

IBM Aspera faspio Gateway 1.3



Contents

- Introduction..... 1**
- PDFs..... 2**
- Installation and configuration..... 2**
 - Installing on macOS..... 2
 - Installing on Linux..... 3
 - Installing on Windows..... 4
 - Configuring the Gateway configuration file..... 5
 - Securing the Gateway..... 8
- Launching faspio Gateway..... 9**
 - Launching on macOS..... 10
 - Launching on Windows..... 10
 - Launching on Linux..... 11
- Testing the Gateway 12**
- Appendices..... 12**
 - Configuring the logging file..... 12
 - Uninstalling on macOS..... 13
 - Uninstalling on Linux..... 13
 - Uninstalling on Windows..... 14

Introduction

IBM Aspera faspio is a lightweight software component for high-speed bidirectional data transport. Using the patented Aspera FASP protocol, faspio achieves speeds of up to 4 Gb/sec of aggregated FASP traffic over unmanaged networks.

Aspera faspio fully utilizes available bandwidth to transfer data in byte-order sequence at the maximum possible speed with near-zero latency. It removes the barriers of size, distance, and complexity to move data between on-premises and cloud infrastructures.

Aspera faspio provides significant improvements in performance and service quality when transferring data between highly remote or dispersed locations in unfavorable network conditions, such as high latency and packet loss.

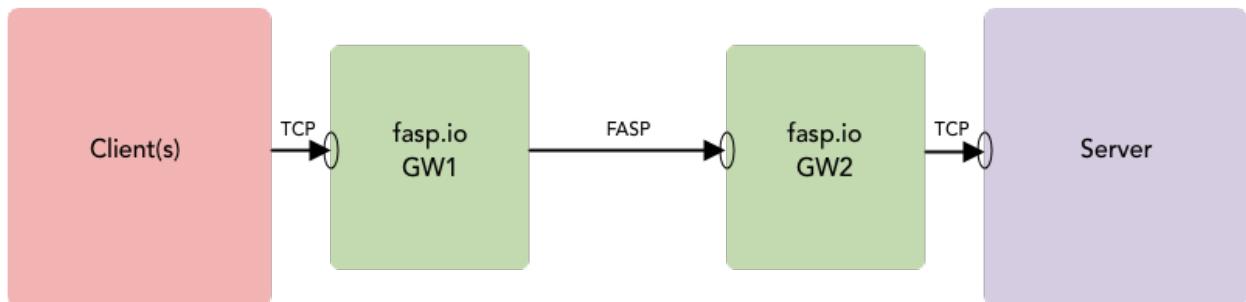
IBM Aspera Gateway is a software component that can be integrated quickly and easily with existing applications that use a TCP connection for their data flow. It improves nearly all server-to-server TCP-based data flows regardless of the distance and network conditions.

IBM Aspera faspio Gateway acts as a transport layer proxy between TCP and Aspera FASP.

Usage

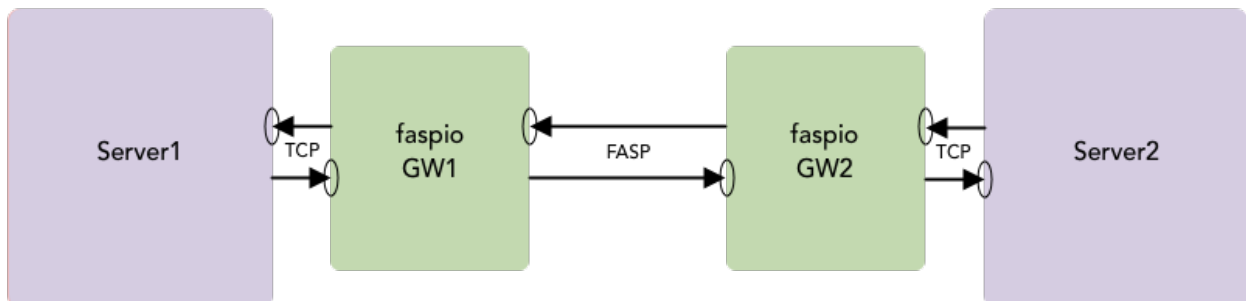
Gateway Client/Server Usage

In this configuration, two faspio Gateways are used to bridge one (or several) TCP connections from TCP clients to a TCP server over FASP:



Gateway Server/Server Usage

For some use cases, such as DB replication or messaging services (like MQ or Event Streams), communication must be established by both sides. In this mode, each server initiates a connection to the other:



For information on how to configure these two cases, see [“Configuring the Gateway configuration file”](#) on page 5.

License requirements

The faspio Gateway requires a license for using the Gateway as a FASP server. No license is required to use the Gateway as a client.

If you start the Gateway with an invalid license, you may see one of these errors.

- If you do not have a license, faspio Gateway reports:

```
The current license does not allow for FASP to run in server-mode (i.e.
bridge.local.protocal="fasp")
[2022-01-25 12:26:42.899] [ gateway ] [ error ] license-checking failure - "no license
loaded"
```

- If your license has expired, faspio Gateway reports:

```
The current license does not allow for FASP to run in server-mode (i.e.
bridge.local.protocal="fasp")
[2022-01-25 12:26:42.899] [ gateway ] [ error ] license-checking failure - "license expired"
```

PDFs

1.3

- [IBM Aspera faspio Gateway 1.3 Guide](#)
- [IBM Aspera faspio Gateway 1.3 Release Notes](#)

1.2

- [IBM Aspera faspio Gateway 1.2 Guide](#)
- [IBM Aspera faspio Gateway 1.2 Release Notes](#)

1.1

- [IBM Aspera faspio Gateway 1.1 Guide](#)
- [IBM Aspera faspio Gateway 1.1.1 Release Notes](#)
- [IBM Aspera faspio Gateway 1.1.0 Release Notes](#)

1.0

- [IBM Aspera faspio Gateway 1.0 Guide](#)
- [IBM Aspera faspio Gateway 1.0.1 Release Notes](#)

Installation and configuration

Installing on macOS

Download and install IBM Aspera faspio Gateway on your macOS system.

To install faspio Gateway on your macOS machine:

1. Ensure that you're on a supported version of macOS. Supported versions are listed in the release notes.
2. Download the macOS .pkg installer from IBM Passport Advantage: <https://www.ibm.com/products/software>

The name of the installer will be similar to:

```
ibm-faspio-gateway_1.1.0-version_macOS.pkg
```

3. Run the installer by opening the file and following the prompts.
4. If you are using faspio Gateway as a FASP server, you must provide a valid license at `/usr/local/etc/faspio/aspera-license`.

For more information about configuration and client versus server usage, see [“Configuring the Gateway configuration file”](#) on page 5.

The locations of installed files are shown below:

Executable

```
/usr/local/bin/faspio-gateway
```

IBM Aspera license

```
/usr/local/etc/faspio/aspera-license
```

Configuration files

```
/usr/local/etc/faspio/gateway.toml  
/usr/local/etc/faspio/logging.toml
```

Launchd service configuration file

```
/Library/LaunchDaemons/com.asperasoft.faspio.gateway.plist
```

IBM SWID file

```
/usr/local/share/faspio-gateway/iso-swid/ibm.com_IBM_Aspira_faspio_Gateway-1.0.0.swidtag
```

IBM software licenses

```
/usr/local/share/faspio-gateway/license/*
```

Installing on Linux

Download and install IBM Aspera faspio Gateway on your Linux system.

To install faspio Gateway on your Linux machine:

1. Ensure that you're on a supported version of Linux. Supported versions are listed in the release notes.
2. Download the Linux installer from IBM Passport Advantage: <https://www.ibm.com/products/software>
3. Choose the `.deb` or `.rpm` package as appropriate to your system. The names of the installers follow the syntax:

```
ibm-faspio-gateway_1.2.0-version_arch.deb  
ibm-faspio-gateway_1.2.0-version_arch.rpm
```

4. Install from either the `.deb` or `.rpm` package:

- `.deb`:

```
sudo apt install ./ibm-faspio-gateway_1.2.0-version_arch.deb
```

- `.rpm`:

```
sudo yum install ./ibm-faspio-gateway_1.2.0-version_arch.rpm
```

5. If you are using faspio Gateway as a FASP server, you must provide a valid license at `/usr/local/etc/faspio/aspera-license`.

For more information about configuration and client versus server usage, see [“Configuring the Gateway configuration file”](#) on page 5.

The locations of installed files are shown below:

Executable

```
/usr/local/bin/faspio-gateway
```

IBM Aspera license

```
/usr/local/etc/faspio/aspera-license
```

Configuration files

```
/usr/local/etc/faspio/gateway.toml  
/usr/local/etc/faspio/logging.toml
```

SystemD service configuration file

```
/usr/local/lib/systemd/system/faspio-gateway.service
```

IBM SWID file

```
/usr/local/share/faspio-gateway/iso-swid/ibm.com_IBM_Aspira_faspio_Gateway-1.0.0.swidtag
```

IBM software licenses

```
/usr/local/share/faspio-gateway/license/*
```

Installing on Windows

Download and install IBM Aspera faspio Gateway on your Windows system.

To install faspio Gateway on your Windows machine:

1. Ensure that you're on a supported version of Windows and are logged in with an account that has administrator privileges. Supported versions are listed in the release notes.
2. Download the Windows .msi installer from IBM Passport Advantage: <https://www.ibm.com/products/software> The name of the installer will be similar to:

```
ibm-faspio-gateway_version_win64.msi
```

3. Open the .msi file. The installer wizard launches. Follow the prompts to complete the installation.
4. If you are using faspio Gateway as a FASP server, you must provide a valid license at C:\Program Files\IBM\faspio Gateway\config\aspera-license.

For more information about configuration and client versus server usage, see [“Configuring the Gateway configuration file”](#) on page 5.

The locations of installed files are shown below. All files are found under:

```
C:\Program Files\IBM\faspio Gateway\
```

Executable

```
bin\faspio-gateway
```

IBM Aspera license

```
config\aspera-license
```

Configuration files

```
config\gateway.toml
config\logging.toml
```

IBM SWID file

```
iso-swid\ibm.com_IBM_Aspira_faspio_Gateway-1.0.0.swidtag
```

IBM software licenses

```
license\*
```

Configuring the Gateway configuration file

Configure two or more faspio Gateway servers to connect to each other.

The Gateway configuration file

The IBM Aspera faspio Gateway configuration file, `gateway.toml`, is located here:

Linux, macOS:

```
/usr/local/etc/faspio/gateway.toml
```

Windows:

```
C:\Program Files\IBM\faspio Gateway\config\gateway.toml
```

The `gateway.toml` file included in the installation is provided as a template. Modify this file to specify your ports, host names, and so on.

Every time you modify `gateway.toml` to make changes for your configuration, you must restart the IBM faspio Gateway service. For information on starting and stopping the Gateway service, see [“Launching faspio Gateway”](#) on page 9.

Note: The version of the FASP protocol included in faspio Gateway uses a single UDP port. Whatever port you configure for your FASP connection over the WAN must have the same UDP port open on any firewalls along the connection path.

License requirements

The faspio Gateway requires a license for using the Gateway as a FASP server. No license is required to use the Gateway as a client.

If you start the Gateway with an invalid license, you may see one of these errors.

- If you do not have a license, faspio Gateway reports:

```
The current license does not allow for FASP to run in server-mode (i.e.
bridge.local.protocal="fasp")
[2022-01-25 12:26:42.899] [ gateway ] [ error ] license-checking failure - "no license
loaded"
```

- If your license has expired, faspio Gateway reports:

```
The current license does not allow for FASP to run in server-mode (i.e.
bridge.local.protocal="fasp")
[2022-01-25 12:26:42.899] [ gateway ] [ error ] license-checking failure - "license expired"
```

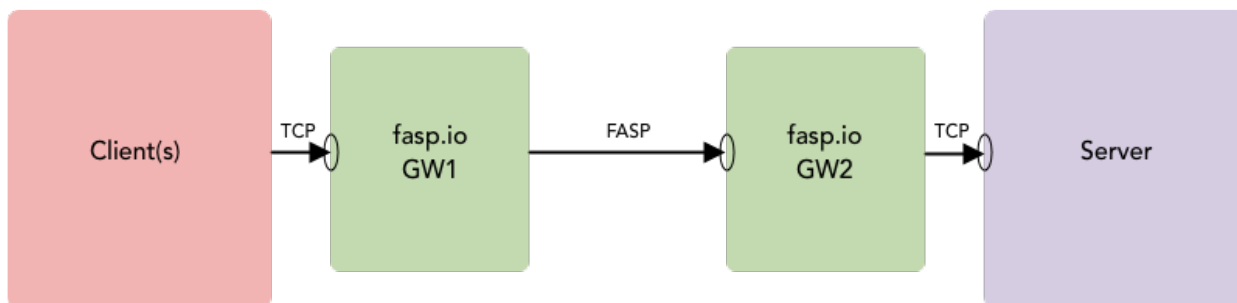
Configuration examples

Important: For simplicity, the examples in this section do not include security settings. After understanding how to configure the connection between servers, make sure to [understand how to secure](#)

your servers. Security is enabled by default and you need to properly configure security settings before you can use your Gateway servers.

Example: Client to Server

In this configuration, two Gateways are used to establish a FASP bridge between two TCP connections, a TCP client and a TCP server:



Given a server listening on port 12345, configure your client to point to Gateway 1 (GW1 IP) and port 12345:

GW1 Configuration

```
[[bridge]]
[bridge.local]
  protocol = "tcp"
  host = "0.0.0.0"
  port = 12345

[bridge.forward]
  protocol = "fasp"
  host = "GW2"
  port = 12345
```

GW2 Configuration

```
[[bridge]]
[bridge.local]
  protocol = "fasp"
  host = "0.0.0.0"
  port = 12345

[bridge.forward]
  protocol = "tcp"
  host = "Server"
  port = 12345
```

protocol

The transfer protocol type, either "tcp" or "fasp". The protocol type must be quoted.

host

The host name or IP address, always quoted.

port

Either of these:

- Port number (quoting optional)
- Known port/service name (quoting required), such as "http" or "ftp-data"

You can specify a range of numbered ports using a hyphen; for example, 100-110. Details:

- The range of port numbers must be ascending (low to high); for example, 100-110, but not 100-90.
- The size of the range cannot exceed 50 ports.
- If a port range is used, the range for the local port and the forward port must be the same size.
- The ranges do not need to consist of the same ports. For example, the local range could be 501-509 and the forward range could be 511-519; however, the forward range could not be 511-520.

Example: Forwarding to the first available host

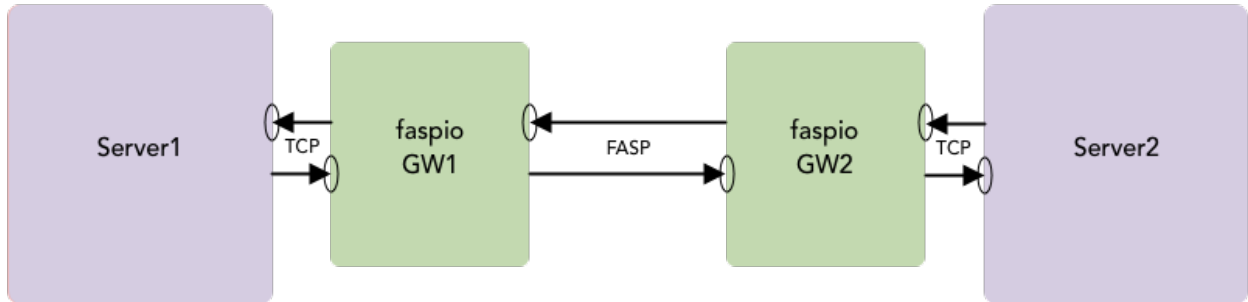
For `bridge.forward`, Gateway can loop through an array of specified host names or IP addresses and forward to the first available host it finds. For example:

```
[bridge.forward]
  protocol = "fasp"
  host = ["GW2", "10.0.0.2"]
  port = 12345
```


A hostname can resolve to multiple IP addresses. If a hostname is specified—either as a single entry or as an entry within the array—each of its IP addresses are tried until a connection is established.

Example: Server to Server

For some use cases, such as DB replication or messaging services (like MQ or Event Streams), communication must be established by both sides. In this mode, each server initiates a connection to the other:



GW1 configuration

```
[[bridge]]
  name = "Outbound"
  [bridge.local]
    protocol = "tcp"
    host = "0.0.0.0"
    port = 12345

  [bridge.forward]
    protocol = "fasp"
    host = "GW2"
    port = 12345

[[bridge]]
  name = "Inbound"
  [bridge.local]
    protocol = "fasp"
    host = "0.0.0.0"
    port = 54321

  [bridge.forward]
    protocol = "tcp"
    host = "Server1"
    port = 54321
```

GW2 configuration

```
[[bridge]]
  name = "Outbound"
  [bridge.local]
    protocol = "tcp"
    host = "0.0.0.0"
    port = 54321

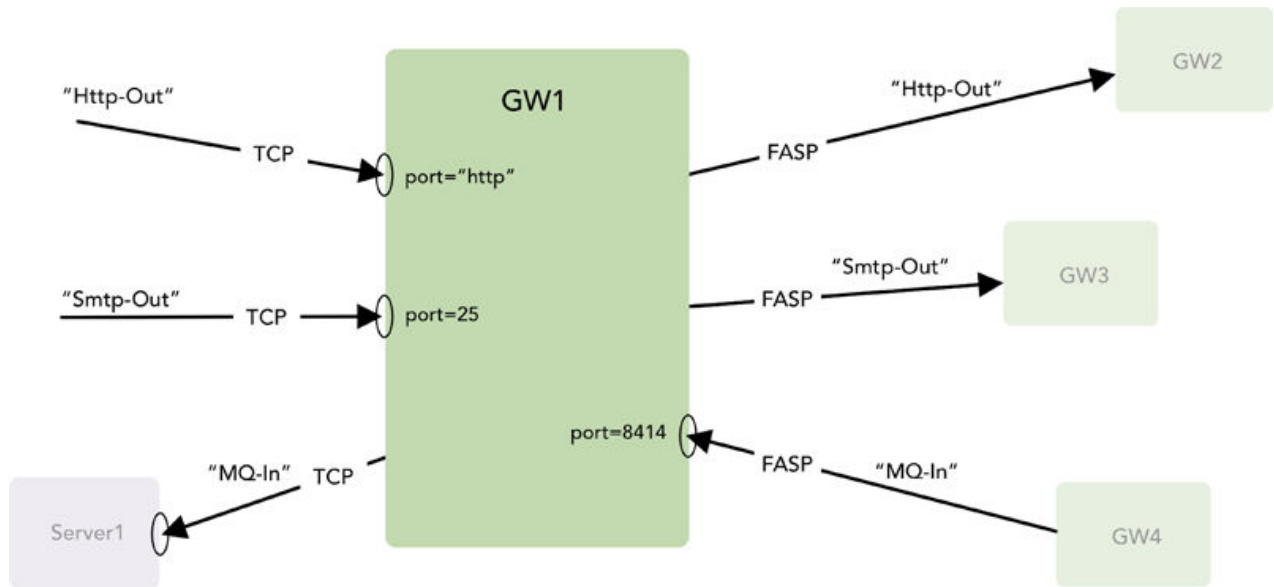
  [bridge.forward]
    protocol = "fasp"
    host = "GW1"
    port = 54321

[[bridge]]
  name = "Inbound"
  [bridge.local]
    protocol = "fasp"
    host = "0.0.0.0"
    port = 12345

  [bridge.forward]
    protocol = "tcp"
    host = "Server2"
    port = 12345
```

Gateway configuration options

Gateway can also be configured with multiple bridges, multiple ports, multiple destinations, and multiple services. The image below shows a single gateway configured with examples of these combinations.



Bridge: HTTP-Out

```
[[bridge]]
  name = "Http-Out"
  [bridge.local]
    protocol = "tcp"
    host =
"0.0.0.0"
    port = "http"
  [bridge.forward]
    protocol = "fasp"
    host = "GW2"
    port = "http"
```

Bridge: Smt-Out

```
[[bridge]]
  name = "Smt-Out"
  [bridge.local]
    protocol =
"tcp"
    host =
"0.0.0.0"
    port = 25
  [bridge.forward]
    protocol =
"fasp"
    host =
"GW3"
    port = 25
```

Bridge: MQ-In

```
[[bridge]]
  name = "MQ-In"
  [bridge.local]
    protocol =
"fasp"
    host =
"0.0.0.0"
    port = 8414
  [bridge.forward]
    protocol =
"tcp"
    host =
"Server1"
    port = 8414
```

Securing the Gateway

The faspio Gateway uses Transport Layer Security (TLS) to secure your TCP connections and initiate key exchange for the FASP protocol.

Important: TLS is enabled by default on all your bridges. You must provide valid certificates to your Gateways before they can connect.

Configuring TLS

When two Gateways connect to each other, they use Mutual TLS (mTLS) authentication to ensure traffic is secure and trusted in both directions. mTLS requires a certificate chain, a certificate key, and a verification key on both servers. Here is a sample configuration of two servers configured with mTLS:

GW1 Configuration (Client)

```
[[bridge]]
  [bridge.local]
    protocol = "tcp"
    host = "127.0.0.1"
    port = "2000-2001"
    tls_enabled = true
    cert_chain = "tls/
gw1_cert_chain.pem"
    cert_key = "tls/gw1_cert_key.pem"
    verify = "tls/verify.pem"
```

GW2 Configuration (Server)

```
[[bridge]]
  [bridge.local]
    protocol = "fasp"
    host = "127.0.0.1"
    port = "3000-3001"
    tls_enabled = true
    cert_chain = "tls/
gw2_cert_chain.pem"
    cert_key = "tls/gw2_cert_key.pem"
    verify = "tls/verify.pem"
```

GW1 Configuration (Client)

```
[bridge.forward]
  protocol = "fasp"
  host = "Gateway2"
  port = "3000-3001"
  tls_enabled = true
  cert_chain = "tls/
gw1_cert_chain.pem"
  cert_key = "tls/gw1_cert_key.pem"
  verify = "tls/verify.pem"
host_verify_enabled = true
```

GW2 Configuration (Server)

```
[bridge.forward]
  protocol = "tcp"
  host = "127.0.0.1"
  port = "4000-4001"
  tls_enabled = true
  cert_chain = "tls/
gw2_cert_chain.pem"
  cert_key = "tls/gw2_cert_key.pem"
  verify = "tls/verify.pem"
```

cert_chain

The relative path to the certificate chain signed by a valid CA.

cert_key

The relative path to the private key matching the cert chain.

verify

The relative path to the verification key that verifies the other server's chain is signed by the correct CA.

host_verify_enabled

Optionally enforce additional security by requiring that the host field defined in the client's `bridge.forward` section matches the Common Name in the server's SSL certificate.

Disabling TLS

To disable TLS, add `tls_enabled = false` to the section for which connection you want to disable TLS. For example, if your faspio Gateway servers and HSTS servers run in the same private and secure network, you may consider disabling TLS for those sections only:

GW1 Configuration (Client)

```
[[bridge]]
  [bridge.local]
    protocol = "tcp"
    host = "127.0.0.1"
    port = "2000-2001"
    tls_enabled = false

  [bridge.forward]
    protocol = "fasp"
    host = "Gateway2"
    port = "3000-3001"
    tls_enabled = true
    cert_chain = "tls/
gw1_cert_chain.pem"
    cert_key = "tls/gw1_cert_key.pem"
    verify = "tls/verify.pem"
  host_verify_enabled = true
```

GW2 Configuration (Server)

```
[[bridge]]
  [bridge.local]
    protocol = "fasp"
    host = "127.0.0.1"
    port = "3000-3001"
    tls_enabled = true
    cert_chain = "tls/
gw2_cert_chain.pem"
    cert_key = "tls/gw2_cert_key.pem"
    verify = "tls/verify.pem"

  [bridge.forward]
    protocol = "tcp"
    host = "127.0.0.1"
    port = "4000-4001"
    tls_enabled = false
```

Launching faspio Gateway

To launch IBM Aspera faspio Gateway, you start or stop the **faspio-gateway** service using one of these methods:

- Linux: Uses the **systemd** service manager. Start the service from the command line.
- Windows: Start the service from the command line or from the Windows Services panel.
- macOS: Uses the **launchd** service manager. Start the service from the command line.

Note: Each time you change your Gateway configuration—by modifying `gateway.toml`—you must restart the service. For information about configuring your system, see [“Configuring the Gateway configuration file”](#) on page 5.

Launching on macOS

The OS service manager on macOS is **launchd**, and the terminal command for controlling it is **launchctl**. You can load, unload, start, stop, or check the status of the **faspio-gateway** service by running the **launchctl** sub-commands.

launchctl sub-commands

- Load the service. Loading the service also starts it:

```
$ sudo launchctl load /Library/LaunchDaemons/com.asperasoft.faspio.gateway.plist
```

- Start the loaded service (such as a stopped service that is still loaded):

```
$ sudo launchctl start com.asperasoft.faspio.gateway
```

- Check the status of the service:

```
$ sudo launchctl list | grep com.asperasoft.faspio.gateway
```

- Stop the service:

```
$ sudo launchctl stop com.asperasoft.faspio.gateway
```

- Unload the service:

```
$ sudo launchctl unload /Library/LaunchDaemons/com.asperasoft.faspio.gateway.plist
```

Launching on Windows

To launch the faspio Gateway service on Windows, you start the service from either the command line or the Windows Services panel. Both methods require that you have admin privileges.

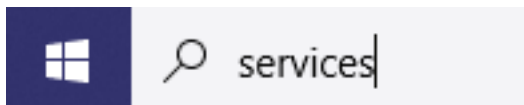
To start the service from the command line, run:

```
net start faspio-gateway
```

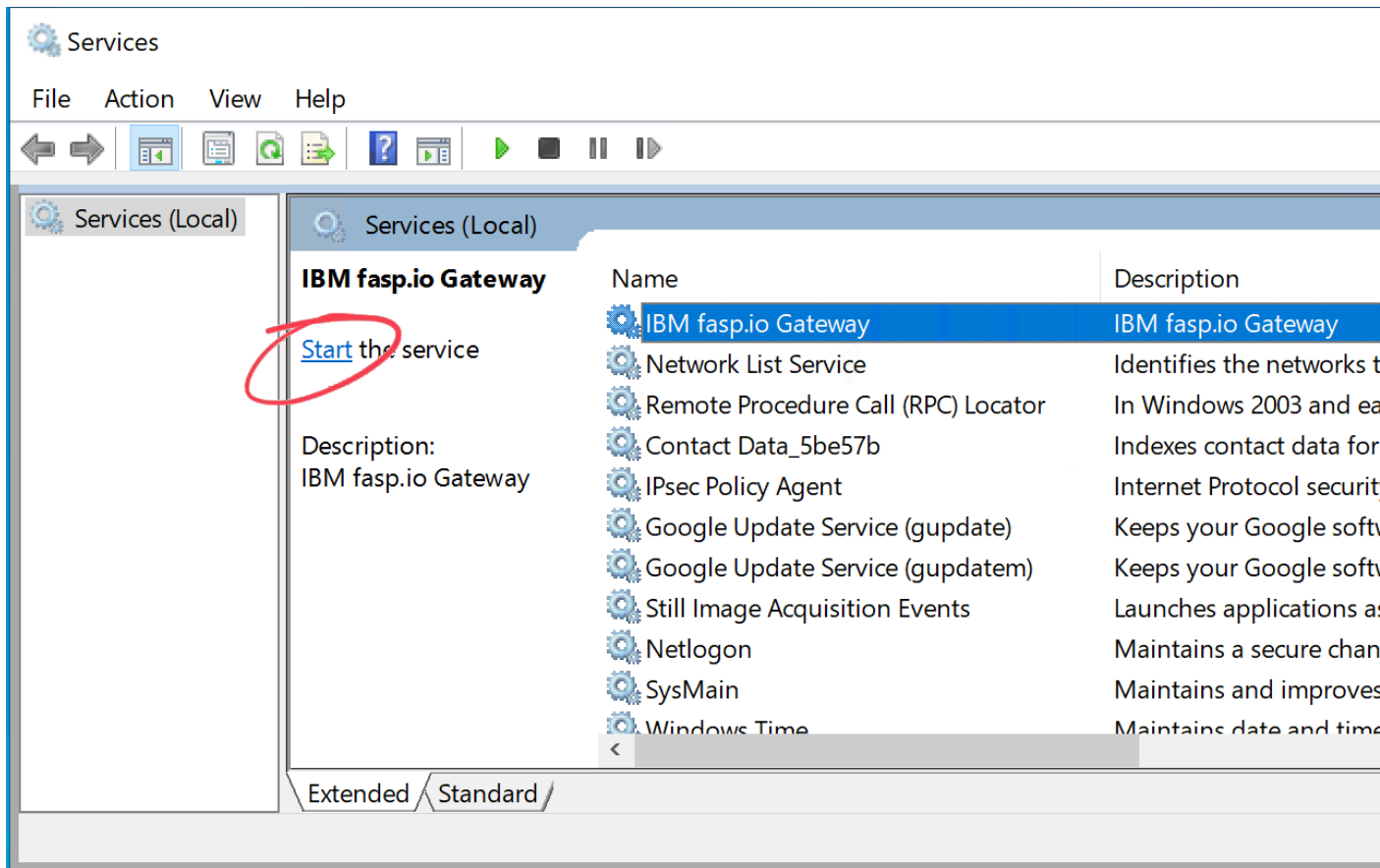
To stop the service:

```
net stop faspio-gateway
```

To start the service from the Windows UI, open the Services panel. To access it quickly, enter "services" in the search box next to the **Start** button:



In the display that appears, click **Services**. The Windows **Services** panel opens. In the list of services, find and select **IBM faspio Gateway**. To launch the service, click **Start the service**.



Launching on Linux

The OS service manager used on Linux is **systemd**, and the terminal command for controlling it is **systemctl**. You can load, unload, start, stop, or check the status of the **faspio-gateway** service by running **systemctl** sub-commands. You can also use **journalctl** to check service logs.

systemctl sub-commands

- Start the service:

```
sudo systemctl start faspio-gateway
```

- Stop the service:

```
sudo systemctl stop faspio-gateway
```

- Enable the service (to restart after reboot):

```
sudo systemctl enable faspio-gateway
```

- Check service status:

```
sudo systemctl status faspio-gateway
```

- See service logs:

```
sudo journalctl --unit=faspio-gateway
```

Service not found error message

If you see the following error message when running one of these commands, `systemd` may not be monitoring the folder for the `faspio Gateway` service:

```
Unit faspio-gateway.service not found.
```

If you get the error, the following command usually resolves the problem:

```
sudo systemctl daemon-reload
```

Testing the Gateway

The tests below use the sample configuration from [“Configuring the Gateway configuration file”](#) on page 5.

Simple echo test through netcat

On both machines, run:

```
faspio-gateway
```

On the server machine (GW2), listen:

```
nc -v -l 12345
```

On the client machine (GW1), connect to the server (GW2):

```
nc GW2_ip_addr 12345
```

netcat should report a connection from your client machine.

Stream data through netcat

On both machines, run:

```
faspio-gateway
```

On the server machine (GW2), listen:

```
nc -v -l 12345 > /dev/null
```

On the client machine (GW1), connect and stream 1 GB of data to the server (GW2):

```
dd if=/dev/zero count=1024 bs=1M | nc -v GW2_ip_addr 12345 >/dev/null
```

netcat should report a successful transfer.

Appendices

Configuring the logging file

Logging Configuration File

Logging configuration is defined in `logging.toml`, which is located here:

Linux, macOS:

```
/usr/local/etc/faspio/logging.toml
```

Windows:

```
C:\Program Files\IBM\faspio Gateway\config\logging.toml
```

Error, warning, and info logs are sent to the console by default. In the default `logging.toml` file, four loggers are made available:

- gateway – High-level logger for the gateway.
- s2s – Stream-to-stream session class logger.
- faspio-cpp – Logger for the Asio/C++ FASP SDK.
- faspio-c – Logger for the FASP protocol.

Note: Every time you modify `logging.toml`, you must restart the faspio Gateway service.

For more information on how to configure logging, see the full reference at:

https://github.com/guangie88/spdlog_setup

Notes:

- The `level` setting is optional for both sinks and loggers.
- The level for error logging is `err`, not `error`.
- The `_st` suffix means single-threaded.
- The `_mt` suffix means multi-threaded.
- `syslog_sink` is thread-safe by default. No `_mt` suffix is required.

The `spdlog` default logging format is:

```
[2014-10-31 23:46:59.678] [loggername] [info] message
```

For information on how to customize `spdlog` formatting, see:

<https://github.com/gabime/spdlog/wiki/3.-Custom-formatting>

Uninstalling on macOS

To uninstall faspio Gateway:

1. Unload the service:

```
sudo launchctl unload /Library/LaunchDaemons/com.asperasoft.faspio.gateway.plist
```

2. Delete all the faspio installed files listed in the section above (executable, configuration files, `launchd` service config file, SWID file, and license files).

Note: Previous versions of faspio Gateway (1.1 and before) are called `ibm-fasp.io-gateway`.

Uninstalling on Linux

To uninstall faspio Gateway, run:

- `.deb`:

```
sudo apt uninstall ./ibm-faspio-gateway
```

- `.rpm`:

```
sudo yum uninstall ./ibm-faspio-gateway
```

Note: Previous versions of faspio Gateway (1.1 and before) are called `ibm-fasp.io-gateway`.

Uninstalling on Windows

To uninstall faspio Gateway:

1. From the **Start** menu, go to **Settings > System > Apps & Features**.
2. Select the **IBM faspio Gateway** app. The **Uninstall** button appears.
3. Click **Uninstall** and confirm.

Note: Previous versions of faspio Gateway (1.1 and before) are called `ibm-fasp.io-gateway`.

