Quick install

Use this guide to get a quick, hands-on look at what PingGateway software can do. You will download, install, and use PingGateway on your local computer. Find more installation options in Install.

This guide assumes familiarity with the following topics:

- HTTP, including how clients and servers exchange messages, and the role that a reverse proxy (gateway) plays
- JSON, the format for PingGateway configuration files
- Managing services on operating systems and application servers
- Configuring network connections on operating systems

Product names changed when ForgeRock became part of Ping Identity. PingGateway was formerly known as ForgeRock Identity Gateway, for example. Learn more about the name changes from New names for ForgeRock products.

Download PingGateway

The .zip file unpacks into a `/path/to/identity-gateway-2024.11.0` directory with the following content:

- `bin` : Start and stop executables
- `classes` : Initially empty; used to install patches from ForgeRock support
- `docker/Dockerfile` : Dockerfile and README to build a PingGateway Docker image
- `legal-notices` : Licenses and copyrights
- `lib` : PingGateway and third-party libraries

    1. Create a local installation directory for PingGateway. The examples in this section use `/path/to` .

        IMPORTANT

        The installation directory should be a new, empty directory. Installing PingGateway into an existing installation directory can cause errors.

2. Download `PingGateway-2024.11.0.zip` from the BackStage download site, and copy the .zip file to the installation directory:

   ```
   $ cp PingGateway-2024.11.0.zip /path/to/PingGateway-2024.11.0.zip
   ```

3. Unzip the file:

   ```
   $ unzip PingGateway-2024.11.0.zip
   ```

   The directory `/path/to/identity-gateway-2024.11.0` is created.

Prepare the network

Configure the network to include hosts for PingGateway, AM, and the sample application. Learn more about host files from the Wikipedia entry, Hosts (file).

1. Add the following entry to your host file:

   Linux | Windows

   ```
   /etc/hosts
   ```

   ```
   127.0.0.1  localhost ig.example.com app.example.com am.example.com
   ```

Start and stop PingGateway

Start PingGateway with default settings

Use the following step to start the instance of PingGateway, specifying the configuration directory where PingGateway looks for configuration files.

1. Start PingGateway:

   Linux | Windows

   ```
   $ /path/to/identity-gateway-2024.11.0/bin/start.sh
   ```

```
...
... started in 1234ms on ports : [8080]
```

By default, PingGateway configuration files are located under `$HOME/.openig` (on Windows `%appdata%\OpenIG`). For information about how to use a different location, refer to Configuration location.

2. Check that PingGateway is running in one of the following ways:
   - Ping PingGateway at `http://ig.example.com:8080/openig/ping` and make sure an `HTTP 200` is returned.
   - Display the product version and build information at `http://ig.example.com:8080/openig/api/info`.

Stop PingGateway

Use the `stop.sh` script to stop an instance of PingGateway, specifying the instance directory as an argument. If the instance directory isn't specified, PingGateway uses the default instance directory:

Linux Windows

```
$ /path/to/identity-gateway-2024.11.0/bin/stop.sh
$HOME/.openig
```

Use the sample application

ForgeRock provides a mockup web application for testing PingGateway configurations. The sample application is used in the examples in this guide and the *Gateway guide*.

Download the sample application

1. Download `PingGateway-sample-application-2024.11.0-jar-with-dependencies.jar`, from the BackStage download site.

Start the sample application

1. Start the sample application:

```
$ java -jar PingGateway-sample-application-2024.11.0-jar-with-
dependencies.jar
```

```
...
[...] [INFO   ] Press Ctrl+C to stop the server.
```

By default this server listens for HTTP on port 8081 and for HTTPS on port 8444. If one or both of those ports aren't free, specify other ports:

```
$ java -jar PingGateway-sample-application-2024.11.0-jar-with-dependencies.jar 8888 8889
```

By default, OpenTelemetry support is disabled in the sample application. To enable it, set `OTEL_SDK_DISABLED=false` when starting the sample application:

```
$ OTEL_SDK_DISABLED=false java -jar PingGateway-sample-application-2024.11.0-jar-with-dependencies.jar
```

The tracing configuration for the sample application uses automatic configuration. For example, to enable OpenTelemetry support and publish to a remote service:

```
$ OTEL_SDK_DISABLED=false \
OTEL_EXPORTER_OTLP_TRACES_ENDPOINT=http://exporter.example.com:1234/traces \
java -jar PingGateway-sample-application-2024.11.0-jar-with-dependencies.jar
```

Learn more about configuration options in the OpenTelemetry documentation on automatic configuration. The sample application supports only `http/protobuf` for the exporter protocol.

2. Check that the sample application is running in one of the following ways:
   a. Go to http://app.example.com:8081/home to access the home page of the sample application. Information about the browser request is displayed.
   b. Go to http://app.example.com:8081/login to access the login page of the sample application, and then log in with username `demo` and password `Ch4ng31t`. The username and some information about the browser request is displayed.

Stop the sample application

1. In the terminal where the sample application is running, select CTRL+C to stop the sample application.

Serve static resources

When the sample application is used with PingGateway in examples, it must serve static resources, such as the .css. Similarly, browser requests often serve resources that don't need protection, such as icons and .gif files.

Add the following route to PingGateway to serve the sample application .css and other static resources:

Linux Windows

```
$HOME/.openig/config/routes/00-static-resources.json
```

```
{
  "name" : "00-static-resources",
  "baseURI" : "http://app.example.com:8081",
  "condition": "${find(request.uri.path,'^/css') or
matchesWithRegex(request.uri.path, '^/.*\\\\.ico$') or
matchesWithRegex(request.uri.path, '^/.*\\\\.gif$')}",
  "handler": "ReverseProxyHandler"
}
```

Configuration options

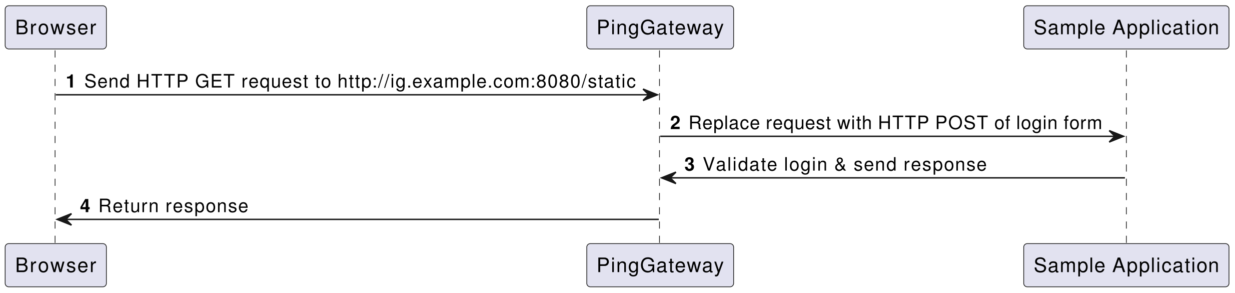
To view the command-line options for the sample application, start it with the `-h` option:

```
$ java -jar PingGateway-sample-application-2024.11.0-jar-with-
dependencies.jar -h
Usage: <main class> [options]
  Options:
    --http
      The HTTP port number.
      Default: 8081
    --https
      The HTTPS port number.
      Default: 8444
    --session
      The session timeout in seconds.
      Default: 60
    --am-discovery-url
      The AM URL base for OpenID Provider Configuration.
      Default: http://openam.example.com:8088/openam/oauth2
    --latency
      The simulated request latency in seconds.
      Default: 2
    -h, --help
      Default: false
```

Protect an application with PingGateway

This section gives a simple example of how to use PingGateway to protect an application. For many more examples of how to protect applications with PingGateway, refer to the Gateway guide.

In the following example, a browser requests access to the sample application, and PingGateway intercepts the request to log the user into the application. The following image shows the flow of data in the example:


Browser
PingGateway
Sample Application
1 Send HTTP GET request to http://ig.example.com:8080/static
2 Replace request with HTTP POST of login form
3 Validate login & send response
4 Return response
Browser
PingGateway
Sample Application

1. The browser sends an HTTP GET request to the HTTP server on `ig.example.com`.
2. PingGateway replaces the HTTP GET request with an HTTP POST login request containing credentials to authenticate.
3. The sample application validates the credentials, and returns the page for the user `demo`.

   If PingGateway did not provide the credentials, or if the sample application couldn't validate the credentials, the sample application returns the login page.
4. PingGateway returns this response to the browser.

Configure PingGateway to log you in to an application

1. Set up PingGateway and the sample application as described in this guide.
2. Add the following route to PingGateway to serve the sample application .css and other static resources:

   Linux Windows

   ```
   $HOME/.openig/config/routes/00-static-resources.json
   ```

```
{
  "name" : "00-static-resources",
  "baseURI" : "http://app.example.com:8081",
  "condition": "${find(request.uri.path,'^/css') or
matchesWithRegex(request.uri.path, '^/.*\\\\.ico$') or
matchesWithRegex(request.uri.path, '^/.*\\\\.gif$')}",
  "handler": "ReverseProxyHandler"
}
```

3. Add the following route to PingGateway:

Linux Windows

```
$HOME/.openig/config/routes/01-static.json
```

```
{
  "handler": {
    "type": "Chain",
    "config": {
      "filters": [
        {
          "type": "StaticRequestFilter",
          "config": {
            "method": "POST",
            "uri": "http://app.example.com:8081/login",
            "form": {
              "username": [
                "demo"
              ],
              "password": [
                "Ch4ng31t"
              ]
            }
          }
        }
      ],
      "handler": "ReverseProxyHandler"
    }
  },
  "condition": "${find(request.uri.path, '^/static')}"
}
```

Notice the following features of the route:

- The route matches requests to `/static`.
- The StaticRequestFilter replaces the request with an HTTP POST, specifying the resource to post the request to, and a form to include in the request. The form includes credentials for the username `demo`.
- The ReverseProxyHandler replays the request to the sample application.

4. Check that the route system log includes a message that the new files are loaded into the config:

```
INFO  o.f.o.handler.router.RouterHandler - Loaded the route
with id '00-static-resources' registered with the name '00-
static-resources'
INFO  o.f.o.handler.router.RouterHandler - Loaded the route
with id '01-static' registered with the name '01-static'
```

5. Go to http://ig.example.com:8080/static.

   You are directed to the sample application, and logged in automatically with the username `demo`.

Next steps

This section describes some basic options to help you with PingGateway. For information about other installation options, refer to the Installation guide.

Add a base configuration file

The entry point for requests coming in to PingGateway is a JSON-encoded configuration file, expected by default at:

Linux Windows

```
$HOME/.openig/config/config.json
```

The base configuration file initializes a heap of objects and provides the main handler to receive incoming requests. Configuration in the file is inherited by all applicable objects in the configuration.

At startup, if PingGateway doesn't find a base configuration file, it provides a default version given in Default configuration. The default looks for routes in:

Linux Windows

```
$HOME/.openig/config/routes
```

Consider adding a custom `config.json` for these reasons:

- To prevent using the default `config.json`, whose configuration might not be appropriate in your deployment.
- To define an object once in `config.json`, and then use it multiple times in your configuration.

After adding or editing `config.json`, stop and restart PingGateway to bring the changes into effect.

Learn more in GatewayHttpApplication (`config.json`), Heap objects, and Router.

Add a base configuration for PingGateway

1. Add the following file to PingGateway:

Linux Windows

```
$HOME/.openig/config/config.json
```

```
{
  "handler": {
    "type": "Router",
    "name": "_router",
    "baseURI": "http://app.example.com:8081",
    "capture": "all"
  },
  "heap": [
    {
      "name": "capture",
      "type": "CaptureDecorator",
      "config": {
        "captureEntity": true,
        "_captureContext": true
      }
    }
  ],
  "session": {
    "type": "JwtSessionManager"
    }
}
```

Notice the following features of the file:

- The handler contains a main router named `_router`. When PingGateway receives an incoming request, `_router` routes the request to the first route in the configuration whose condition is satisfied.
- The `baseURI` changes the request URI to point the request to the sample application.
- The `capture` captures the body of the HTTP request and response.
- The configuration uses stateless sessions. Learn more in Sessions.

2. Stop and restart PingGateway.
3. Check that the route system log includes a message that the file is loaded into the config:

```
INFO  o.f.openig.web.Initializer - Reading the configuration
from ...config.json
```

Add a default route

When there are multiple routes in the PingGateway configuration, they are ordered lexicographically, by route name. For example, `01-static.json` is ordered before `zz-default.json`.

When PingGateway processes a request, the request traverses the routes in the configuration. If the request matches the condition for `01-static.json` it is processed by that route. Otherwise, it passes to the next route in the configuration. If a route has no condition, it can process any request.

A default route is the last route in a configuration to which a request is routed. If a request matches no other route in the configuration, it is processed by the default route.

Add a default route to prevent errors described in No handler to dispatch to.

1. Add the following route to PingGateway:

Linux | Windows

```
$HOME/.openig/config/routes/zz-default.json
```

```
{
  "handler": "ReverseProxyHandler"
}
```

Notice the following features of the route:

- The route name starts with `zz`, so it is the last route that is loaded into the configuration.
- There is no `condition` property, so the route processes all requests.
- The route calls a ReverseProxyHandler with the default configuration, which proxies the request to the application and returns the response, without changing either the request or the response.

2. Check that the route system log includes a message that the file is loaded into the config:

```
INFO  o.f.o.handler.router.RouterHandler - Loaded the route
with id
'zz-default' registered with the name 'zz-default'
```

Switch from production mode to development mode

To prevent unwanted changes to the configuration, PingGateway is by default in production mode after installation. For a description of the modes and information about switching between modes, refer to Operating modes.

Use PingGateway Studio

PingGateway Studio is a user interface to help you build and deploy your PingGateway configuration. For more information, refer to the Studio guide.

Was this helpful? 👍 👎