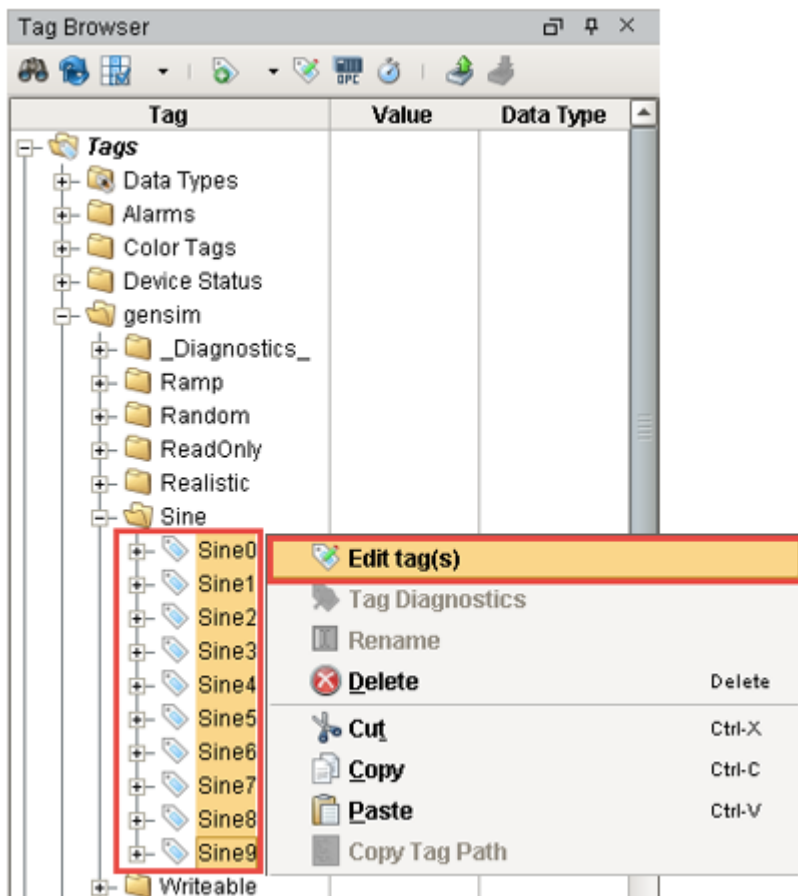
Storing history for your Tags is simple, and once that's done we can display it on screen in a variety of ways. Let's add history to some of our Sine Tags.

Add History to your Tags

1. In the Designer's **Tag Browser**, select one or more tags.

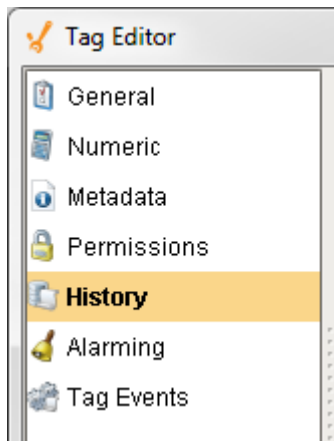
For example, select all the **Sine** tags in the Sine folder by holding the **Ctrl** key and clicking on several tags.

2. Right-click on the selected tags, and then select **Edit tag(s)**.



The Tag Editor window is displayed. Here, you can edit the tag and change the name, data type, scaling options, metadata, permissions, history, and alarming.

3. Click on **History** on the left side of **Tag Editor**.

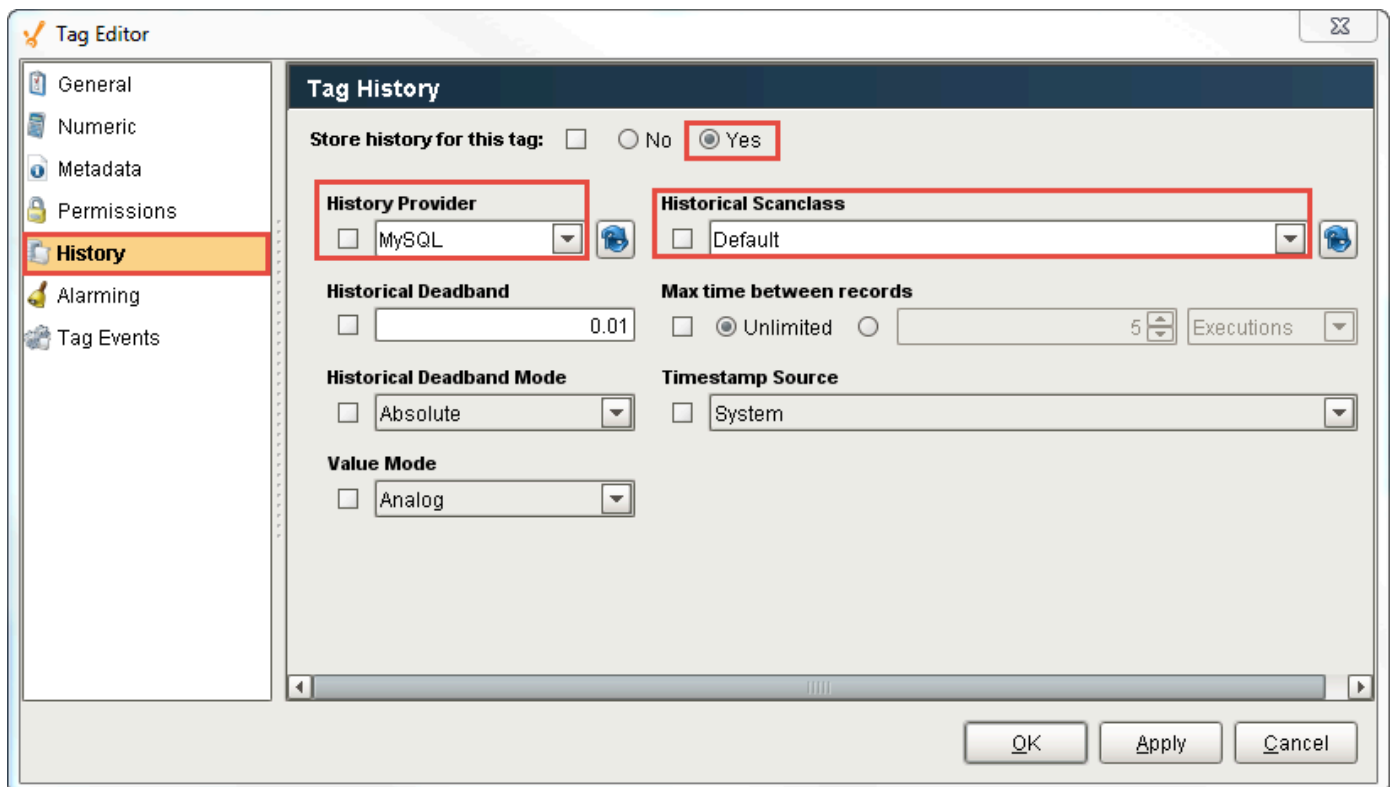


4. Click **Yes** for the **Store history for this tag** field. This will enable all of the fields.

5. Choose a database from the **History Provider** dropdown.

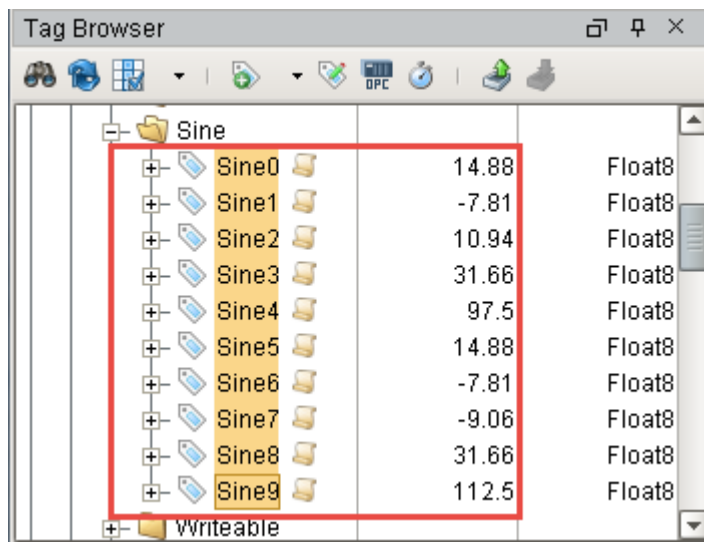
For example, choose **MySQL** as the database you want to log to.

6. Choose **Default** for **Historical Scansclass**. This sets the rate at which data is logged. This Default is set to 1 second which means data is logged every second.



7. Click **OK**.

Now look in the **Tag Browser**. To the right of each **Sine** tag that is storing history, an icon appears letting you know it is set up.



Tag Name	Value	Data Type
Sine0	14.88	Float8
Sine1	-7.81	Float8
Sine2	10.94	Float8
Sine3	31.66	Float8
Sine4	97.5	Float8
Sine5	14.88	Float8
Sine6	-7.81	Float8
Sine7	-9.06	Float8
Sine8	31.66	Float8
Sine9	112.5	Float8

If you were to look in your database, you can see all the tables and data Ignition has created for you.

[Learn more about the Historian](#)

This is documentation for **Ignition 7.9**, which is no longer actively maintained.

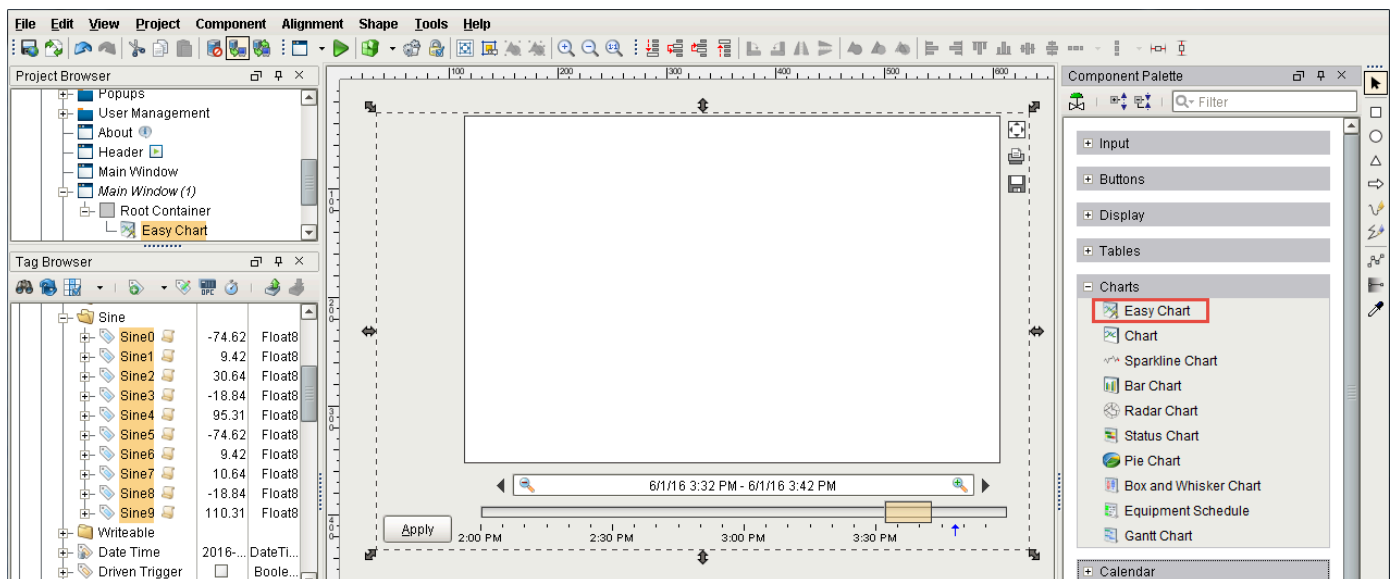
For documentation on the latest stable release, see the [latest version](#) (8.3).

10. Create a Graph

You can see the Tag History data we just created by adding a graph to a window. Your graph can include subplots, axes, and a moving average pen. You can view the runtime data in realtime or historical mode.

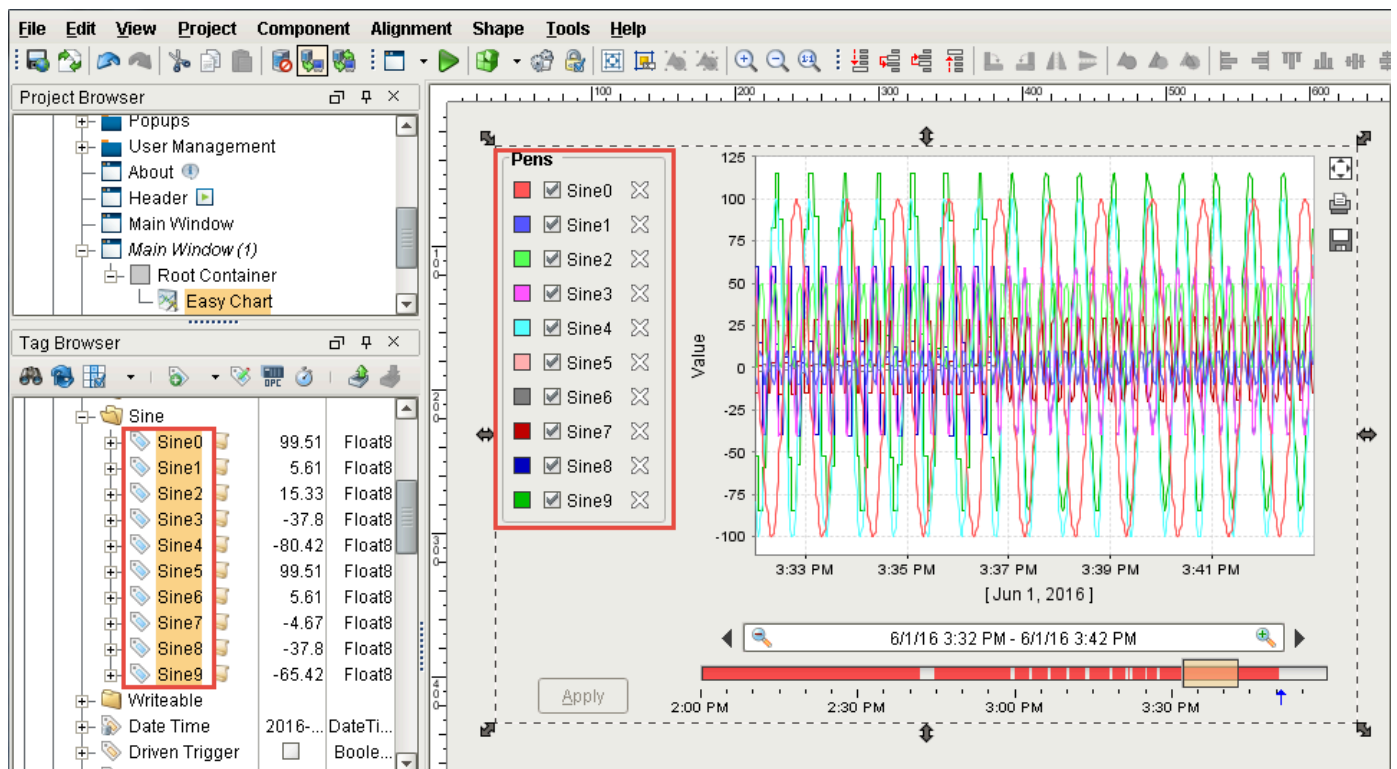
Create a Graph

1. Open your Main Window. From the Charts section of the Component Palette, drag in the Easy Chart component.



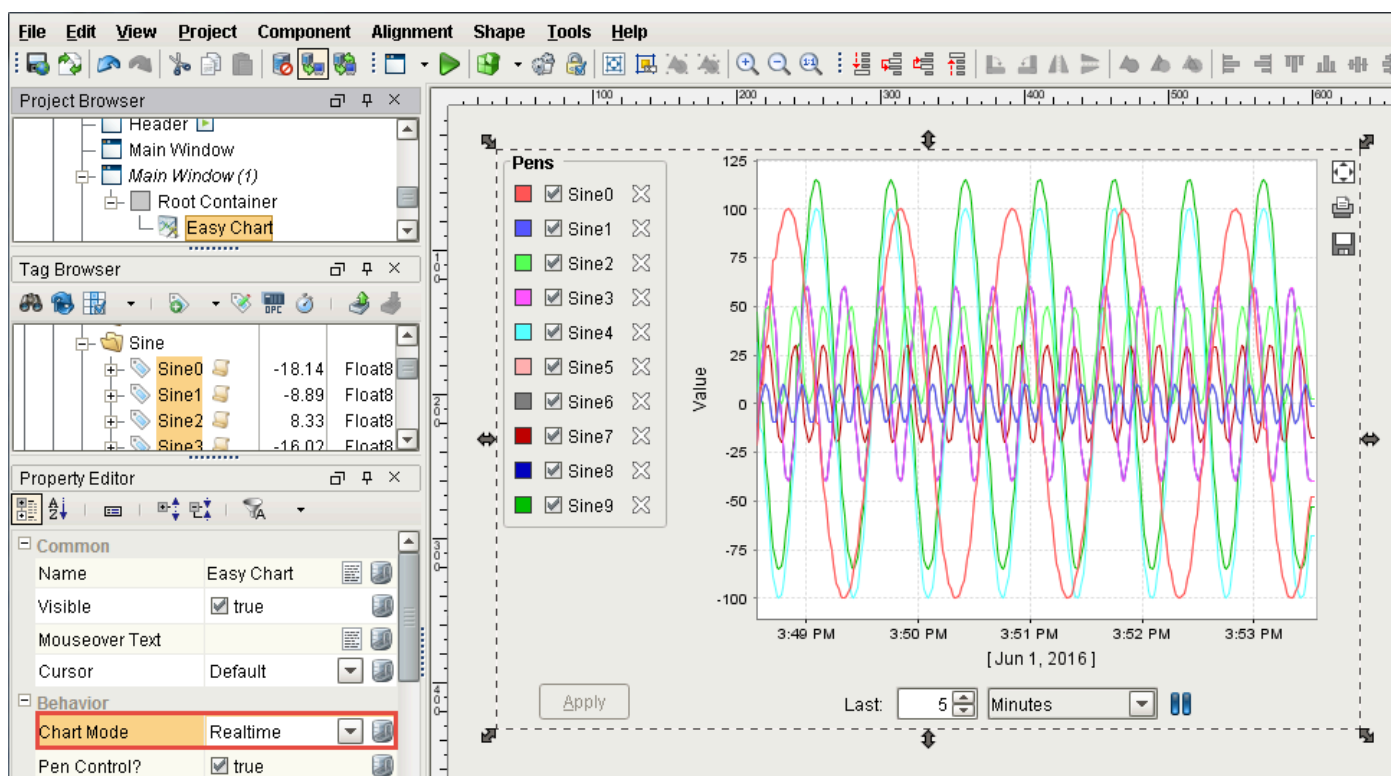
We can add the tags to the graph the same way we created the status and control windows through drag and drop.

2. Drag all the Sine tags from Tags Browser to the graph.



The tags are displayed on your Main Window but because you have just started logging data, not much data will show up.

3. To show more data, set the Chart Mode property of Easy Chart to Realtime.



That's it! Now open your client and view your windows.

[Learn more about Charts](#)