Upgrade

This guide shows you how to upgrade PingGateway software.

Read the Release notes before you upgrade.

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.

Plan the upgrade

Do these planning tasks **before** you start an upgrade:

Planning task	Description
Find the upgrade path	Refer to Upgrade to see if you need a drop-in upgrade or a major upgrade.
Find out what changed	Read the release notes for all releases between the current version and the new version. Understand the new features and changes in the new version compared to the current version.
Check the requirements	Make sure you meet all the requirements in the release notes for the new version. In particular, make sure you have a recent, supported Java version.
Plan for server downtime	At least one of your PingGateway servers will be down during upgrade. Plan to route client applications to another server until the upgrade process is complete and you have validated the result. Make sure the owners of client application are aware of the change, and let them know what to expect. If you have a single PingGateway server, make sure the downtime happens in a low-usage window, and make sure you let client application owners plan accordingly.

Planning task	Description
Back up	The PingGateway configuration is a set of files, including `admin.json`, `config.json`, `logback.xml`, routes, and scripts. Back up the PingGateway configuration and store it in version control, so that you can roll back if something goes wrong. Back up any tools scripts you have edited for your deployment and any trust stores used to connect securely.
Plan for rollback	Sometimes even a well-planned upgrade fails to go smoothly. In such cases, you need a plan to roll back smoothly to the pre-upgrade version. For PingGateway servers, roll back by restoring a backed-up configuration.
Prepare a test environment	Before applying the upgrade in your production environment, always try to upgrade PingGateway in a test environment. This will help you gauge the amount of work required, without affecting your production environment, and will help smooth out unforeseen problems. The test environment should resemble your production environment as closely as possible.

Upgrade

Learn about upgrade between supported versions of PingGateway in Product Support Lifecycle Policy | PingGateway and Agents.

Learn about upgrade of routes in Studio in Upgrade from an earlier version of Studio.

This section describes how to upgrade a single PingGateway instance. The most straightforward option when upgrading sites with multiple PingGateway instances is to upgrade in place. One by one, stop, upgrade, and then restart each server individually, leaving the service running during the upgrade.

PingGateway supports the following types of upgrade:

Drop-in software update

> Usually, an update from a version of PingGateway to a newer minor version, as defined in Product Support Lifecycle Policy | PingGateway and Agents. For example, the update from 2023.2 to 2023.4.

Drop-in software updates can introduce additional functionality and fix bugs or security issues. Consider the following restrictions for drop-in software updates:

- Don’t require any update to the configuration
- Cannot cause feature regression
- Can change default or previously configured behavior **only** for bug fixes and security issues
- Can deprecate **but not remove** existing functionality

Major upgrade

Usually, an upgrade from a version of PingGateway to a newer major version, as defined in Product Support Lifecycle Policy | PingGateway and Agents. For example, the upgrade from 7.2 to 2023.2.

Major upgrades can introduce additional functionality and fix bugs or security issues. Major upgrades don’t have the restrictions of drop-in software update. Consider the following features of major upgrades:

- Can require code or configuration changes
- Can cause feature regression
- Can change default or previously configured behavior
- Can deprecate **and** remove existing functionality

Drop-in software update with binaries

1. Read and act on Plan the upgrade.
2. Back up the PingGateway configuration and store it in version control so that you can roll back if something goes wrong.
3. Download PingGateway
4. Stop PingGateway.
5. Make the new configuration available on the file system.

   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.
6. Restart PingGateway from the new installation directory.
7. In a test environment that simulates your production environment, validate that the upgraded service performs as expected with the new configuration. Check the logs for new or unexpected notifications or errors.
8. Allow client application traffic to flow to the upgraded site.

Drop-in software update with Docker files

1. Read and act on Plan the upgrade.
2. Back up the PingGateway configuration and store it in version control so that you can roll back if something goes wrong.
3. Stop the Docker image.
4. Build the new base image for PingGateway.
5. Run the Docker image.
6. In a test environment that simulates your production environment, validate that the upgraded service performs as expected with the new configuration. Check the logs for new or unexpected notifications or errors.
7. Allow client application traffic to flow to the upgraded site.

Major upgrade with binaries

1. Read and act on Plan the upgrade.
2. Use the release notes for **all** releases between the version you currently use and the new version, and create a new configuration as follows:
   - Review all incompatible changes and removed functionality, and adjust your configuration as necessary.
   - Switch to the replacement settings for deprecated functionality. Although deprecated objects continue to work, they add to the notifications in the logs and are eventually removed.
   - Check the lists of fixes, limitations, and known issues to find out if they impact your deployment.
   - Recompile your Java extensions. The method signature or imports for supported and evolving APIs can change in each version.
   - Read the documentation updates for new examples and information that can help with your configuration.
3. Back up the PingGateway configuration and store it in version control so that you can roll back if something goes wrong.
4. Download PingGateway
5. Stop PingGateway.
6. Make the new configuration available on the file system.

   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.
7. Restart PingGateway from the new installation directory.

8. In a test environment that simulates your production environment, validate that the upgraded service performs as expected with the new configuration. Check the logs for new or unexpected notifications or errors.
9. Allow client application traffic to flow to the upgraded site.

Major upgrade with Docker files

1. Read and act on Plan the upgrade.
2. Use the release notes for **all** releases between the version you currently use and the new version, and create a new configuration as follows:
   - Review all incompatible changes and removed functionality, and adjust your configuration as necessary.
   - Switch to the replacement settings for deprecated functionality. Although deprecated objects continue to work, they add to the notifications in the logs and are eventually removed.
   - Check the lists of fixes, limitations, and known issues to find out if they impact your deployment.
   - Recompile your Java extensions. The method signature or imports for supported and evolving APIs can change in each version.
   - Read the documentation updates for new examples and information that can help with your configuration.
3. Back up the PingGateway configuration and store it in version control so that you can roll back if something goes wrong.
4. Stop the Docker image.
5. Build the new base image for PingGateway.
6. Run the Docker image.
7. In a test environment that simulates your production environment, validate that the upgraded service performs as expected with the new configuration. Check the logs for new or unexpected notifications or errors.
8. Allow client application traffic to flow to the upgraded site.

Post upgrade tasks

After upgrade, review the what’s new section in the release notes and consider activating new features and functionality.

Rollback

IMPORTANT

IMPORTANT

Before you roll back to a previous version of PingGateway, consider whether any change to the configuration during or since upgrade could be incompatible with the previous version.

Roll back with binaries

1. Plan for server downtime

   Plan to route client applications to another server until the rollback process is complete and you have validated the result. Make sure the owners of client application are aware of the change, and let them know what to expect.

2. Stop PingGateway
3. Download the replacement PingGateway .zip file
4. Make the new configuration available on the file system.

   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.

5. Restart PingGateway.

Roll back with Dockerfiles

1. Plan for server downtime

   Plan to route client applications to another server until the rollback process is complete and you have validated the result. Make sure the owners of client application are aware of the change, and let them know what to expect.

2. Stop the Docker image.
3. Build the new base image for PingGateway.
4. Run the Docker image.

Migrate from web container mode to standalone mode

An PingGateway .war file isn't created or delivered from PingGateway 2024.3. Consider these points when migrating from a .war delivery to a .zip delivery.

Session replication between PingGateway instances

High-availability of sessions isn't supported by PingGateway in the .zip delivery.

Streaming asynchronous responses and events

In ClientHandler and ReverseProxyHandler, use only the default mode of `asyncBehavior:non_streaming`; responses are processed when the entity content is entirely available.

If the property is set to `streaming`, the setting is ignored.

Connection reuse when client certificates are used for authentication

In ClientHandler and ReverseProxyHandler, use only the default mode of `stateTrackingEnabled:true`; when a client certificate is used for authentication, connections can’t be reused.

If the property is set to `false`, the setting is ignored.

Replacement settings for migration from web container mode with Tomcat

Feature	Setting for web container mode with Tomcat	Replacement setting
Port number	Configure in the `Connector` element of `/path/to/tomcat/conf/server.xml`: `<Connector port="8080" protocol="HTTP/1.1" connectionTimeout="20000" redirectPort="8443" />`	Configure the `connectors` property of admin.json.
HTTPS server-side configuration	Create a keystore, and set up the SSL port in the `Connector` element of `/path/to/tomcat/conf/server.xml`.	Create a keystore, set up secrets, and configure secrets stores, ports, and ServerTlsOptions in admin.json. For information, refer to Configure PingGateway for TLS (server-side).
Session cookie name	Configure `WEB-INF/web.xml` when you unpack the PingGateway .war file.	Configure the `session` property of admin.json.

Feature	Setting for web container mode with Tomcat	Replacement setting
Access logs	Configure with `AccessLogValve`.	Configure in the Audit framework. For information, refer to Audit the deployment and Audit framework.
JDBC datasource	Configure in the `GlobalNamingResources` element of `/path/to/tomcat/conf/server.xml`.	Configure with the JdbcDataSource object. For information, refer to JdbcDataSource. For an example, refer to Password replay from a database.
Environment variables	Configure in `/path/to/tomcat/bin/setenv.sh`.	Configure in `$HOME/.openig/bin/env.sh`, where `$HOME/.openig` is the instance directory.
Jar files	Add to to web container classpath; for example `/path/to/tomcat/webapps/ROOT/WEB-INF/lib`.	Add to `$HOME/.openig/extra`, where `$HOME/.openig` is the instance directory.

Was this helpful? 👍 👎