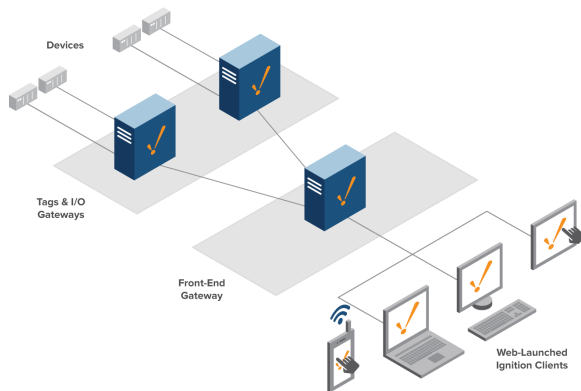


Scale Out Architecture

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.



On this page

...

- [Scaling Out Ignition for Larger Systems](#)
- [Load Balancing](#)
- [Gateway Types](#)
- [Database Placement](#)
- [Scaling Out the Databases](#)
- [Why Use the Scale Out Architecture?](#)
 - [Decentralized Servers](#)
 - [Pooled Resource Usage](#)
 - [Easy Growth](#)

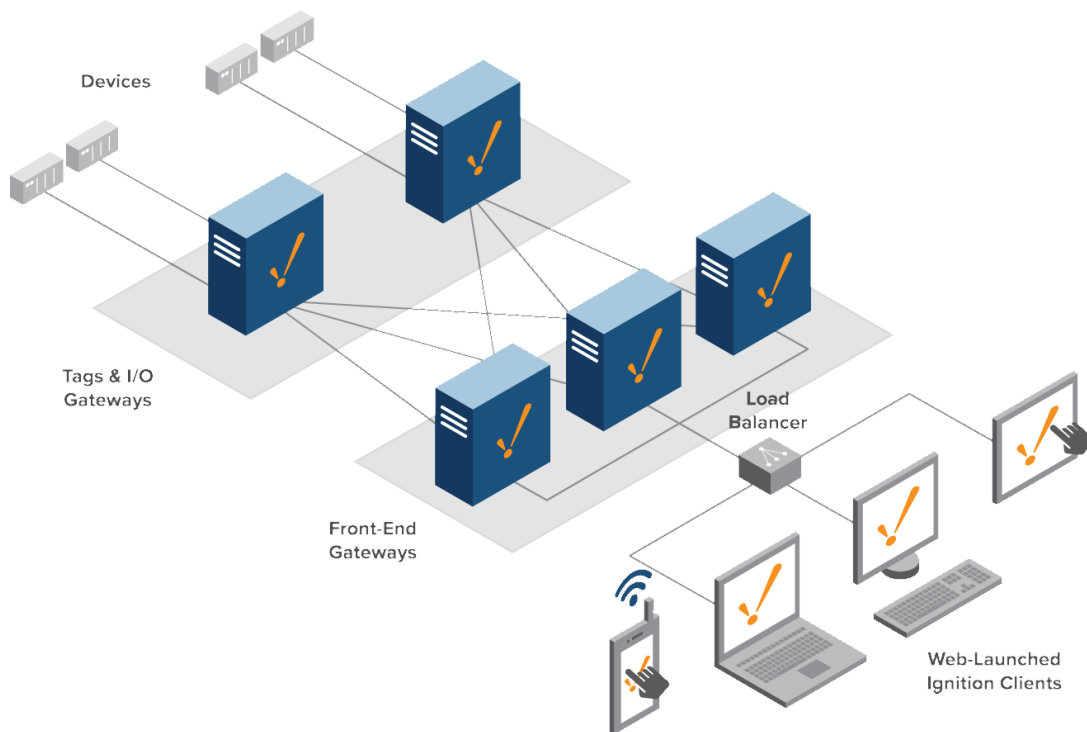


Scale Out Architecture

[Watch the Video](#)

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.



This feature is new in Ignition version **7.9.15**
[Click here](#) to check out the other new features

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 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.

Related Topics ...

- [Hub and Spoke Architecture](#)