

# IIoT: Getting Started with the WAGO PFC Series

## Hardware Selection



For MQTT/direct cloud remote IO, use the **PFC100 Series**:

<https://www.wago.com/us/pfc100>

For a Node-RED based Automation Controller/Cloud Gateway with IO points (SIM card option available), use the **PFC200 Series**: <https://www.wago.com/us/pfc200>

## Connecting to the Internet



Static IP Method:

1. Switch the PLC to "STOP" and hold the recessed "RST" button down on the front of the PLC for 8 seconds until the system light begins to flash amber. The amber light indicates a change to a temporary static IP address of 192.168.1.17.
2. Connect an ethernet cable between your computer and the PFC.
3. Set the wired ethernet adapter on your computer to an IP address in the 192.168.1.X subnet (ex. 192.168.1.2).
4. Use a web browser to navigate to 192.168.1.17
5. Log in with user: admin / password: wago
6. Navigate to the Configuration Tab and select "Networking"
7. You must now change the static IP address under "Network Details Bridge 1 (br0)" and ensure you have selected "Static IP" and not "DHCP". You can keep the same default IP of 192.168.1.17 by re-typing it here.
8. Set a new DNS server IP to 8.8.8.8
9. Under the Configuration Tab and select "Routing".
10. Under "IP Forwarding through multiple interfaces" select enabled and select submit.
11. Under "Default Static Routes" select enabled. Then set the IP address of your WAN internet gateway under "Gateway Address". This should be listed on the router provided by your internet service provider. When finished, select submit.
12. Under the Configuration Tab and select "Ports and Services".
13. In "Network Services", enable "HTTPS" and select submit. Also enable "NTP" and select submit.
14. For some TLS connections, time servers are required to authenticate the connection. It is always good practice to set up an NTP server. Under "NTP Client" in "Ports and Services", enable the time server and select a Time Server 1 (you can use 216.239.35.0) and submit.

## Simple Setup: Provisioning Tool



All of the applications and settings you need to get started with the PFC series with one shell command. Requires an Internet connection.

**Video:** <https://www.youtube.com/watch?v=kd9kTZgTI8U>

**GitHub:** <https://github.com/braunku/pfc-provisioning-tool>

**Command:** `curl -L https://raw.githubusercontent.com/braunku/pfc-provisioning-tool -o menu.sh -s && sh menu.sh`

For MQTT Remote IO (PFC100), select disable PLC Runtime and install MQTT KBUS API.

## Shell and SSH



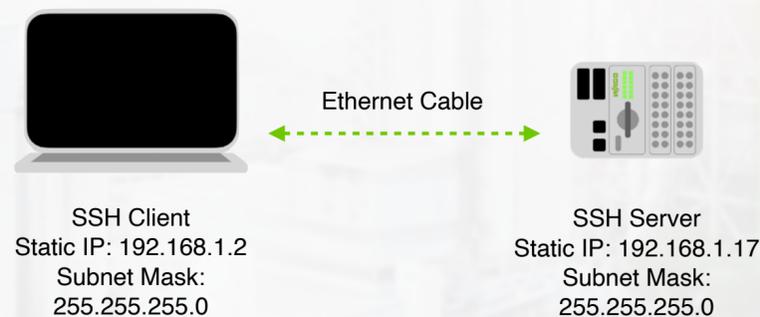
The "shell" is a program that takes commands from the keyboard and gives them to the operating system to perform. The SSH protocol (also referred to as Secure Shell) is a method for secure remote "shell" login from one computer to another.

Default credentials for WAGO:

- username: root / password: wago
- username: admin / password: wago

Networking Requirements:

- Both devices are on the same network IP range.
- Both devices are configured with the same subnet mask.
- The SSH port is open and the service is active on the server (this is enabled by default on the WAGO standard firmware).



See the full Linux Shell Quickstart guide at: <https://www.wago.community/t/how-to-getting-started-with-the-linux-shell/83>

## Docker



Docker is an open containerization platform for developing, shipping, and running applications. Popular Docker applications include:

- Node-RED
- InfluxDB
- VPN / Networking Platforms

**Image:** A Docker image is a file used to execute code in a Docker container. Docker images act as a set of instructions to build a Docker container, like a template.

**Container:** A Docker container is a virtualized runtime environment used in application development. It is used to create, run and deploy applications that are isolated from the underlying hardware.

**Volume:** Docker volumes are file systems mounted on Docker containers to preserve data generated by the running container.

See the full Docker Quickstart guide at: <https://www.wago.community/t/how-to-getting-started-with-docker/95>

## Node-RED



Node-RED is a flow-based development tool for visual programming. It allows for the connectivity of hardware devices, APIs, and online services as part of the Internet of Things.

You can access the IO points on a PFC with MQTT using the **MQTT KBUS API**: <https://github.com/jessejamescox/pfc-kbus-api> OR provided in the "Provisioning Tool"

Access the IO points inside the Node-RED editor with the "node-red-io-api" palette:

- Nodes connect directly to MQTT broker either onboard the controller or externally.
- Status and error messages from the PFC are provided in the debug.
- Integrated scaling for analog input signals and process scaling for analog outputs.
- Very low CPU resource usage.

Default MQTT Broker is 127.0.0.1 and NodeID is "PFC200" unless changed in configuration file.

