IBM Aspera Faspex 5.0 Guide

Contents

Welcome to Faspex

As easy as email

Faspex is a centralized transfer solution that enables users to exchange files with each other using an email-like workflow. Faspex enables high-speed transfers using IBM Aspera's proprietary FASP protocol, fully utilizing available network bandwidth to maximize speed while preserving control and security. User-uploaded files and folders are sent to, stored on, and downloaded from Aspera transfer servers.

Use Faspex to send files and folders to other members of your organization. Sending files and folders through Faspex is similar to sending an email:

Step	Instructions
Start sending a new package.	Click **Send files** to open the send form.
Choose your package recipients.	Enter contacts in the **To** field.
Add your content to the package.	Drag-and-drop files and folders from your computer.
Send the package.	Click **Send**.

When you send a package, you upload content to an Aspera transfer server for storage. Faspex notifies the recipients that the package is available. Recipients can then download a copy of the package from the remote server.

Faspex uses IBM Aspera Connect to facilitate high-speed uploads and downloads with an Aspera transfer server. Connect integrates all of Aspera's high-performance transport technology in a small, easy-to-use package that provides unequaled control over transfer parameters. An administrator can also provide an alternative form of transfer by configuring the IBM Aspera HTTP Gateway with Faspex, enabling users to transfer without using Connect.

Some Faspex file-exchange and management features include:

- Send a package to a list of recipients by sending to a distribution list, a shared inbox, or a workgroup.
- Forward packages on the server to other users without re-uploading files and folders.
- Manage user permissions through membership in workgroups and shared inboxes, or by direct configuration.
- Create customizable email notifications for Faspex events (such as receiving a package).
- Import members from your SAML providers.

Faspex administrators

You can find topics and procedures specific to administration under "Admin" on page 65.

Quick links

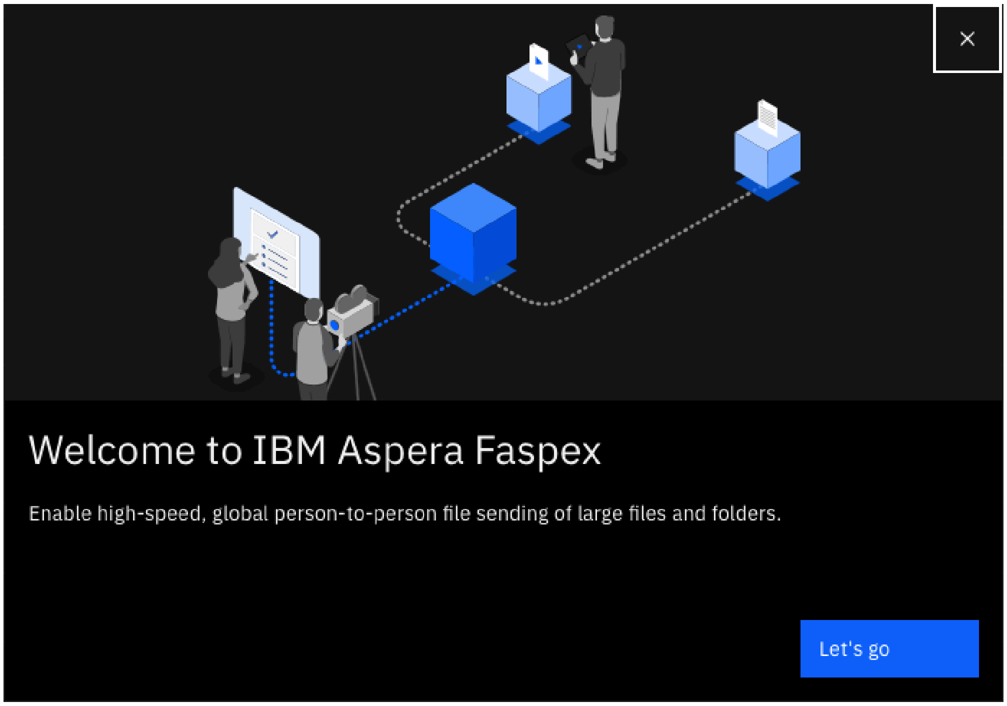
- "Differences and breaking changes between Faspex 5 and previous versions" on page 4.
- "Installing Faspex for the first time" on page 7
- "Upgrading Faspex before 5.X" on page 13

Faspex and Connect

IBM Aspera Connect is an install-on-demand browser extension and desktop client that facilitates high-speed uploads and downloads with an Aspera transfer server. You must install Connect to upload to and download from Faspex, unless you have HTTP Gateway.

Connect installation prompt

When you log in to Faspex, Faspex prompts you to install Connect unless you already have Connect installed or your admin suppressed the installation prompt.



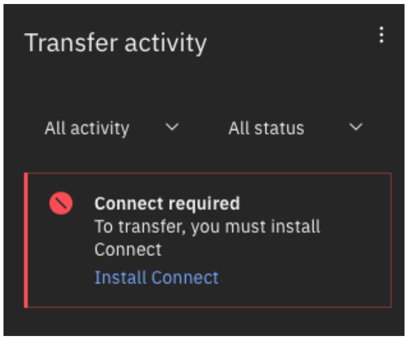
Welcome to IBM Aspera Faspex

Enable high-speed, global person-to-person file sending of large files and folders.

Click **Let's go** to go to a page where you can install the client and the extension.

If you do not want to install Connect, you can opt out of using Connect by clicking **I'll skip for now**.

⚠ **Warning:** If you opt out, you cannot upload to and download from Faspex unless HTTP Gateway is enabled by an admin. You can download Connect after opting out by opening the transfer monitor and clicking **Install Connect**.



Transfer activity

Connect required
To transfer, you must install Connect

For more information, see "Using HTTP Gateway instead of Connect" on page 46.

Navigating Faspex applications

Faspex divides features into two applications: Packages and Admin.

Application	Usage	Required permissions	Navigation
Packages	Send and download packages	User	Click the application switcher icon (⁝⁝⁝) and select **Packages** from the drop-down menu.
Admin	Configure Faspex server settings, manage packages, and manage user accounts	Admin Manager (limited view) Workgroup admin (limited view)	Click the application switcher icon (⁝⁝⁝) and select **Admin** from the drop-down menu.

User roles

Admins assign user roles to an account when creating a new account or when configuring an account's permissions.

Important: You cannot change your own assigned role.

User accounts can have these user roles:

Role	Description
Regular user	Regular users can send and receive packages, as permitted by admin-configured server settings.
Manager	In addition to regular user permissions, managers can manage: • regular users • workgroups • external users • SAML groups Managers can access all shared inboxes and manage shared inbox members and workgroups. Managers cannot create new managers, edit admin accounts, or promote another user to an admin or manager role.
Admin	In addition to manager permissions, admins can adjust server configurations, access all packages and relays, manage all workgroups and shared inboxes, and manage all users.
External user	A external user is a user that is not associated with a Faspex user account. Faspex users can send *public packages* to external users with a *public submission link* and invite external users to send them a package through an email invitation or with a *public submission link*.

Role	Description
	By default, external users do not need to create an account to download a public package, but an admin can configure Faspex to require external users to create an account before downloading the package (see "Allowing public packages" on page 105).

Differences and breaking changes between Faspex 5 and previous versions

Why Faspex 5?

Faspex 5 offers the following advantages:

- Modern API and enhanced security: Faspex 5, provides a state-of-the-art API that seamlessly integrates with your existing systems. Also, Faspex 5 includes security fixes and robust security models, safeguarding your information against potential threats.
- Updated infrastructure for optimal performance: Faspex 5 introduces several infrastructure enhancements. Containerization and upgraded components, such as the latest version of Rails, the transition from Mongrel to Puma, and support for TLS 1.3, ensure optimal performance and speed. The modern database and High Availability (HA) capabilities guarantee seamless failover with this true High Availability architecture.
- Forward looking compliance: By adhering to FIPS (Federal Information Processing Standards), we ensure that your system aligns with industry-leading standards.
- Rapid upgrades and updates: Because Faspex 5 is offered in a containerized format, you can update the software with a simple command.
- Rapid implementation of updates: The Faspex 5 software stack enables faster development to address customer enhancement requests.

What is the difference between Faspex 4.X and Faspex 5.0?

Architecture

Faspex 5 uses a new platform architecture, which is the foundation for consolidating Aspera web applications. Currently, the Faspex 5 platform has two applications:

- The Packages application for sending and downloading packages (Faspex 4).
- The Admin application for configuring server settings and manage users.

In the future, the Faspex 5 platform is expected to include applications to share content in folders (Shares) and to monitor and automate transfers (Console).

As part of the new architecture, Faspex 5 is fully containerized and runs multiple containers. Faspex 5 is fully REST API driven and the Faspex 5 web interface leverages that API. Any external integration developers looking to access server resources will use the same API as the Faspex 5 web interface.

Faspex 5 uses Nginx as a reverse proxy to direct traffic to the different containers. You can modify the `nginx.conf` configuration file to adjust Nginx server settings and to use your own SSL certificates. See "Configure Nginx settings" on page 37.

Navigation and display

Faspex 5 has a completely new web user interface created with the IBM Carbon Design framework (and dark mode is available). The Faspex 5 web interface is a single-page reactive application, allowing streamlined, new user experiences. One example is the transfer activity panel that allows a user to keep monitoring transfers as the user switches from one application to another. Another example is a user can make multiple HTTP Gateway uploads without blocking navigation.

Shared inboxes (formerly dropboxes)

Dropboxes are now called shared inboxes. When you send a package to shared inboxes, instead of selecting a shared inbox from a drop-down menu, simply include the shared inbox as a package recipient.

Manage shared inboxes in the left sidebar of the Packages application.

Database

Faspex 5 uses MariaDB for the database. Even though Faspex 5 ships with a `faspex_db` container (running a MariaDB database), as a best practice use a compatible, external database with Faspex 5, especially for high availability deployments.

The Faspex 5 database schema and backups are now managed using the Faspex Utility web application, accessed at `https://`*`your_faspex_server`*`/aspera/faspex/utility`. Use Faspex Utility to backup, restore, and migrate the Faspex database schema and content. Make sure that you installed and are running the Utility application if you plan on using it for database management. See "The Faspex Utility web application" on page 71 for more information.

Authentication and authorization

Faspex 5 adopts OAuth 2 as the authorization mechanism for its APIs. The Faspex V3 API used less secure HTTP basic authorization and the Faspex V4 API did not decouple user authentication from authorization. A Faspex 5 administrator can register an API client to retrieve a bearer token to interact with the endpoints. For more information about the Faspex V5 API, see the Developer guide.

Feature name changes

- File storage is now called nodes and storage.
- File storage shares are now called storage locations.
- In the Admin **Configuration** menu, **Post processing** is now called **Package processing**. **Non-blocking post processing** is supported starting in Faspex 5.0.4, and is now called **Package processing webhooks**.
- On the send package form, **Obfuscation** is now called **Mask file names**.
- On the send package form, the **Encryption** option is now called **Password protection**. You now provide the password in the send form instead of in Connect.
- On the send package form, the **Obfuscation** option is now called **Mask file names**.
- On the send package form, the **Show Private Recipients** link is now called **Bcc**.
- Editing the template user is now called configuring self-registered user defaults.

Breaking changes

Accounts

- To log in to new Faspex 5 accounts, you must now use the user account email address. If you upgraded from Faspex 4.X, you can still log in to existing user accounts with their Faspex 4.X usernames.
- New Faspex 5 users must change their password on first login. You can no longer disable this requirement in server settings.
- Admins can no longer set another user's password. A user can still change their own password.
- Admins can no longer reset another user's password. A user can still reset their own password.
- Admins can no longer choose to display users using their usernames. Faspex always displays a user's first name and last name.

Directory services

For security reasons, Faspex 5 does not support directory services. You must instead front your directory service with a SAML Identity Provider (IdP) and use SAML based authentication for your users.

SAML

The SAML metadata and callback URL routes are different from previous versions. Retrieve the new metadata and callback URLs from Faspex and update your SAML Identity Provider (see "Reconfigure SAML after upgrading to Faspex 5" on page 24).

Sending packages

- Instead of using the `(external)` flag to allow a Faspex user to download a package without logging in, enable **Recipients with an account can download without logging in** when sending a package. This feature requires an admin to turn on **Senders can allow IBM Aspera users to download their packages without logging in**. For more information, see "Allowing public packages" on page 105.
- The send package form no longer prepends an asterisk (*) to workgroup and shared inbox names.
- The send package form no longer presents the option to choose whether to send with Connect or HTTP Gateway. To send using HTTP Gateway, open the transfer activity monitor and enable **Force to use Aspera HTTP Gateway** (if HTTP Gateway is available).

Metadata profiles

Faspex 5 does not support previewing a metadata profile.

Nodes

- Nodes added using `localhost` and `127.0.0.1` may need to be re-added with their appropriate public IP address or hostnames. Faspex no longer automatically configures and adds a collocated HSTS on install.
- You must upgrade your existing nodes to HSTS 4.3 and later *before upgrading*. Faspex 5 uses the HSTS 4.3+ activity logging feature to retrieve transfer information. See "Enable activity logging on a HSTS node" on page 23.
- For setups where collocation of Faspex 5 with HSTS is unavoidable, the IP address included in the node configuration in Faspex 5 can no longer be `127.0.0.1` or `localhost`. This is because Faspex 5 is containerized and `localhost` refers to the container itself, not the server that is running the container. Use the private IP address or FQDN of the server instead.

HTTP fallback

The **`asperahttpd`** service in HSTS does not currently report HTTP transfer activity through the `/ops/transfers` endpoint. For this reason, Faspex does not support HTTP fallback until HSTS addresses this use case.

Customization

Faspex 5 no longer supports custom CSS, custom HTML or custom Ruby. Use the branding options available in **Configurations > Display settings** (see "Configure display settings" on page 94).

Faspex 5 no longer supports the `faspex.yml` configuration file. If you rely on options in `faspex.yml` that cannot be set in another way in Faspex 5, then do not upgrade. For a list of supported and unsupported options, see "faspex.yml options in Faspex 5" on page 23.

Faspex 5 does not currently support out-of-transfer file validation or post-processing scripts.

Integration

Faspex 5 no longer provides rake tasks for automating tasks. Instead, use the Faspex 5 API to perform automation.

The Faspex 5 API is not backwards-compatible with prior versions of the APIs.

Post processing

Lua scripts are not supported in Faspex 5. To migrate the post processing scripts from Faspex 4, review the Configuring package processing webhooks section to prepare the webhooks.

Light mode and dark mode

Faspex 5 uses, by default, the color mode exposed by the browser and set by your computer.

You can change the mode by clicking the profile icon (👤) in the top-right of the banner and selecting either **Switch to light mode** or **Switch to dark mode** from the drop-down menu.

Note: User color-mode selection is not saved between sessions. If you change to dark mode but your computer defaults to light mode, Faspex uses light mode the next time you sign in.

Installation and upgrades

Faspex 5 is a web application. Installation only applies to system administrators.

Installing Faspex for the first time

Installing Faspex with the default database

Perform a clean install on a new machine using the provided database.

Requirements

Operating system

Faspex 5 is supported only on Linux. Your Linux server must run on CentOS 7, RHEL 7, or RHEL 8.

Important: Docker and Podman cannot be installed in the same instance.

Your Linux server must run Docker for Faspex versions 5.0.0 to 5.0.3. Faspex 5.0.4+ users can run either Docker or Podman. For tips on installing Docker on RHEL 8, see #unique_27.

Note: Podman is only supported on RHEL 8.

You should use a server with at least 16GB of available RAM with at least 4 CPU cores.

Your server must also have network access to the IBM Cloud Container Registry (icr.io). The installer pulls container images from `icr.io`.

IBM Aspera High-Speed Transfer Server (HSTS) version requirement

Faspex 5 uses *nodes* to store user-uploaded content. Your nodes must run HSTS version 4.3 or later to retrieve package information.

Aspera ecosystem

IBM only supports collocating Faspex 5 with HTTP Gateway. Do not collocate Faspex 5 with HSTS or any other Aspera application. You should create separate silos for the control plane (Faspex 5) and data plane (HSTS). If collocation is unavoidable due to specific requirements, it is advisable to create silos with different dedicated system users to run Faspex 5 and HSTS. Dedicated system users should not share access to any files.

Tip: Collocating an HTTP Gateway server with Faspex 5 requires changing the default TCP port or fronting it with Nginx.

Installation steps

1. **Note:** For Podman and operating system compatibility, see the Faspex 5 Release notes.

   Install these Podman components: `podman-docker`, `podman-plugins`, and `skopeo`.

   ```
   yum install podman-docker podman-plugins skopeo
   ```

2. Download and install the RPM file:

```
rpm -ivh ibm-aspera-faspex-version.build.x86_64.rpm
```

3. You can customize deployment options by editing the `.env` files located at `/opt/aspera/faspex/conf/docker`. These options include:

   - Which ports to use for the various containers
   - Where to access the database (if you choose not to install the container)
   - What credentials to use for the database
   - What local folder to mount database backups

   For more information, see "Reference: Faspex configuration files" on page 149.

4. Install Faspex containers:

```
faspexctl setup
```

5. Select the containers that you would like to install, set the server address, and provide the credentials to create the first admin user.
6. Confirm all containers are present and running:

```
faspexctl status
```

7. Log into your Faspex UI at `https://your_server_hostname/aspera/faspex`.
8. Add a node:

   For instruction on setting up a *node*, see "Adding a node to Faspex" on page 76.

   a) In the Admin app, go to **Nodes and Storage**.

   b) Click **Create node**.

   c) Enter a unique name to identify the node.

   d) To encrypt the connection to the node using SSL, enable **Use SSL**.

   e) To verify the SSL certificate, enable **Verify SSL Certificate**.

   f) Provide Faspex the information needed to connect to the Node API on the transfer server:

Field	Description
Host	The node's hostname or IP address. To avoid connectivity problems, do not specify a hostname that contains underscores.
Port	The Node API port number. By default, the port is 9092.
Username	The Node API username on the node machine.
Password	The Node API password on the node machine.

   g) Click **Create**.

9. Add a default storage location to Faspex.

   a) Select the node you just created.

   b) Go to the **Storage locations** tab.

   c) Click **Create storage location**.

   d) Enter a name for the file storage.

   e) Click **Create**.

   f) Right-click the storage location you just created and select **Make default inbox** from the overflow menu.

10. Connect an email notification server for notifications.

    Faspex sends a welcome email to a new user's email account when a new user is created. New users cannot set their password and log in without the welcome email.

11. Send a file to yourself to test that Faspex is working:
    a) Switch to the **Packages** application.
    b) Click **Send files**.
    c) Add yourself to the **To** field.
    d) Enter a package title.
    e) Add a test file to the package by clicking the **Add files +** button, selecting **Files** from the drop-down, and choosing a file.
    f) Click **Send**.
    g) Go to **Received** and find the package you sent yourself.
    h) Right-click and select **Download** from the drop-down menu.

Installing Faspex with a remote database

Perform a clean install on a new machine using your own database. The `faspex-db` container is provided as a convenience. Customers are free to bring their own external database, especially for scalable, high-availability environments. Any external database must fulfill the Faspex 5 database requirements. For information about supported databases, see the release notes.

Requirements

Operating system

Faspex 5 is supported only on Linux. Your Linux server must run on CentOS 7, RHEL 7, or RHEL 8.

Important: Docker and Podman cannot be installed in the same instance.

Your Linux server must run Docker for Faspex versions 5.0.0 to 5.0.3. Faspex 5.0.4+ users can run either Docker or Podman. For tips on installing Docker on RHEL 8, see #unique_27.

Note: Podman is only supported on RHEL 8.

You should use a server with at least 16GB of available RAM with at least 4 CPU cores.

Your server must also have network access to the IBM Cloud Container Registry (icr.io). The installer pulls container images from `icr.io`.

IBM Aspera High-Speed Transfer Server (HSTS) version requirement

Faspex 5 uses *nodes* to store user-uploaded content. Your nodes must run HSTS version 4.3 or later to retrieve package information.

Aspera ecosystem

IBM only supports collocating Faspex 5 with HTTP Gateway. Do not collocate Faspex 5 with HSTS or any other Aspera application. You should create separate silos for the control plane (Faspex 5) and data plane (HSTS). If collocation is unavoidable due to specific requirements, it is advisable to create silos with different dedicated system users to run Faspex 5 and HSTS. Dedicated system users should not share access to any files.

Tip: Collocating an HTTP Gateway server with Faspex 5 requires changing the default TCP port or fronting it with Nginx.

Installation steps

On the remote server, provision required users and create the Faspex 5 database. Then run the Faspex 5 installer and configure the Faspex 5 environment files to connect to the remote database before finishing the installation.

Note: Refer to the Enabling TLS to connect to the database section if you need to connect to the database using TLS.

Important: If you were using MySQL 5.7 (not supported after Faspex 5.0.5), you need to create a database backup and setup and configure a new remote instance of MySQL 8.0 following the steps in this section. For information on how to install MySQL 8.0, follow the official MySQL documentation.

1. Configure the remote database for use with Faspex 5. On the remote database server, enter the MySQL console:

   ```
   mysql -u root_user -p
   ```

   Faspex 5 uses three database users:

   - A root-level user for creating the database and for provisioning the other two users. Faspex 5 only uses this user when configuring the `faspex-db` container. In this documentation, this user is `root`.
   - An admin user for updating, creating, and dropping tables. This user is restricted to the Faspex 5 Utility application for migrations. In this documentation, this user is `aspera`.
   - A restricted user for performing production read and write operations. In this documentation, this user is `faspex`.

   a) Disable the `ONLY_FULL_GROUP_BY` SQL mode:

   ```
   SET GLOBAL sql_mode=(SELECT REPLACE(@@sql_mode,'ONLY_FULL_GROUP_BY',''));
   ```

   Note: MariaDB on Amazon RDS does not have `ONLY_FULL_GROUP_BY` enabled by default. You can check if your instance has this SQL mode enabled by accessing your remote database through the MySQL console and running:

   ```
   SELECT @@SESSION. sql_mode;
   ```

   If you have `ONLY_FULL_GROUP_BY` enabled, you must use an Amazon DB parameter group to change the mode on your database.

   b) Create an admin user (`aspera`) for updating, creating, and dropping tables and grant the user access to the required tables.

   ```
   ## Create an admin user named 'aspera' with password '<REPLACE_WITH_SECURE_PASSWORD>'
   CREATE USER IF NOT EXISTS 'aspera'@'%' IDENTIFIED WITH mysql_native_password BY
   '<REPLACE_WITH_SECURE_PASSWORD>';
   ALTER USER 'aspera'@'%' IDENTIFIED BY '<REPLACE_WITH_SECURE_PASSWORD>';
   CREATE USER IF NOT EXISTS 'aspera'@'localhost'IDENTIFIED BY
   '<REPLACE_WITH_SECURE_PASSWORD>';

   # Grant the admin user permissions to the mysql and faspex databases
   GRANT SELECT ON `mysql`.* TO 'aspera'@'%';
   GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, INDEX, ALTER, CREATE TEMPORARY
   TABLES, CREATE VIEW, EVENT, TRIGGER, SHOW VIEW, LOCK TABLES, CREATE ROUTINE, ALTER
   ROUTINE, EXECUTE ON `faspex%`.* TO 'aspera'@'%';
   ```

   c) Create a restricted user (`faspex`) for production read and write operations and grant the user access to the required tables.

   ```
   # Create a restricted user named 'faspex' with password
   CREATE USER IF NOT EXISTS 'faspex'@'%' IDENTIFIED WITH mysql_native_password BY
   '<REPLACE_WITH_SECURE_PASSWORD>;
   ALTER USER 'faspex'@'%' IDENTIFIED BY '<REPLACE_WITH_SECURE_PASSWORD>;

   # Grant the restricted user limited permissions to the faspex database
   GRANT SELECT, INSERT, UPDATE, DELETE ON `faspex%`.* TO 'faspex'@'%';
   ```

   d) Create the Faspex database and flush privileges:

   ```
   CREATE DATABASE IF NOT EXISTS faspex;
   FLUSH PRIVILEGES;
   ```

2. On your Faspex 5 server, install these Podman components: `podman-docker`, `podman-plugins`, and `skopeo`.

   ```
   yum install podman-docker podman-plugins skopeo
   ```

3. On your Faspex 5 server, run the Faspex 5 installer, but **do not** run **`faspexctl setup`**.

   ```
   rpm -ivh ibm-aspera-faspex-version.build.x86_64.rpm
   ```

4. Configure the generated database environment variables configured by the `.env` files in `/opt/aspera/faspex/conf/docker`:

   `db.env`

Variable	Description	Example
`FASPEX_DB_HOST`	Remote database hostname	`mariadb-remote.example.com`
`FASPEX_DB_PORT`	Remote database port	`3306`
`FASPEX_DB_USERNAME`	Restricted user name	`faspex`
`FASPEX_DB_PASSWORD`	Restricted user password	`0bcecff1-e7df-4596-94f2-1a4b241be252`

   `db_admin.env`

Variable	Description	Example
`FASPEX_DB_ASPERA_ADMIN_USERNAME`	Admin user name	`aspera`
`FASPEX_DB_ASPERA_ADMIN_PASSWORD`	Admin user password	`ed953178-8d04-4101-96f7-77f61ca9ac0a`

   Note: You can leave the `FASPEX_DB_ROOT_USERNAME` and `FASPEX_DB_ROOT_PASSWORD` variables unset, as a remote database set up does not require access to the root user.

5. Run `faspexctl setup`.

   Make sure you respond no (n) when the installer prompts you to install the db container.

6. Log into your Faspex UI at `https://`*`your_server_hostname`*`/aspera/faspex`.

7. Add a node:

   For instruction on setting up a *node*, see “Adding a node to Faspex” on page 76.

   a) In the Admin app, go to **Nodes and Storage**.

   b) Click **Create node**.

   c) Enter a unique name to identify the node.

   d) To encrypt the connection to the node using SSL, enable **Use SSL**.

   e) To verify the SSL certificate, enable **Verify SSL Certificate**.

   f) Provide Faspex the information needed to connect to the Node API on the transfer server:

Field	Description
Host	The node's hostname or IP address. To avoid connectivity problems, do not specify a hostname that contains underscores.
Port	The Node API port number. By default, the port is 9092.
Username	The Node API username on the node machine.
Password	The Node API password on the node machine.

   g) Click **Create**.

8. Add a default storage location to Faspex.

a) Select the node you just created.

b) Go to the **Storage locations** tab.

c) Click **Create storage location**.

d) Enter a name for the file storage.

e) Click **Create**.

f) Right-click the storage location you just created and select **Make default inbox** from the overflow menu.

9. Connect an email notification server for notifications.

   Faspex sends a welcome email to a new user's email account when a new user is created. New users cannot set their password and log in without the welcome email.
10. Send a file to yourself to test that Faspex is working:

    a) Switch to the **Packages** application.

    b) Click **Send files**.

    c) Add yourself to the **To** field.

    d) Enter a package title.

    e) Add a test file to the package by clicking the **Add files +** button, selecting **Files** from the drop-down, and choosing a file.

    f) Click **Send**.

    g) Go to **Received** and find the package you sent yourself.

    h) Right-click and select **Download** from the drop-down menu.

Upgrading Faspex from 5.X

Warning:

Before proceeding with any upgrade, you should verify the following:

1. Perform a full environment back up and ensure the back up is successful. In case the upgrade fails, the only reliable, short-term fix is to roll back the environment using the back up.
2. Test the upgrade in a test environment comparable to the production environment.
3. If upgrading the test environment is successful, upgrade the production environment, but do not bring the production environment back online.
4. Prior to bringing the production environment back online, the customer must test the application to determine if an immediate rollback is needed. Otherwise, customers risk losing all data generated between upgrade and rollback. If a rollback is needed to undo the installation of patch containers on a server, you would be able to rollback to a specific version of Faspex that has not been patched via `faspexctl set_image_tags VERSION`.

   Note: You should not attempt to perform a rollback across different patch Faspex versions (ie. from 5.0.5 to 5.0.4).

1. Fully back up your Faspex instance.
2. Install these Podman components: `podman-docker`, `podman-plugins`, and `skopeo`.

   Note: If you are performing the upgrade from a podman-docker setup you can skip this step.

   ```
   yum install podman-docker podman-plugins skopeo
   ```
3. Download the newest installer and install the RPM file:

   ```
   rpm -Uvh ibm-aspera-faspex-version.build.x86_64.rpm
   ```
4. Update Faspex containers:

```
faspexctl setup
```

Note: If you are using a remote database, make sure to say no `(n)` to installing the `faspex_db` container.

5. Confirm all containers are present and running:

   ```
   faspexctl status
   ```

6. Open Faspex Utility and login.
7. Go to **Manage database** and make sure that Faspex is using the latest schema. If there is a new migration, migrate the database.

Upgrading Faspex before 5.X

Differences and breaking changes between Faspex 5 and previous versions

Why Faspex 5?

Faspex 5 offers the following advantages:

- Modern API and enhanced security: Faspex 5, provides a state-of-the-art API that seamlessly integrates with your existing systems. Also, Faspex 5 includes security fixes and robust security models, safeguarding your information against potential threats.
- Updated infrastructure for optimal performance: Faspex 5 introduces several infrastructure enhancements. Containerization and upgraded components, such as the latest version of Rails, the transition from Mongrel to Puma, and support for TLS 1.3, ensure optimal performance and speed. The modern database and High Availability (HA) capabilities guarantee seamless failover with this true High Availability architecture.
- Forward looking compliance: By adhering to FIPS (Federal Information Processing Standards), we ensure that your system aligns with industry-leading standards.
- Rapid upgrades and updates: Because Faspex 5 is offered in a containerized format, you can update the software with a simple command.
- Rapid implementation of updates: The Faspex 5 software stack enables faster development to address customer enhancement requests.

What is the difference between Faspex 4.X and Faspex 5.0?

Architecture

Faspex 5 uses a new platform architecture, which is the foundation for consolidating Aspera web applications. Currently, the Faspex 5 platform has two applications:

- The Packages application for sending and downloading packages (Faspex 4).
- The Admin application for configuring server settings and manage users.

In the future, the Faspex 5 platform is expected to include applications to share content in folders (Shares) and to monitor and automate transfers (Console).

As part of the new architecture, Faspex 5 is fully containerized and runs multiple containers. Faspex 5 is fully REST API driven and the Faspex 5 web interface leverages that API. Any external integration developers looking to access server resources will use the same API as the Faspex 5 web interface.

Faspex 5 uses Nginx as a reverse proxy to direct traffic to the different containers. You can modify the `nginx.conf` configuration file to adjust Nginx server settings and to use your own SSL certificates. See “Configure Nginx settings” on page 37.

Navigation and display

Faspex 5 has a completely new web user interface created with the IBM Carbon Design framework (and dark mode is available). The Faspex 5 web interface is a single-page reactive application, allowing streamlined, new user experiences. One example is the transfer activity panel that allows a user to keep monitoring transfers as the user switches from one application to another. Another example is a user can make multiple HTTP Gateway uploads without blocking navigation.

Shared inboxes (formerly dropboxes)

Dropboxes are now called shared inboxes. When you send a package to shared inboxes, instead of selecting a shared inbox from a drop-down menu, simply include the shared inbox as a package recipient.

Manage shared inboxes in the left sidebar of the Packages application.

Database

Faspex 5 uses MariaDB for the database. Even though Faspex 5 ships with a `faspex_db` container (running a MariaDB database), as a best practice use a compatible, external database with Faspex 5, especially for high availability deployments.

The Faspex 5 database schema and backups are now managed using the Faspex Utility web application, accessed at `https://`*`your_faspex_server`*`/aspera/faspex/utility`. Use Faspex Utility to backup, restore, and migrate the Faspex database schema and content. Make sure that you installed and are running the Utility application if you plan on using it for database management. See "The Faspex Utility web application" on page 71 for more information.

Authentication and authorization

Faspex 5 adopts OAuth 2 as the authorization mechanism for its APIs. The Faspex V3 API used less secure HTTP basic authorization and the Faspex V4 API did not decouple user authentication from authorization. A Faspex 5 administrator can register an API client to retrieve a bearer token to interact with the endpoints. For more information about the Faspex V5 API, see the Developer guide.

Feature name changes

- File storage is now called nodes and storage.
- File storage shares are now called storage locations.
- In the Admin **Configuration** menu, **Post processing** is now called **Package processing**. **Non-blocking post processing** is supported starting in Faspex 5.0.4, and is now called **Package processing webhooks**.
- On the send package form, **Obfuscation** is now called **Mask file names**.
- On the send package form, the **Encryption** option is now called **Password protection**. You now provide the password in the send form instead of in Connect.
- On the send package form, the **Obfuscation** option is now called **Mask file names**.
- On the send package form, the **Show Private Recipients** link is now called **Bcc**.
- Editing the template user is now called configuring self-registered user defaults.

Breaking changes

Accounts

- To log in to new Faspex 5 accounts, you must now use the user account email address. If you upgraded from Faspex 4.X, you can still log in to existing user accounts with their Faspex 4.X usernames.
- New Faspex 5 users must change their password on first login. You can no longer disable this requirement in server settings.
- Admins can no longer set another user's password. A user can still change their own password.
- Admins can no longer reset another user's password. A user can still reset their own password.
- Admins can no longer choose to display users using their usernames. Faspex always displays a user's first name and last name.

Directory services

For security reasons, Faspex 5 does not support directory services. You must instead front your directory service with a SAML Identity Provider (IdP) and use SAML based authentication for your users.

SAML

The SAML metadata and callback URL routes are different from previous versions. Retrieve the new metadata and callback URLs from Faspex and update your SAML Identity Provider (see "Reconfigure SAML after upgrading to Faspex 5" on page 24).

Sending packages

- Instead of using the `(external)` flag to allow a Faspex user to download a package without logging in, enable **Recipients with an account can download without logging in** when sending a package. This feature requires an admin to turn on **Senders can allow IBM Aspera users to download their packages without logging in**. For more information, see "Allowing public packages" on page 105.
- The send package form no longer prepends an asterisk (*) to workgroup and shared inbox names.
- The send package form no longer presents the option to choose whether to send with Connect or HTTP Gateway. To send using HTTP Gateway, open the transfer activity monitor and enable **Force to use Aspera HTTP Gateway** (if HTTP Gateway is available).

Metadata profiles

Faspex 5 does not support previewing a metadata profile.

Nodes

- Nodes added using `localhost` and `127.0.0.1` may need to be re-added with their appropriate public IP address or hostnames. Faspex no longer automatically configures and adds a collocated HSTS on install.
- You must upgrade your existing nodes to HSTS 4.3 and later *before upgrading*. Faspex 5 uses the HSTS 4.3+ activity logging feature to retrieve transfer information. See "Enable activity logging on a HSTS node" on page 23.
- For setups where collocation of Faspex 5 with HSTS is unavoidable, the IP address included in the node configuration in Faspex 5 can no longer be `127.0.0.1` or `localhost`. This is because Faspex 5 is containerized and `localhost` refers to the container itself, not the server that is running the container. Use the private IP address or FQDN of the server instead.

HTTP fallback

The **`asperahttpd`** service in HSTS does not currently report HTTP transfer activity through the `/ops/transfers` endpoint. For this reason, Faspex does not support HTTP fallback until HSTS addresses this use case.

Customization

Faspex 5 no longer supports custom CSS, custom HTML or custom Ruby. Use the branding options available in **Configurations > Display settings** (see "Configure display settings" on page 94).

Faspex 5 no longer supports the `faspex.yml` configuration file. If you rely on options in `faspex.yml` that cannot be set in another way in Faspex 5, then do not upgrade. For a list of supported and unsupported options, see "faspex.yml options in Faspex 5" on page 23.

Faspex 5 does not currently support out-of-transfer file validation or post-processing scripts.

Integration

Faspex 5 no longer provides rake tasks for automating tasks. Instead, use the Faspex 5 API to perform automation.

The Faspex 5 API is not backwards-compatible with prior versions of the APIs.

Post processing

Lua scripts are not supported in Faspex 5. To migrate the post processing scripts from Faspex 4, review the Configuring package processing webhooks section to prepare the webhooks.

Upgrading a Faspex 4.X server

Upgrade from a version before Faspex 5.0 and confirm the persistence of files and data.

Note: IBM Aspera Faspex 4.x will no longer be supported after September 2023.

⚠ **Warning:** Do not continue with this upgrade until you have reviewed all the breaking changes and upgrade considerations.

Important:

If you are upgrading from a version before 4.4.1, first upgrade to 4.4.1. See the IBM Aspera Faspex 4.4.1 Admin Guide.

Review breaking changes and upgrade considerations

Operating system

Faspex 5 does not support Windows as an operating system. If you're currently using Windows, in order to upgrade to Faspex 5, migrate your Faspex 4.X instance to a Linux server and test before upgrading to Faspex 5 on Linux.

Your server must run CentOS7, RHEL 7, RHEL 8, or (starting in 5.0.4) Amazon Linux 2.

Your server must have Docker or (starting in 5.0.4) Podman 4.1 that is installed. For tips on installing Docker and Podman, see #unique_27.

Note: For Podman and operating system compatibility, see the Faspex 5 Release notes.

Your server must also have network access to the IBM Cloud Container Registry (icr.io). The installer pulls container images from icr.io.

Database

If you are using the default Faspex 4 database, the upgrade process migrates your data to the `faspex-db` container. If you have an external database running, you will have to back up your Faspex 4 external database and run migration against the external database.

Note: The following error is displayed during the upgrade: "`Column count of mysql.proc is wrong. Expected 21, found 20. Created with MariaDB 50737, now running 100803. Please use mariadb-upgrade to fix this error`" This is expected to happen and the installer will continue without any issues.

Directory services

For security reasons, Faspex 5 does not support directory services. You must instead front your directory service with a SAML Identity Provider (IdP) and use SAML based authentication for your users. Do not upgrade if you rely on a direct connection between Faspex and directory services. If you have Directory Services users, you will need to migrate those users to SAML before upgrading from Faspex 4 to Faspex 5.

High-Speed Transfer Server version requirement

Faspex 5 uses a more modern API for interacting with IBM Aspera High-Speed Transfer Server (HSTS). Faspex 5 uses the activity logging feature available on HSTS 4.3 or later. You must upgrade HSTS on your nodes and enable activity logging before upgrading (see "Enable activity logging on a HSTS node" on page 23).

For setups where collocation of Faspex 5 with HSTS is unavoidable, the IP address included in the node configuration in Faspex 5 can no longer be `127.0.0.1` or `localhost`. This is because Faspex 5 is containerized and `localhost` refers to the container itself, not the server that is running the container. Use the private IP address or FQDN of the server instead.

Connect version

Faspex 5 requires users run Connect version 4.1.3 or later and does not currently support locally hosting the Connect SDK.

Do not upgrade if your users rely on downloading Connect from the Faspex server (instead of the Cloudfront CDN). Hosting the Connect SDK and installers locally is expected in a future version of Faspex 5.

If your users cannot upgrade to Connect 4.1.3 and later, you can default users to transfer with HTTP Gateway.

Email addresses

Faspex 5 requires new users to have a unique email address. To log in to new Faspex 5 accounts, you must log in using the account email address. You can still log in using usernames for accounts created before the upgrade.

On upgrade, Faspex automatically reconciles external users that have the same email address as a regular user or a SAML user. While new Faspex users must log in using their email address, users created before the upgrade can still log in to those accounts with their usernames (but not with their email addresses).

After upgrading, you can see a list of users with duplicate email addresses by logging in to the Faspex Utility application. Then, from the Admin application, you can manually decide to keep, edit, or remove accounts with duplicate email addresses (see "Review accounts with duplicate email addresses" on page 109).

New users must change their password on first login

New Faspex 5 users must change their password on first login. You can no longer disable this requirement in server settings.

SAML

The SAML metadata and callback URL routes are different from previous versions. After upgrading, retrieve the new metadata and callback URLs from Faspex and update your SAML Identity Provider (see "Reconfigure SAML after upgrading to Faspex 5" on page 24).

Customization

Faspex 5 no longer supports custom CSS, custom HTML or custom Ruby. Use the branding options available in **Configurations > Display settings** (see "Configure display settings" on page 94).

Faspex 5 no longer supports the `faspex.yml` configuration file. If you rely on options in `faspex.yml` that cannot be set in another way in Faspex 5, then do not upgrade. For a list of supported and unsupported options, see "faspex.yml options in Faspex 5" on page 23.

Faspex 5 does not currently support out-of-transfer file validation or post-processing scripts. Starting in 5.0.4, Faspex supports package processing webhooks to notify external systems (through a POST request) that a package is ready for download.

Integration

Faspex 5 no longer provides rake tasks for automating tasks. Instead, use the Faspex 5 API to perform automation.

The Faspex 5 API is not backwards-compatible with prior versions of the APIs.

Public submission links

Public submission links and external user invitations created in Faspex 4 do not work in Faspex 5. Admins can resend the invitation as long as the invitation is not expired

Note: For a full list of differences, see "Differences and breaking changes between Faspex 5 and previous versions" on page 4.

If you have questions or issues with the upgrading process from Faspex 4 to Faspex 5, after reviewing the breaking changes and upgrade considerations, contact IBM support.

Upgrading steps

⚠ **Warning:**

Review the following before performing any upgrade:

1. Fully back up your environment and ensure that the back up is successful. In case the upgrade fails, the only reliable, short-term fix is to roll back the environment using the back up.
2. Test the upgrade in a test environment comparable to the production environment.
3. If upgrading the test environment is successful, upgrade the production environment, but do not bring the production environment back online.
4. Before bringing the production environment back online, the customer must test the application to determine if an immediate rollback is needed. Otherwise, customers risk losing all data generated between upgrade and rollback.

1. Back up your Faspex 4.X database.

   ```
   asctl faspex:backup_database
   ```

2. Backup your `/opt/aspera/faspex` folder entirely.
3. For each HSTS node, upgrade the IBM Aspera High-Speed Transfer Server version to 4.3 and enable activity logging. Do this before upgrading Faspex to retain package information. For instructions, see "Enable activity logging on a HSTS node" on page 23.

   If your Faspex 4 server is collocated with an HSTS node, you will need to add it again post-upgrade.

   If you are setting the token encryption key for users in your `aspera.conf` file, you should use dynamically generated token encryption keys instead. For more information, see the *IBM Aspera High-Speed Transfer Server:Secrets Management with askmscli* section.
4. If you have a collocated HSTS node

   Note: You should change the node host pre-upgrade, but it's possible to do this after as well.
5. Install Docker CE or Podman as needed. Review the system requirements section of the release notes to confirm OS compatibility with these components.
6. Download the newest installer and install the RPM file:

   ```
   rpm -Uvh ibm-aspera-faspex-version.build.x86_64.rpm
   ```

7. Update Faspex containers:

   ```
   faspexctl setup
   ```

   Agree to upgrade from Faspex 4.X and agree to clean the filesystem.
8. Confirm all containers are present and running:

   ```
   faspexctl status
   ```

9. Confirm the presence of legacy files and old database files:
   Legacy files are stored at `/opt/aspera/faspex_legacy`.
   Old database files are stored at `/opt/aspera/faspex/backup/`.
10. Confirm the upgrade persisted the data from your database. Run:

    ```
    docker exec -it faspex-db mysql faspex -e "show tables"
    ```

    You should be able to access all data from before the upgrade.
11. In order to migrate your SSL certificates installed, you must copy them from the `faspex_legacy` folder to the Nginx configuration folder:

    Note: If you are using the self-signed certificate from Faspex 4, you should NOT replace the certificates that Faspex 5 created.

    a) Rename and move your certificate file. Replace *server_cert_file* with either `server-ca.crt` or `server.crt` as needed:

```
cp -iv /opt/aspera/faspex_legacy/conf/server_cert_file /opt/aspera/faspex/conf/nginx/
cert.pem
```

b) Rename and move your key file (`server.key`):

```
cp -iv /opt/aspera/faspex_legacy/conf/server.key /opt/aspera/faspex/conf/nginx/key.pem
```

c) Restart `faspex-router`:

```
faspexctl restart router
```

12. Update your minimum version of Connect to 4.1.3 or later. Log in to the Faspex UI, go to **Transfer options** and update **Minimum Connect version**.

Upgrading a Faspex 4.X server with a remote database

Upgrade from a version before Faspex 5.0 and confirm the persistence of files and data.

Note: IBM Aspera Faspex 4.x will no longer be supported after September, 2023.

⚠ **Warning:** Do not continue with this upgrade until you have reviewed all the breaking changes and upgrade considerations.

Important:

If you are upgrading from a version before 4.4.1, first upgrade to 4.4.1. See the IBM Aspera Faspex 4.4.1 Admin Guide.

Review breaking changes and upgrade considerations

Operating system

Faspex 5 does not support Windows as an operating system. If you're currently using Windows, in order to upgrade to Faspex 5, migrate your Faspex 4.X instance to a Linux server and test before upgrading to Faspex 5 on Linux.

Your server must run CentOS7, RHEL 7, RHEL 8, or (starting in 5.0.4) Amazon Linux 2.

Your server must have Docker or (starting in 5.0.4) Podman 4.1 that is installed. For tips on installing Docker and Podman, see #unique_27.

Note: For Podman and operating system compatibility, see the Faspex 5 Release notes.

Your server must also have network access to the IBM Cloud Container Registry (icr.io). The installer pulls container images from icr.io.

Database

If you are using the default Faspex 4 database, the upgrade process migrates your data to the `faspex-db` container. If you have an external database running, you will have to back up your Faspex 4 external database and run migration against the external database.

Note: The following error is displayed during the upgrade: "`Column count of mysql.proc is wrong. Expected 21, found 20. Created with MariaDB 50737, now running 100803. Please use mariadb-upgrade to fix this error`" This is expected to happen and the installer will continue without any issues.

Directory services

For security reasons, Faspex 5 does not support directory services. You must instead front your directory service with a SAML Identity Provider (IdP) and use SAML based authentication for your users. Do not upgrade if you rely on a direct connection between Faspex and directory services. If you have Directory Services users, you will need to migrate those users to SAML before upgrading from Faspex 4 to Faspex 5.

High-Speed Transfer Server version requirement

Faspex 5 uses a more modern API for interacting with IBM Aspera High-Speed Transfer Server (HSTS). Faspex 5 uses the activity logging feature available on HSTS 4.3 or later. You must upgrade HSTS on your nodes and enable activity logging before upgrading (see "Enable activity logging on a HSTS node" on page 23).

For setups where collocation of Faspex 5 with HSTS is unavoidable, the IP address included in the node configuration in Faspex 5 can no longer be `127.0.0.1` or `localhost`. This is because Faspex 5 is containerized and `localhost` refers to the container itself, not the server that is running the container. Use the private IP address or FQDN of the server instead.

Connect version

Faspex 5 requires users run Connect version 4.1.3 or later and does not currently support locally hosting the Connect SDK.

Do not upgrade if your users rely on downloading Connect from the Faspex server (instead of the Cloudfront CDN). Hosting the Connect SDK and installers locally is expected in a future version of Faspex 5.

If your users cannot upgrade to Connect 4.1.3 and later, you can default users to transfer with HTTP Gateway.

Email addresses

Faspex 5 requires new users to have a unique email address. To log in to new Faspex 5 accounts, you must log in using the account email address. You can still log in using usernames for accounts created before the upgrade.

On upgrade, Faspex automatically reconciles external users that have the same email address as a regular user or a SAML user. While new Faspex users must log in using their email address, users created before the upgrade can still log in to those accounts with their usernames (but not with their email addresses).

After upgrading, you can see a list of users with duplicate email addresses by logging in to the Faspex Utility application. Then, from the Admin application, you can manually decide to keep, edit, or remove accounts with duplicate email addresses (see "Review accounts with duplicate email addresses" on page 109).

New users must change their password on first login

New Faspex 5 users must change their password on first login. You can no longer disable this requirement in server settings.

SAML

The SAML metadata and callback URL routes are different from previous versions. After upgrading, retrieve the new metadata and callback URLs from Faspex and update your SAML Identity Provider (see "Reconfigure SAML after upgrading to Faspex 5" on page 24).

Customization

Faspex 5 no longer supports custom CSS, custom HTML or custom Ruby. Use the branding options available in **Configurations > Display settings** (see "Configure display settings" on page 94).

Faspex 5 no longer supports the `faspex.yml` configuration file. If you rely on options in `faspex.yml` that cannot be set in another way in Faspex 5, then do not upgrade. For a list of supported and unsupported options, see "faspex.yml options in Faspex 5" on page 23.

Faspex 5 does not currently support out-of-transfer file validation or post-processing scripts. Starting in 5.0.4, Faspex supports package processing webhooks to notify external systems (through a POST request) that a package is ready for download.

Integration

Faspex 5 no longer provides rake tasks for automating tasks. Instead, use the Faspex 5 API to perform automation.

The Faspex 5 API is not backwards-compatible with prior versions of the APIs.

Public submission links

Public submission links and external user invitations created in Faspex 4 do not work in Faspex 5. Admins can resend the invitation as long as the invitation is not expired

Note: For a full list of differences, see “Differences and breaking changes between Faspex 5 and previous versions” on page 4.

If you have questions or issues with the upgrading process from Faspex 4 to Faspex 5, after reviewing the breaking changes and upgrade considerations, contact IBM support.

Perform the following steps if you have already set up a blank Faspex 5 environment before upgrading from Faspex 4:

- Drop the Faspex table before properly upgrading from Faspex 4.
- Revert the `service.env` file to its original state:

  ```
  cp /opt/aspera/faspex/conf/docker/original/service.env /opt/aspera/faspex/conf/docker
  ```

Upgrade steps

Warning:

Follow these steps before performing any upgrade:

1. Perform a full environment back up and ensure the back up is successful. In case the upgrade fails, the only reliable, short-term fix is to roll back the environment using the back up.
2. Test the upgrade in a test environment comparable to the production environment.
3. If upgrading the test environment is successful, upgrade the production environment, but do not bring the production environment back online.
4. Prior to bringing the production environment back online, the customer must test the application to determine if an immediate rollback is needed. Otherwise, customers risk losing all data generated between upgrade and rollback.

Note: Refer to the Enabling TLS to connect to the database section if you need to connect to the database using TLS.

1. Back up your Faspex 4.X database.

   ```
   asctl faspex:backup_database
   ```

2. Backup your `/opt/aspera/faspex` folder entirely.
3. For each HSTS node, upgrade the IBM Aspera High-Speed Transfer Server version to 4.3 and enable activity logging. You must do this before upgrading Faspex to retain package information. For instructions, see “Enable activity logging on a HSTS node” on page 23.

   If your Faspex 4 server is collocated with a HSTS node, you will need to add it again post-upgrade.

   If you are setting the token encryption key for users in your `aspera.conf` file, use dynamically generated token encryption keys instead. For more information, see the *IBM Aspera High-Speed Transfer Server:Secrets Management with askmscli* section.
4. If you have a co-located HSTS node

   Note: You should change the node host pre-upgrade, but it's possible to do this after upgrading as well.
5. Install these Podman components: `podman-docker`, `podman-plugins`, and `skopeo`.

   ```
   yum install podman-docker podman-plugins skopeo
   ```

6. Download the newest installer and install the RPM file:

   ```
   rpm -Uvh ibm-aspera-faspex-version.build.x86_64.rpm
   ```

7. Configure the generated database environment variables configured by the `.env` files in `/opt/aspera/faspex/conf/docker`:

   `db.env`

Variable	Description	Example
FASPEX_DB_HOST	Remote database hostname	mariadb-remote.example.com
FASPEX_DB_PORT	Remote database port	3306
FASPEX_DB_USERNAME	Restricted user name	faspex
FASPEX_DB_PASSWORD	Restricted user password	0bcecff1-e7df-4596-94f2-1a4b241be252

   `db_admin.env`

Variable	Description	Example
FASPEX_DB_ASPERA_ADMIN_USERNAME	Admin user name	aspera
FASPEX_DB_ASPERA_ADMIN_PASSWORD	Admin user password	ed953178-8d04-4101-96f7-77f61ca9ac0a

   Note: You can leave the `FASPEX_DB_ROOT_USERNAME` and `FASPEX_DB_ROOT_PASSWORD` variables unset, as a remote database set up does not require access to the root user.
8. Run `faspexctl setup`.

   Agree to upgrade from Faspex 4.X and agree to clean the filesystem. Make sure you respond no (n) when the installer prompts you to install the db container.
9. Confirm all containers are present and running:

   ```
   faspexctl status
   ```
10. Confirm the presence of legacy files and old database files:

    Legacy files are stored at `/opt/aspera/faspex_legacy`.

    Old database files are stored at `/opt/aspera/faspex/backup/`.
11. Restore your Faspex 4 database backup using the Faspex Utility application:
    a) Log in to Faspex Utility.
    b) Go to **Manage database**.

       ```
       faspexctl start utility
       ```
    c) In the **Backup or restore the database** area, select **Restore** and select the database file and click **Restore**.
12. Migrate the database:

    ```
    faspexctl setup
    ```
13. Confirm all containers are present and running:

    ```
    faspexctl status
    ```
14. Confirm the upgrade persisted the data from your database. Run:

    ```
    docker exec -it faspex-db mysql faspex -e "show tables"
    ```

    You should be able to access all data from before the upgrade.

15. In order to migrate your SSL certificates installed, you must copy them from the `faspex_legacy` folder to the Nginx configuration folder:

    Note: If you are using the self-signed certificate from Faspex 4, you should NOT replace the certificates that Faspex 5 created.

    a) Rename and move your certificate file. Replace *server_cert_file* with either `server-ca.crt` or `server.crt` as needed:

    ```
    cp -iv /opt/aspera/faspex_legacy/conf/server_cert_file /opt/aspera/faspex/conf/nginx/
    cert.pem
    ```

    b) Rename and move your key file (`server.key`):

    ```
    cp -iv /opt/aspera/faspex_legacy/conf/server.key /opt/aspera/faspex/conf/nginx/key.pem
    ```

    c) Restart `faspex-router`:

    ```
    faspexctl restart router
    ```

16. Update your minimum version of Connect to 4.1.3 or later. Log in to the Faspex UI, go to **Transfer options** and update **Minimum Connect version**.

Enable activity logging on a HSTS node

Faspex 5 uses the activity logging feature of available on IBM Aspera High-Speed Transfer Server version 4.3 or later to retrieve package information. Upgrade and enable activity logging on each of your existing nodes.

1. Enable activity logging for the IBM Aspera High-Speed Transfer Server:

    ```
    asconfigurator -x "set_server_data;activity_logging,true"
    ```

2. Restart `asperanoded`:

    - **Linux:**

        ```
        systemctl restart asperanoded
        ```

    - **Windows:** Go to **Control Panel > Administrative Tools > Computer Management Services and Applications Services**, click **IBM Aspera NodeD**, and click **Restart**.
    - **MacOS:**

        ```
        sudo launchctl stop com.aspera.asperanoded
        sudo launchctl start com.aspera.asperanoded
        ```

3. Verify that the node has enabled activity logging by making **`curl`** call against the HSTS `/ops/transfers` endpoint:

    ```
    curl -I -X GET -u node_api_username:node_api_password https://
    node_hostname:node_api_port/ops/transfers
    ```

    A `200 OK` response confirms the node is correctly configured.

faspex.yml options in Faspex 5

Faspex 5 does not use a `faspex.yml` configuration file. If you relied on a setting configured by that file, check the Faspex 5 coverage.

For detailed information on each setting, go to the IBM Aspera Faspex 4.4 Linux and Windows Admin Guides.

Migrated settings

These parameters are in the web user interface rather than the `faspex.yml` file. Click the parameter for details of the user interface implementation.

- `RequireExternalRecipientsToRegister`
- `ForcePasswordResetForNewUsers` (enabled by default)
- `StrongPasswordRegex`
- `StrongPasswordRequirements`
- `PackageUploadTimeout`
- `HideRelayInformation`
- `SaveMetadataInPackage`

Unsupported settings

These settings are not supported in Faspex 5 web UI:

- `AcceptedHosts`
- `LiveUpdateInterval`
- `HideSenderUsernameToExternalRecipients`
- `MaxTagsLength`
- `UserFieldsInTags`
- `ExcludeMetadataFromCookie`

For security reasons, Faspex 5 no longer supports directory service options:

- `CanonicalizeLdapGroupMemberSearch`
- `DsCheckPeriod`
- `DsSyncActiveState`
- `DsSyncPeriod`
- `DsUsernameAttribute`
- `SearchPrimaryDNs`

Faspex 5 requires email address to be unique:

- `EnforceSelfRegisteredUserEmailUniqueness`
- `SelfRegistrationUsesEmailAsLogin`

Faspex 5 uses Nginx to handle certificates:

- `SSLCAFile`

Faspex 5 Nginx will not proxy HTTP Fallback transfers:

- `UseApachePortsForHttpFallback`

Reconfigure SAML after upgrading to Faspex 5

The SAML metadata and callback URL routes are different from previous versions. Retrieve the new metadata and callback URLs from Faspex and update your SAML Identity Provider (IdP).

1. Go to the Admin application.
2. Go to **Authentication > SAML**.
3. Right-click the SAML configuration and click **Metadata**.
4. Copy the metadata in the modal and upload the new metadata to your IdP.

   Alternatively, you can find the metadata URL and callback URL in the modal:

- The metadata URL is the `entityID`.
- The callback URL is the `Location`.

Faspex 5.0.6 upgrade using patch

Important: Do not use `faspexctl pull` when patching container images, use `faspexctl setup` instead.

Upgrading Faspex 5.0.6 to 5.0.6.2

These instructions apply only to upgrade Faspex 5.0.6 to 5.0.6.2.

1. Download, extract, and apply the update following the instructions included in the `.md` file.
2. Run the following command to complete the setup:

```
faspexctl setup
```

Upgrading Faspex 4 or 5.0.x to 5.0.6

These instructions apply only to upgrade Faspex 4 or 5.0.5 to 5.0.6.

1. Install Faspex 5.0.6 rpm.
2. Download, extract, and apply the update following the instructions included in the `.md` file.
3. Run the following command to complete the setup:

```
faspexctl setup
```

High-availability (HA) installation

Faspex can be deployed in a HA environment by setting up multiple Faspex instances to use a shared remote database.

Intended audience of this procedure

High-availability configuration requires that:

- You have background as a network engineer.
- You are knowledgeable about HA environments.
- You are familiar with the requirements of your use case.
- You are familiar with the Faspex and High Speed Transfer Server products.
- You have expertise and experience configuring third-party systems, such as shared storage and load balancers.

How to use this procedure

This document is intended to be used as a basic guideline to inform and facilitate the customer's ability to craft a HA solution to match the requirements of the customer's particular use case. Each use case differs in requirements, including:

- Approved technology and software vendors
- Hardware sizing requirements
- Security requirements
- Estimated traffic load and required bandwidth
- Budget to cover costs

Procedures for setting up third-party systems such as load balancers are outside the scope of this document.

Note: Because of the complexity of any high-availability setup due to differing requirements for each customer, you should engage with IBM professional services to do an HA install or upgrade of IBM Aspera products. This document primarily serves as a reference detailing basic considerations and requirements to operate Faspex in an HA configuration. You can engage professional services by contacting your sales representative.

Covered use case

The use case described in this guide is a Faspex instance cluster sharing a remote database. The case does not cover setting up a load balancer to route traffic between Faspex instances, nor how to deploy the load-balancer and the remote database with High Availability. For information about these third-party system procedures, refer to their corresponding documentation.

Single point of failure

Aspera strongly encourages customers to consider SPOF (single points of failure) in the environment and to recognize the risks of SPOF. Often, these are situations where all nodes are plugged into the same power strip or surge.

Installing and configuring the HA environment

Configure a HA environment running two or more Faspex instances with a shared remote database.

This procedure requires you to have at least three separate servers:

- A server for the remote database
- Two servers running Faspex 5

Using this procedure, you should be able to configure as many Faspex 5 instances as desired. When the HA environment is provisioned correctly, Faspex can continue running without problems as long as:

- One pair of `faspex-core` and `faspex-ui` services on the same server are healthy
- At least one instance of `faspex-service` is healthy
- The remote database is healthy and accessible for all `faspex-core`, `faspex-service` and `faspex-utility` containers. These containers can run on all hosts of the cluster.

This procedure is intended to be used as a basic guideline to inform and facilitate the customer's ability to craft a HA solution to match the requirements of the customer's particular use case.

In a typical HA setup, the load balancer is provisioned with a virtual IP address (VIP) for user access and then manages all the traffic for Faspex. A fully qualified domain name (FQDN) is used to access the Faspex services and points to the VIP configured on the load balancer. The load balancer’s VIP routes the inbound request to the IP addresses of the appropriate Faspex instance.

This procedure configures multiple Faspex 5 instances sharing one remote database, but does not cover setting up a load balancer, which is required for the system to be a true HA environment. A true HA environment would:

- Set up a load balancer to direct traffic to the Faspex 5 UI.
- Set up a load balancer to direct traffic to the Faspex 5 API server.
- Use a MariaDB database cluster for the remote database. Running a single remote database would be a single point of failure.
- Leverage HSTS clusters.

  Note: You should not collocate HSTS with Faspex 5 for performance HA reasons.

These configurations are beyond the scope of this procedure but should work with Faspex 5 with little configuration. Here are a few tips for working with a load balancer:

- Enable sticky sessions (also known as session affinity) on your load balancer. Faspex 5 initially requires session cookies to correctly generate a bearer token. Once the token is generated, session cookies are no longer needed.
- Configure Faspex 5 on every server added to the load balancer before testing if Faspex 5 works.

Set up the remote database:

1. Configure the remote database for use with Faspex 5. On the remote database server, enter the MySQL console:

   ```
   mysql -u root_user -p
   ```

   Faspex 5 uses three database users:

   - A root-level user for creating the database and for provisioning the other two users. Faspex 5 only uses this user when configuring the `faspex-db` container. In this documentation, this user is `root`.
   - An admin user for updating, creating, and dropping tables. This user is restricted to the Faspex 5 Utility application for migrations. In this documentation, this user is `aspera`.
   - A restricted user for performing production read and write operations. In this documentation, this user is `faspex`.

   a) Disable the `ONLY_FULL_GROUP_BY` SQL mode:

   ```
   SET GLOBAL sql_mode=(SELECT REPLACE(@@sql_mode,'ONLY_FULL_GROUP_BY',''));
   ```

   Note: MariaDB on Amazon RDS does not have `ONLY_FULL_GROUP_BY` enabled by default. You can check if your instance has this SQL mode enabled by accessing your remote database through the MySQL console and running:

   ```
   SELECT @@SESSION. sql_mode;
   ```

   If you have `ONLY_FULL_GROUP_BY` enabled, you must use an Amazon DB parameter group to change the mode on your database.

   b) Create an admin user (`aspera`) for updating, creating, and dropping tables and grant the user access to the required tables.

   ```
   ## Create an admin user named 'aspera' with password '<REPLACE_WITH_SECURE_PASSWORD>'
   CREATE USER IF NOT EXISTS 'aspera'@'%' IDENTIFIED WITH mysql_native_password BY
   '<REPLACE_WITH_SECURE_PASSWORD>';
   ALTER USER 'aspera'@'%' IDENTIFIED BY '<REPLACE_WITH_SECURE_PASSWORD>';
   CREATE USER IF NOT EXISTS 'aspera'@'localhost'IDENTIFIED BY
   '<REPLACE_WITH_SECURE_PASSWORD>';

   # Grant the admin user permissions to the mysql and faspex databases
   GRANT SELECT ON `mysql`.* TO 'aspera'@'%';
   GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, INDEX, ALTER, CREATE TEMPORARY
   TABLES, CREATE VIEW, EVENT, TRIGGER, SHOW VIEW, LOCK TABLES, CREATE ROUTINE, ALTER
   ROUTINE, EXECUTE ON `faspex%`.* TO 'aspera'@'%';
   ```

   c) Create a restricted user (`faspex`) for production read and write operations and grant the user access to the required tables.

   ```
   # Create a restricted user named 'faspex' with password
   CREATE USER IF NOT EXISTS 'faspex'@'%' IDENTIFIED WITH mysql_native_password BY
   '<REPLACE_WITH_SECURE_PASSWORD>;
   ALTER USER 'faspex'@'%' IDENTIFIED BY '<REPLACE_WITH_SECURE_PASSWORD>;

   # Grant the restricted user limited permissions to the faspex database
   GRANT SELECT, INSERT, UPDATE, DELETE ON `faspex%`.* TO 'faspex'@'%';
   ```

   d) Create the Faspex database and flush privileges:

   ```
   CREATE DATABASE IF NOT EXISTS faspex;
   FLUSH PRIVILEGES;
   ```

Set up the first Faspex 5 instance to use the remote database.

2. On your Faspex 5 server, run the Faspex 5 installer, but **do not** run **`faspexctl setup`**.

   ```
   rpm -ivh ibm-aspera-faspex-version.build.x86_64.rpm
   ```

3. Front the Faspex 5 instances with a Load-Balancer in an HA configuration. You should have a minimum of 3 Faspex 5 instances. Use `Nginx version 1.25.1`, refer to the official Nginx documentation (Linux packages) for more information.

   This an Nginx version 1.25.1 `nginx.conf` example:

   Note: For other versions of Nginx consult with your system admin. This example is meant to be used only as a guideline.

   ```
   upstream backend {
      hash $request_uri consistent;
      server IP/FQDN:443 weight=4 max_fails=3 fail_timeout=30s;
      server IP/FQDN:443 weight=4 max_fails=3 fail_timeout=30s;
      server IP/FQDN:443 weight=4 max_fails=3 fail_timeout=30s;
   }

   server {
      listen 443 ssl http2;
      server_name <FQDN>;

      ssl_certificate           /tls/certs/cert.pem;
      ssl_certificate_key       /tls/certs/key.pem;

      ssl_session_cache  builtin:1000  shared:SSL:10m;
      ssl_protocols  TLSv1.2 TLSv1.3;
      ssl_ciphers 'ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-
   GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128-GCM-
   SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-
   ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA256:AES256-GCM-SHA384';
      ssl_prefer_server_ciphers on;

      location ~ files/0(/files)?$ {
         deny all;
      }

      location / {
         if ($request_method ~* "(GET|POST)") {
            add_header "Access-Control-Allow-Origin" *;
         }

         if ($request_method = OPTIONS) {
            add_header "Access-Control-Allow-Origin" *;
            add_header "Access-Control-Allow-Methods" "GET, POST, OPTIONS, HEAD";
            add_header "Access-Control-Allow-Headers" "Authorization, Origin, X-Requested-
   With, Content-Type, Accept";
            return 200;
         }

         proxy_set_header        Host $http_host;
         proxy_set_header        X-Real-IP $remote_addr;
         proxy_set_header        X-Forwarded-For $proxy_add_x_forwarded_for;
         proxy_set_header        X-Forwarded-Proto $scheme;
         proxy_set_header        Origin $scheme://$http_host;

         proxy_pass          https://backend;
         proxy_read_timeout  90;
         proxy_redirect      https://backend https://<IP/FQDN>;
      }
   }
   ```

4. Configure the generated database environment variables configured by the `.env` files in `/opt/aspera/faspex/conf/docker`:

   `db.env`

Variable	Description	Example
FASPEX_DB_HOST	Remote database hostname	mariadb-remote.example.com
FASPEX_DB_PORT	Remote database port	3306

Variable	Description	Example
FASPEX_DB_USERNAME	Restricted user name	faspex
FASPEX_DB_PASSWORD	Restricted user password	0bcecff1-e7df-4596-94f2-1a4b241be252

db_admin.env

Variable	Description	Example
FASPEX_DB_ASPERA_ADMIN_USERNAME	Admin user name	aspera
FASPEX_DB_ASPERA_ADMIN_PASSWORD	Admin user password	ed953178-8d04-4101-96f7-77f61ca9ac0a

Note: You can leave the FASPEX_DB_ROOT_USERNAME and FASPEX_DB_ROOT_PASSWORD variables unset, as a remote database set up does not require access to the root user.

5. For the initial Faspex 5 HA setup, run the following on each Faspex5 instance:

   Important: All the *.env files in /opt/aspera/faspex/conf/docker need to be identical on each Faspex 5 instance. You should first run faspexctl setup on one node and make sure the installation is complete. Confirm that you can access the environment either directly or via the Load-Balancer. Once confirmed, copy the *.env values to the remaining Faspex 5 instances within the HA cluster and repeat the setup up process.

   Note: Make sure to respond no (n) when the installer prompts you to install the db container when running a remote database.

```
# EXAMPLE IF USING REMOTE-DB

faspexctl setup

Faspex
    Install the faspex-db container? [y/n] (default: y): n
    Install the faspex-utility container? [y/n] (default: y):
    Install the faspex-core container? [y/n] (default: y):
    Install the faspex-service container? [y/n] (default: y):
    Install the faspex-ui container? [y/n] (default: y):
    Install the faspex-router container? [y/n] (default: y):
    Install the connect-deployer container? [y/n] (default: y):

Network
    Enter the complete URL for this IBM Aspera Faspex instance. Example: https://
faspex.mydomain.com:8443: LOAD_BALANCER_IP_OR_FQDN

    WARNING: Unable to connect to specified address: LOAD_BALANCER_IP_OR_FQDN


============= Faspexctl Setup Options =============
Faspex
  Install faspex-db?:        NO
  Install faspex-utility?:   YES
  Install faspex-core?:      YES
  Install faspex-service?:   YES
  Install faspex-ui?:        YES
  Install faspex-router?:    YES
  Install connect-deployer?: YES
Network
  Public URL:                LOAD_BALANCER_IP_OR_FQDN (WARNING: Could not connect to
specified address)


  Are these settings correct? [y/n]: y
```

6. Log into your Faspex UI at https://LOAD_BALANCER_IP_OR_FQDN/aspera/faspex.
7. Add a node:

For instruction on setting up a *node*, see “Adding a node to Faspex” on page 76.

a) In the Admin app, go to **Nodes and Storage**.

b) Click **Create node**.

c) Enter a unique name to identify the node.

d) To encrypt the connection to the node using SSL, enable **Use SSL**.

e) To verify the SSL certificate, enable **Verify SSL Certificate**.

f) Provide Faspex the information needed to connect to the Node API on the transfer server:

Field	Description
Host	The node's hostname or IP address. To avoid connectivity problems, do not specify a hostname that contains underscores.
Port	The Node API port number. By default, the port is 9092.
Username	The Node API username on the node machine.
Password	The Node API password on the node machine.

g) Click **Create**.

8. Add a default storage location to Faspex.

a) Select the node you just created.

b) Go to the **Storage locations** tab.

c) Click **Create storage location**.

d) Enter a name for the file storage.

e) Click **Create**.

f) Right-click the storage location you just created and select **Make default inbox** from the overflow menu.

9. Connect an email notification server for notifications.

Faspex sends a welcome email to a new user's email account when a new user is created. New users cannot set their password and log in without the welcome email.

10. Send a file to yourself to test that Faspex is working:

a) Switch to the **Packages** application.

b) Click **Send files**.

c) Add yourself to the **To** field.

d) Enter a package title.

e) Add a test file to the package by clicking the **Add files +** button, selecting **Files** from the drop-down, and choosing a file.

f) Click **Send**.

g) Go to **Received** and find the package you sent yourself.

h) Right-click and select **Download** from the drop-down menu.

Now that the first Faspex instance is running with the remote database, configure each additional Faspex instance:

11. **Note:** For Podman and operating system compatibility, see the Faspex 5 Release notes.

On your Faspex 5 server, install these Podman components: `podman-docker`, `podman-plugins`, and `skopeo`.

```
yum install podman-docker podman-plugins skopeo
```

12. Again, on your Faspex 5 server, run the Faspex 5 installer, but **do not** run **`faspexctl setup`**.

```
rpm -ivh ibm-aspera-faspex-version.build.x86_64.rpm
```

13. Configure the generated database environment variables configured by the `.env` files in `/opt/aspera/faspex/conf/docker`:

    Make sure all the `db.env`, `db_admin.env`, `service.env` files contain the same values as the files on the first Faspex 5 instance.

    The `core.env` file also needs to match the files on the first Faspex 5 instance except for the `FASPEX_CORE_URL`, which must be unique for each Faspex 5 instance:

    ```
    ######################################################################
    # Variables for faspex-core container (on top of variables for Faspex-DB)
    ######################################################################
    # Port of the api rails app
    FASPEX_CORE_PORT=3000

    # Encryption secrets
    FASPEX_CORE_SECRET=same_value_as_first_faspex_instance
    FASPEX_CORE_OLD_SECRET=same_value_as_first_faspex_instance

    # Login values
    FASPEX_CORE_URL=unique_hostname_for_this_faspex_instance
    FASPEX_CORE_UI_CLIENT_ID=same_value_as_first_faspex_instance
    ```

14. Run `faspexctl setup` on the Faspex 5 instance and make sure to say no `(n)` to installing the `faspex-db` container:

Upgrading a Faspex 5 HA environment

To upgrade an existing Faspex 5 HA deployment, you must upgrade each Faspex instance one at a time.

1. Stop all the containers for every Faspex 5 instance except for one.

   On each server, run:

   ```
   faspexctl stop
   ```

Starting with the single running server, update Faspex 5 to the latest version:

2. Fully back up your Faspex instance.
3. **Note:** For Podman and operating system compatibility, see the Faspex 5 Release notes.

   Install these Podman components: `podman-docker`, `podman-plugins`, and `skopeo`.

   ```
   yum install podman-docker podman-plugins skopeo
   ```

4. Download the newest installer and install the RPM file:

   ```
   rpm -Uvh ibm-aspera-faspex-version.build.x86_64.rpm
   ```

5. Update Faspex containers and make sure to say no `(n)` to installing the `faspex-db` container:

   ```
   faspexctl setup
   ```

6. Confirm all containers are present and running:

   ```
   faspexctl status
   ```

7. Open Faspex Utility and login.
8. Go to **Manage database** and make sure that Faspex is using the latest schema. If there is a new migration, migrate the database.

On the other stopped servers, update Faspex 5 to the latest version:

9. Download the newest installer and install the RPM file:

   ```
   rpm -Uvh ibm-aspera-faspex-version.build.x86_64.rpm
   ```

10. Update Faspex containers:

```
faspexctl setup
```

Note: If you are using a remote database, make sure to say no (n) to installing the faspex_db container.

11. Confirm all containers are present and running:

```
faspexctl status
```

Note: Make sure that all your environment files (db.env, db_admin.env, core.env, and service.env) files are in sync between Faspex instances, with the exception of the FASPEX_CORE_URL variable which must be unique. Faspex requires the values of FASPEX_CORE_SECRET, FASPEX_CORE_OLD_SECRET, FASPEX_CORE_UI_CLIENT_ID, and FASPEX_SERVICE_CLIENT_ID to match the values set in the database.

Upgrading a Faspex 4.X high-availability configuration

Upgrade from a version before Faspex 5.0 and confirm the persistence of files and data.

Warning: Do not continue with this upgrade until you have reviewed all the breaking changes and upgrade considerations

Review breaking changes and upgrade considerations

Operating system

Faspex 5 does not support Windows as an operating system. If you're currently using Windows, in order to upgrade to Faspex 5, migrate your Faspex 4.X instance to a Linux server and test before upgrading to Faspex 5 on Linux.

Your server must run CentOS7, RHEL 7, RHEL 8, or (starting in 5.0.4) Amazon Linux 2.

Your server must have Docker or (starting in 5.0.4) Podman 4.1 that is installed. For tips on installing Docker and Podman, see #unique_27.

Note: For Podman and operating system compatibility, see the Faspex 5 Release notes.

Your server must also have network access to the IBM Cloud Container Registry (icr.io). The installer pulls container images from icr.io.

Database

If you are using the default Faspex 4 database, the upgrade process migrates your data to the faspex-db container. If you have an external database running, you will have to back up your Faspex 4 external database and run migration against the external database.

Note: The following error is displayed during the upgrade: "Column count of mysql.proc is wrong. Expected 21, found 20. Created with MariaDB 50737, now running 100803. Please use mariadb-upgrade to fix this error" This is expected to happen and the installer will continue without any issues.

Directory services

For security reasons, Faspex 5 does not support directory services. You must instead front your directory service with a SAML Identity Provider (IdP) and use SAML based authentication for your users. Do not upgrade if you rely on a direct connection between Faspex and directory services. If you have Directory Services users, you will need to migrate those users to SAML before upgrading from Faspex 4 to Faspex 5.

High-Speed Transfer Server version requirement

Faspex 5 uses a more modern API for interacting with IBM Aspera High-Speed Transfer Server (HSTS). Faspex 5 uses the activity logging feature available on HSTS 4.3 or later. You must upgrade HSTS on your nodes and enable activity logging before upgrading (see "Enable activity logging on a HSTS node" on page 23).

For setups where collocation of Faspex 5 with HSTS is unavoidable, the IP address included in the node configuration in Faspex 5 can no longer be `127.0.0.1` or `localhost`. This is because Faspex 5 is containerized and `localhost` refers to the container itself, not the server that is running the container. Use the private IP address or FQDN of the server instead.

Connect version

Faspex 5 requires users run Connect version 4.1.3 or later and does not currently support locally hosting the Connect SDK.

Do not upgrade if your users rely on downloading Connect from the Faspex server (instead of the Cloudfront CDN). Hosting the Connect SDK and installers locally is expected in a future version of Faspex 5.

If your users cannot upgrade to Connect 4.1.3 and later, you can default users to transfer with HTTP Gateway.

Email addresses

Faspex 5 requires new users to have a unique email address. To log in to new Faspex 5 accounts, you must log in using the account email address. You can still log in using usernames for accounts created before the upgrade.

On upgrade, Faspex automatically reconciles external users that have the same email address as a regular user or a SAML user. While new Faspex users must log in using their email address, users created before the upgrade can still log in to those accounts with their usernames (but not with their email addresses).

After upgrading, you can see a list of users with duplicate email addresses by logging in to the Faspex Utility application. Then, from the Admin application, you can manually decide to keep, edit, or remove accounts with duplicate email addresses (see "Review accounts with duplicate email addresses" on page 109).

New users must change their password on first login

New Faspex 5 users must change their password on first login. You can no longer disable this requirement in server settings.

SAML

The SAML metadata and callback URL routes are different from previous versions. After upgrading, retrieve the new metadata and callback URLs from Faspex and update your SAML Identity Provider (see "Reconfigure SAML after upgrading to Faspex 5" on page 24).

Customization

Faspex 5 no longer supports custom CSS, custom HTML or custom Ruby. Use the branding options available in **Configurations > Display settings** (see "Configure display settings" on page 94).

Faspex 5 no longer supports the `faspex.yml` configuration file. If you rely on options in `faspex.yml` that cannot be set in another way in Faspex 5, then do not upgrade. For a list of supported and unsupported options, see "faspex.yml options in Faspex 5" on page 23.

Faspex 5 does not currently support out-of-transfer file validation or post-processing scripts. Starting in 5.0.4, Faspex supports package processing webhooks to notify external systems (through a POST request) that a package is ready for download.

Integration

Faspex 5 no longer provides rake tasks for automating tasks. Instead, use the Faspex 5 API to perform automation.

The Faspex 5 API is not backwards-compatible with prior versions of the APIs.

Public submission links

Public submission links and external user invitations created in Faspex 4 do not work in Faspex 5. Admins can resend the invitation as long as the invitation is not expired

Note: For a full list of differences, see "Differences and breaking changes between Faspex 5 and previous versions" on page 4.

If you have questions or issues with the upgrading process from Faspex 4 to Faspex 5, after reviewing the breaking changes and upgrade considerations, contact IBM support.

Differences between Faspex 4.X HA and Faspex 5 HA

- Faspex 5 uses both MariaDB and MySQL 8.
- Faspex 5 uses a remote database cluster instead of typically collocated database cluster.
- Aspera Cluster Manager (ACM) is no longer necessary and not supported
- Faspex 5 HA is a fully active/active setup. All containers except for `faspex_db` can be run on all Faspex 5 HA instances.

Upgrade steps

When you have fully reviewed the breaking changes and upgrade considerations, follow these steps to upgrade:

1. Stop Faspex and MySQL services before performing the upgrade.

   a) On the active node, back up the database:

   ```
   asctl faspex:backup_database
   ```

   b) Stop all Faspex services on the active node.

   ```
   asctl all:stop
   ```

   c) Start MySQL service on the passive node and back up the database.

   ```
   asctl mysql:start
   asctl faspex:backup_database
   ```

   d) Stop MySQL and ensure all Faspex services are stopped on the passive node.

   ```
   asctl mysql:stop
   asctl all:stop
   ```

2. If needed, upgrade or migrate your remote MySQL database to run MariaDB 10.6.7 or later.
3. For each HSTS node, upgrade the IBM Aspera High-Speed Transfer Server version to 4.3 and enable activity logging. You must do this before upgrading Faspex to retain package information. For instructions, see "Enable activity logging on a HSTS node" on page 23.

Set up one of the Faspex 5 instances to use the remote database.

4. On your Faspex 5 server, install these Podman components: `podman-docker`, `podman-plugins`, and `skopeo`.

   ```
   yum install podman-docker podman-plugins skopeo
   ```

5. On your Faspex 5 server, run the Faspex 5 installer to upgrade, but **do not** run **`faspexctl setup`**.

   ```
   rpm -Uvh ibm-aspera-faspex-version.build.x86_64.rpm
   ```

6. Configure the generated database environment variables configured by the `.env` files in `/opt/aspera/faspex/conf/docker`:

   `db.env`

Variable	Description	Example
FASPEX_DB_HOST	Remote database hostname	`mariadb-remote.example.com`

Variable	Description	Example
FASPEX_DB_PORT	Remote database port	3306
FASPEX_DB_USERNAME	Restricted user name	faspex
FASPEX_DB_PASSWORD	Restricted user password	0bcecff1-e7df-4596-94f2-1a4b241be252

db_admin.env

Variable	Description	Example
FASPEX_DB_ASPERA_ADMIN_USERNAME	Admin user name	aspera
FASPEX_DB_ASPERA_ADMIN_PASSWORD	Admin user password	ed953178-8d04-4101-96f7-77f61ca9ac0a

Note: You can leave the `FASPEX_DB_ROOT_USERNAME` and `FASPEX_DB_ROOT_PASSWORD` variables unset, as a remote database set up does not require access to the root user.

7. Run `faspexctl setup`.

   Agree to upgrade from Faspex 4.X and agree to clean the filesystem. Make sure you respond no (n) when the installer prompts you to install the db container.

8. Confirm all containers are present and running:

   ```
   faspexctl status
   ```

9. Confirm the presence of legacy files and old database files:

   Legacy files are stored at `/opt/aspera/faspex_legacy`.

   Old database files are stored at `/opt/aspera/faspex/backup/`.

10. Confirm the upgrade persisted the data from your database. Run:

    ```
    docker exec -it faspex-db mysql faspex -e "show tables"
    ```

    You should be able to access all data from before the upgrade.

Now that the first Faspex instance is running with the remote database, configure each additional Faspex instance:

11. Again, on your Faspex 5 server, run the Faspex 5 installer, but **do not** run **`faspexctl setup`**.

    ```
    rpm -ivh ibm-aspera-faspex-version.build.x86_64.rpm
    ```

12. Configure the generated database environment variables configured by the `.env` files in `/opt/aspera/faspex/conf/docker`:

    Make sure all the `db.env`, `db_admin.env`, `service.env` files contain the same values as the files on the first Faspex 5 instance.

    The `core.env` file also needs to match the files on the first Faspex 5 instance except for the `FASPEX_CORE_URL`, which must be unique for each Faspex 5 instance:

    ```
    ######################################################################
    # Variables for faspex-core container (on top of variables for Faspex-DB)
    ######################################################################
    # Port of the api rails app
    FASPEX_CORE_PORT=3000

    # Encryption secrets
    FASPEX_CORE_SECRET=same_value_as_first_faspex_instance
    FASPEX_CORE_OLD_SECRET=same_value_as_first_faspex_instance

    # Login values
    ```

```
FASPEX_CORE_URL=unique_hostname_for_this_faspex_instance
FASPEX_CORE_UI_CLIENT_ID=same_value_as_first_faspex_instance
```

13. Run `faspexctl setup` on the Faspex 5 instance. Agree to upgrade from Faspex 4.X **but do not** agree to clean the filesystem. Make sure you respond no (n) when the installer prompts you to install the db container.
14. Update your minimum version of Connect to 4.1.3 or later. Log in to the Faspex UI, go to Transfer options and update Minimum Connect version.

Running Faspex 5 as non-root on systems with Podman runtime

Run Faspex 5 as a non-root user for enhanced security and resource efficiency.

Attention: Steps one through seven must be performed as the root user.

1. Install Faspex 5 following the instructions in the "Installation and upgrades" on page 7 section.
2. Create a non-root user to run Faspex 5, for example `newuser`:

```
useradd -m newuser
```

3. Change ownership to the newly created non-root user:

```
chown -R newuser:newuser /opt/aspera/faspex
```

4. Disable Podman in the context of the `root` user:

```
systemctl disable --now podman podman.socket
```

5. Set a password for the non-root user for SSH access:

```
passwd newuser
```

6. Enable linger for the non-root user. This will make Faspex 5 startup after rebooting:

```
loginctl enable-linger newuser
```

7. Give the non-root user sudo permissions, add the following to the `visudo` file:

```
%newuser ALL= NOPASSWD: /bin/systemctl enable --now --user podman podman.socket
%newuser ALL= NOPASSWD: /bin/systemctl disable podman.socket
%newuser ALL= NOPASSWD: /bin/systemctl restart podman.socket

%newuser ALL= NOPASSWD: /bin/systemctl enable faspexctl.service
%newuser ALL= NOPASSWD: /bin/systemctl start faspexctl.service
%newuser ALL= NOPASSWD: /bin/systemctl restart faspexctl.service
```

Attention: Steps eight through 14 must be performed by the new non-root user. You must connect through an SSH session for security reasons, rather than just switching to a non-root user using the `su` command.

8. SSH as the new non-root user:

```
ssh newuser@FQDN/IP_ADDR
```

9. Enable Podman as the new non-root user:

```
systemctl enable --now --user podman podman.socket
```

10. Add the `UID` of the non-root user to the `DOCKER_HOST` unix socket.

```
echo "export DOCKER_HOST=unix:///run/user/$UID/podman/podman.sock" >> ~/.bashrc
```

11. Reload the `bashrc` file:

```
. ~/.bashrc
```

12. Edit the /opt/aspera/faspex/conf/docker/router.env file. The http and https ports must be higher than 1024 (low number ports are reserved for root users). For example, you could use 8085 and 7443. Make sure the ports you choose are not already bound to some other service.
13. Run faspexctl setup. When you receive the ip prompt enter the IP address or FQDN.

    ```
    $IP_ADDR or $FQDN:$CUSTOM_HTTPS_PORT
    ```

14. Confirm that services are run as the new non-root user:

    ```
    ps aux | grep aspera
    ```

Patching container images

It is possible to upgrade Faspex 5 container images to the latest version within the current tag. You can do this when one or more new container images have been pushed to the same tag, for example in the case of bug and security fixes. You don't need to upgrade the Faspex 5 RPM to complete this procedure.

1. Run the following command **as root:**

   ```
   faspexctl setup
   ```

   This will check the IBM Container Registry (ICR) for more recent images associated with the current tag of your Faspex 5 environment .
2. Verify the container images that are currently installed:

   ```
   faspexctl version
   ```

   Example output:

```
    faspex-ui:                    5.0.6 (9119b88)
    faspex-core:                  5.0.6 (5c88fd5)
    faspex-utility:               5.0.6 (de9d2e9)
    faspex-service:               5.0.6 (46e4f88)
    faspex-db:                    5.0.6 (4466999)
    faspex-router:                5.0.6 (5ac85c8)
    connect-deployer:             4.2.5 (b4c2ea7)

    Software Versions:
    MariaDB (in faspex-db):       10.9.3
    Nginx   (in faspex-router):   1.23.4
```

Configure Nginx settings

Faspex 5 uses Nginx as a reverse proxy to route traffic to the appropriate container. You can customize Nginx configuration files and update your SSL certificate and key using the files at /opt/aspera/faspex/conf/nginx/conf.

Nginx configuration files

To customize Nginx configuration, update the files in /opt/aspera/faspex/conf/nginx/conf/custom.

To update your SSL certificate and key, or to replace the self-signed default certificate, replace cert.pem, dhparam.pem, and key.pem, found at /opt/aspera/faspex/conf/nginx/conf.

After editing or udpating any of these files, restart the Faspex router service by running:

```
faspexctl restart router
```

The file nginx.conf.copy contains the settings currently running. This file updates each time you restart the router using the command above.

The files in the directory `/opt/aspera/faspex/conf/nginx/conf/custom/orig` function as a backup of all the `/opt/aspera/faspex/conf/nginx/conf/custom/conf` files prior to restarting the router. You can use the contents of the `/orig` directory for reference and restoration as required.

Tip: Run `faspexctl version` to check the version of Nginx running in the `faspex-router` container.

Limiting Egress

Limiting egress allows administrators to control the amount of data leaving their network. This feature helps organizations to regulate the flow of outbound data and ensure that sensitive information is not being shared outside their network without proper authorization.

To prevent egress to an unauthorized IP address, run the following command replacing `[IP]` with the unauthorized IP address:

```
iptables -A INPUT -s [IP] -j DROP
```

For a list of unapproved IP addresses of botnets and command/control servers, visit IBM X-Force Exchange. You should run the block command on each of the IP addresses listed on the IBM X-Force Exchange page.

Logging in to Faspex

Log in to Faspex with your email address and password.

Logging in with SAML

Instead of logging in with a Faspex username and password, use a SAML identity provider (IdP) to log in to Faspex.

You can log in using a SAML IdP in one of three ways:

- Select an enabled SAML IdP from the login page.
- If Faspex has been configured with a default SAML IdP for authentication, Faspex automatically redirects you to the SAML login page of the default SAML IdP.

To bypass a default SAML IdP, see "Bypassing SAML redirection" on page 38.

Bypassing SAML redirection

If configured, Faspex automatically redirects you to the SAML login page of the default SAML IdP. You can bypass the SAML IdP to authenticate as a local user or with a different SAML IdP.

Logging in from the local login page

To go to the local login page, add `?local=true` to the end of the Faspex URL. For example: `https://faspex.example.com/aspera/faspex?local=true`.

From the local login page, you can log in as a local user or log in by choosing a SAML IdP from the list of alternate logins.

Creating or requesting an account

If enabled by an admin, you can register for an account on the login page.

1. On the login page, click either **Create an account** or **Request an account**, depending on which one is available.

   If you can create an account without needing approval, the login page shows **Create an account**.

If you need an admin or manager to approve your account, the login page shows **Request an account**.

2. Complete the account registration form and submit the form.
3. Check your email for an account confirmation.

   If you are requesting an account, you will not receive an email until an admin approves or declines your request.

Issues with logging in

Invalid credentials

If you are having trouble logging in, click the **Forgot your credentials?** link.

Issues logging in with your email address

If Faspex does not allow you to log in with your email address and Faspex displays the message, "Please contact your administrator or try logging in with a username", it could be that your email address is associated with multiple Faspex accounts. Try logging in with your username instead. For you to log in with your email, an admin needs to reconcile the associated accounts.

Regular user

Sending a package

Sending a package is similar to sending an email with attachments: choose recipients, select content, and send. Use the quick starts to send a file as quickly as possible. Use the detailed procedures for all the bells and whistles.

By default, users can send packages to:

- Users in their contacts
- Personal distribution lists
- Shared inboxes
- Workgroups they are a part of (if workgroup settings allow)
- Members of workgroups they are a part of (if workgroup settings allow)
- Global distribution lists (if allowed to see)

Sending a package with uploaded content using Connect

Send a package using Connect.

1. Click **Send files**.
2. Search for and add recipients to the **To** field.

   A recipient can be:

   - A contact, which is a Faspex user
   - An *external user* (if allowed)
   - A shared inbox you have permission to send to
   - A workgroup that you have permission to send to
   - A distribution list

If you cannot find the contact you're looking for, either the contact does not exist or you do not have permissions to send to that type of contact.

3. Enter a package title.
4. Add content to the package. You can add:

   - Files and folders on your computer by dragging and dropping the files.
   - Files on your computer by clicking the **Add files +** button and selecting **Files**.
   - Folders on your computer by clicking the **Add files +** button and selecting **Folders**.

   If you want to send content from shared folders, see "Sending a package with content from a shared folder" on page 41. You cannot include content from both your computer and a shared folder.
5. If required, click **Next** and fill out required metadata fields.
6. Click **Send**.

Additional options are covered in "Additional options for sending a package" on page 43.

Sending a package with uploaded content using HTTP Gateway

Send a package using HTTP Gateway.

To send using HTTP Gateway, open the transfer activity monitor and enable **Force to use Aspera HTTP Gateway** (if HTTP Gateway is available).

1. Click **Send files**.
2. Search for and add recipients to the **To** field.

   A recipient can be:

   - A contact, which is a Faspex user
   - An *external user* (if allowed)
   - A shared inbox you have permission to send to
   - A workgroup that you have permission to send to
   - A distribution list

   If you cannot find the contact you're looking for, either the contact does not exist or you do not have permissions to send to that type of contact.
3. Enter a package title.
4. Add content to the package. You can add:

   - Files on your computer by dragging and dropping the files.
   - Files on your computer by clicking the **Add files +** button and selecting **Files**.

   At this time, HTTP Gateway does not support uploading folders.

   If you want to send content from shared folders, see "Sending a package with content from a shared folder" on page 41. You cannot include content from both your computer and a shared folder.
5. If required, click **Next** and fill out required metadata fields.
6. Click **Send**.

Additional options are covered in "Additional options for sending a package" on page 43.

⚠ **Warning:** Do not close or refresh your page while a transfer using HTTP Gateway is active. Uploads using HTTP Gateway require an open websocket connection. Closing or refreshing the page ends closes the connection and cancels the transfer.

Sending a package with uploaded content to a shared inbox

Send a package to a shared inbox. When sending a new package, you can send the package to only one shared inbox. You cannot send to both a shared inbox and normal recipients.

1. Click **Send files**.
2. Add a shared inbox to the **To** field.

   The first recipient you add must be a shared inbox. If you have other recipients added, you cannot add a shared inbox.
3. Enter a package title.
4. Add content to the package. You can add:

   - Files and folders on your computer by dragging and dropping the files.
   - Files on your computer by clicking the **Add files +** button and selecting **Files**.
   - Folders on your computer by clicking the **Add files +** button and selecting **Folders**.

   If you want to send content from shared folders, see "Sending a package with content from a shared folder" on page 41. You cannot include content from both your computer and a shared folder.
5. If required, click **Next** and fill out required metadata fields.
6. Click **Send**.

Additional options are covered in "Additional options for sending a package" on page 43.

Sending a package with content from a shared folder

If permitted, you can use files and folders in shared folders as the content source of a package.

A shared folder is a shared directory on the HSTS node. It can be used as the destination or source of a package. This was commonly reffered to as a "share" in Faspex 4.

Only registered Faspex users with the **Create packages from remote sources** option enabled can send packages with content in a shared folder. For more information about shared folders

1. Click **Send files**.
2. Search for and add recipients to the **To** field.

   A recipient can be:

   - A contact, which is a Faspex user
   - An *external user* (if allowed)
   - A shared inbox you have permission to send to
   - A workgroup that you have permission to send to
   - A distribution list

   If you cannot find the contact you're looking for, either the contact does not exist or you do not have permissions to send to that type of contact.
3. Enter a package title.
4. Click the **Add files +** button and select **Import from shared folders**.
5. Select the content you want to send and click **Add**.
6. If required, click **Next** and fill out required metadata fields.
7. Click **Send**.

   If sending the package fails, the HSTS node may be misconfigured or down.

Additional options are covered in "Additional options for sending a package" on page 43.

Note: When sending a package with content from a shared folder, you can send a max of 10,000 files when using file-name masking.

Scheduling a package

Schedule a package for Faspex to release at a later date.

1. Click **Send files**.
2. Search for and add recipients to the **To** field.

   A recipient can be:

   - A contact, which is a Faspex user
   - An *external user* (if allowed)
   - A shared inbox you have permission to send to
   - A workgroup that you have permission to send to
   - A distribution list

   If you cannot find the contact you're looking for, either the contact does not exist or you do not have permissions to send to that type of contact.
3. Enter a package title.
4. Add content to the package. You can add:
   - Files and folders on your computer by dragging and dropping the files.
   - Files on your computer by clicking the **Add files +** button and selecting **Files**.
   - Folders on your computer by clicking the **Add files +** button and selecting **Folders**.
   - Files and folders from shared folders you have access to by clicking the **Add files +** button and selecting **Import from shared folders**.
5. Expand the **Release policy** section and select **Release later**.
6. Choose a release date or select **Set release date later**.

   If you choose to **Set release date later**, you can set the release date after creating the package by selecting it in your **Pending** inbox.
7. If required, click **Next** and fill out required metadata fields.
8. Click **Send**.

Scheduling a package initiates a transfer to upload the selected content to the default Faspex node. The content is stored on the node until the release date (or until you choose to release the package).

Additional options are covered in "Additional options for sending a package" on page 43.

Forwarding a package

Forward a package you sent or received to another recipient.

1. Select a package you can access.
2. Click the **Forwad package** button.

   You must be given permissions to forward a package. If you do not see the button, you do not have permission to forward packages.
3. On the forward package page, enter recipients and decide whether to release the package now or schedule it for later.
4. Click **Send**.

Additional options for sending a package

On the send package form, you can configure email notifications, password-protect a package, set expiration policies, and configure transfer security options.

Make a package public

A public package is a package that recipients can download using a link.

If allowed, enable **Recipients with an account can download without logging in** on the send package form to allow recipients that are Faspex users to download the package without logging in.

Note: Making a package public only affects Faspex users. If required by a Faspex admin, recipients without an account are still required to log in before downloading the package.

Configure email notifications

Expand the **Notifications** section:

Option	Description	Email template used
When the package is available, notify	Add contacts you want to notify when the package contents are uploaded successfully. You cannot include workgroups in these fields.	• Upload Result CC
When a recipient downloads the package, notify	Add contacts you want to notify when the package is successfully downloaded by a recipient. You cannot include workgroups in these fields.	• Package Downloaded CC
When a recipient receives the package, notify	**Note:** An administrator must have enabled **User can edit receipt addresses when sending a package** for your account. Add contacts you want copied on each notification you receive when a recipient receives the package. If an admin has added default contacts for your account, this field is auto-populated with those contacts.	• Package Received CC • Package Sent CC • Package Downloaded CC • Upload Result CC

Password-protect a package

Expand the **Password protection** section and enter a password. Faspex uses encryption-at-rest to secure the package.

Set package expiration policies

Set a package expiration in the **Expiration policy** section. When a package expires, Faspex deletes the files in the package. You can expire a package after a set duration or after recipients download files.

Important: When **Delete after any recipient downloads all files is selected**, a forwarded package can be potentially deleted before the original recipient has downloaded it.

For more information about package expiration, see "Configuring package expiration policies" on page 95.

Mask package file names

Mask package file names by expanding the **More transfer options** and enabling **Mask file names**. For more information about masking file names, see "Masking file names in packages" on page 100.

Note:

When sending a package with content from a shared folder, you can send a max of 10,000 files when using file-name masking.

File-name masking is not supported when using HTTP Gateway 2.2.0. Upgrade HTTP Gateway to version 2.3.0 if you need to use file-name masking with HTTP Gateway.

Prevent recipients from downloading the package over HTTP

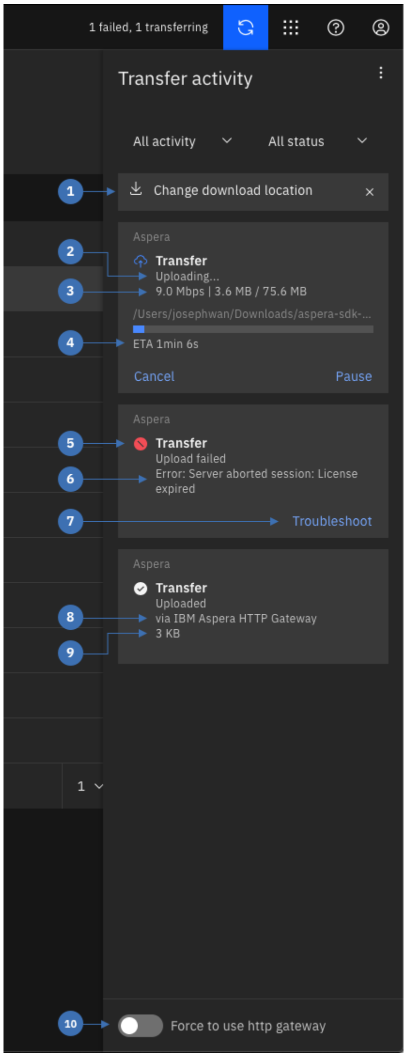
Connect is the fastest and most secure means of file transfer. Enabling this feature requires recipients to download using Connect.

On the send package page, expand the **More transfer options** section and enable **Prevent recipients from downloading the package over less secure HTTP**.

Monitoring transfers

Use the transfer monitor to manage or watch ongoing and recently completed transfers.

Open the transfer monitor by clicking the ∿ icon in the menubar.



Error: Server aborted session: License expired

Callout	Description
1	Link to open the Transfers tab in IBM Aspera Connect preferences where you can change your package download location.

Callout	Description
2	The status of an ongoing transfer.
3	The transfer rate and progress of an ongoing transfer.
4	Estimated time left for the transfer (Connect only).
5	Indication of whether a transfer was successful.
6	Error message if a transfer was unsuccessful.
7	Link to troubleshooting information.
8	Indication of whether the transfer was sent using HTTP Gateway.
9	Total size of the transferred package.
10	Toggle to choose whether to use Connect or HTTP Gateway for transfers. This is only available when an admin has configured Faspex 5 to use Aspera HTTP Gateway.

Viewing and downloading packages as a Faspex user

Mailboxes

These are the available mailbox types:

Mailbox type	Description
Received	Packages that other Faspex users have sent to you. Within the Received category, you have access to multiple mailboxes: • Every user has access to **My packages**. This is a list of packages sent directly to this user. • Every user has access to **All packages**. This is a list of packages sent directly to this user or shared inboxes and workgroups this user belongs to. • You can also see any shared inbox you are a member of.
Sent	Packages that you've sent to other users.
Pending	Packages that you've created but haven't yet sent.

Downloading packages

Important: If you are using HTTP Gateway for downloads, the HTTP Gateway server must be version 2.2 and later to download folders. You can still download individual files.

When viewing any list of packages, you can download individual packages by:

- Right-clicking the package and selecting **Download**.
- Selecting **Download** from the more options (⋮) drop-down menu.
- Selecting multiple packages and clicking **Download** from the menu.

To download individual files in a package, you must view the package details by clicking into a package.

If the package is still transferring, you can download partially completed files but not the entire package.

Note: If you are unable to download a package while it's transferring, an admin might have disabled this option.

When downloading an encrypted package, Connect prompts you for a passphrase. Connect also prompts for a passphrase if the package contains any **.aspera-env** files within the folder hierarchy, even if the package also contains unencrypted files or files encrypted with different passphrases. If you choose to keep downloaded files encrypted, you do not need to enter a password until you attempt to decrypt the files locally.

Hiding packages

You can hide packages by selecting **Hide** from the more options drop-down menu. You cannot undo hiding a package, but you can see hidden packages by clicking the **Full History** tab.

Managing your account

Click the profile icon (👤) in the top-right of the banner and select **Account settings** from the drop-down menu.

Using HTTP Gateway instead of Connect

HTTP Gateway is a standalone product that allows you to leverage IBM Aspera technology to transfer data without needing to install IBM Aspera Connect.

When enabled, Faspex users without Connect can send and download packages using HTTP Gateway. Users with Connect can choose to send using HTTP Gateway instead of Connect by disabling Connect in their account preferences using the **Disable IBM Aspera Connect** toggle.

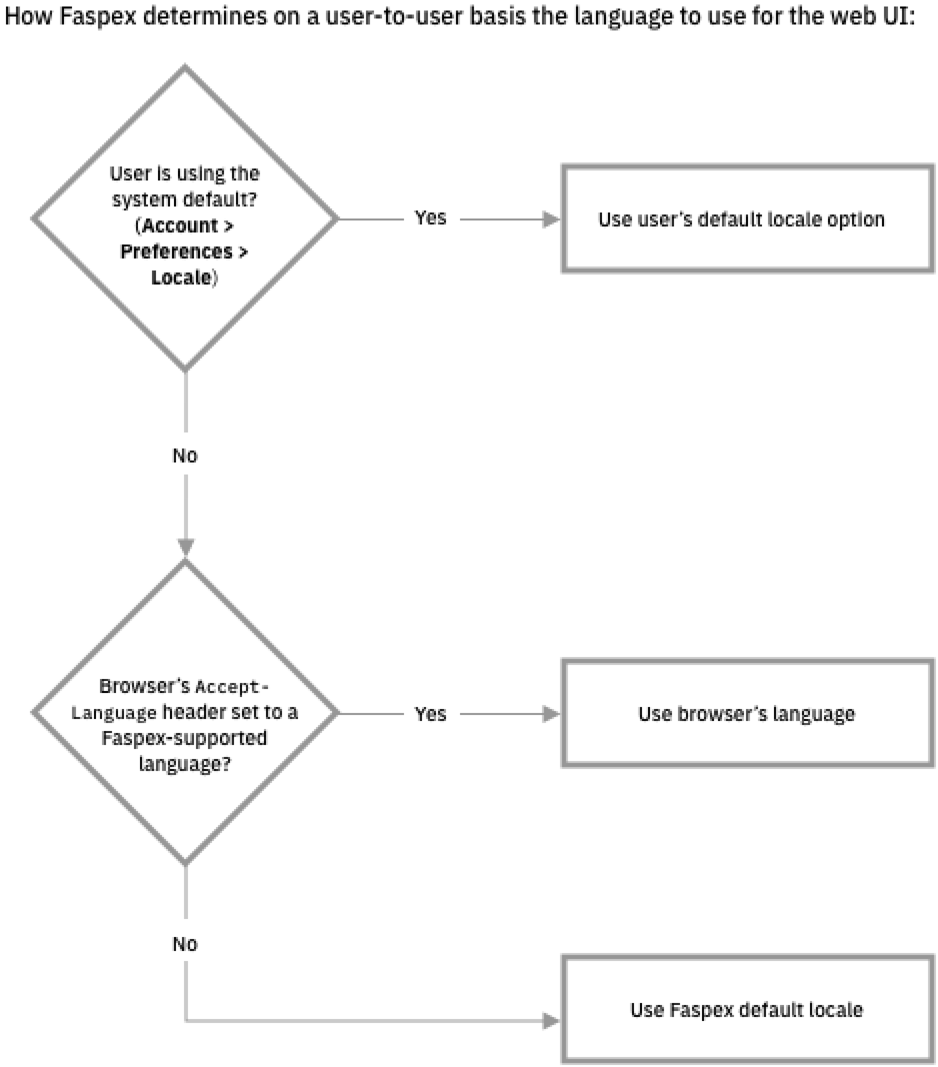
Tip: You can quickly toggle this setting from anywhere by opening the Transfer activity panel and enabling **Force to use Aspera HTTP Gateway**.

To use HTTP Gateway instead of Connect:

1. Click the profile icon (👤) in the top, right-hand corner and select **Account settings** from the drop-down menu.
2. On the **Preferences** tab, toggle **Disable IBM Aspera Connect**.

Changing your language

Change your default language to another supported language.


How Faspex determines on a user-to-user basis the language to use for the web UI:
User is using the system default? (Account > Preferences > Locale)
Yes
Use user's default locale option
No
Browser's Accept-Language header set to a Faspex-supported language?
Yes
Use browser's language
No
Use Faspex default locale

1. Click the profile icon in the top-right of the banner and select **Account** from the drop-down menu.
2. Select your language from the **Locale** drop-down menu.
3. Click **Save**.

Changing your password

Change your account password.

1. Click the profile icon and select **Account settings** from the drop-down menu.
2. Go to the **Preferences** tab and click the **Change password** button.
3. Enter your current password, new password, and new password confirmation.

   By default, the requirement is a strong password that contains at least six characters (with a minimum of one letter, one number and one symbol).
4. Click **Save**.

Setting your public key

You can add your public key to your own user account.

1. Go to **User menu > Account settings > Preferences > Public key**.
2. Add your key in PEM format. Click **Save**.

Managing your contacts list

Whenever you send packages to an email address not associated with a Faspex user, Faspex automatically saves the email to your external contacts list.

To remove contacts from your contacts list:

1. Click the profile icon (👤) in the top-right of the banner and select **Account** from the drop-down menu.
2. Go to the **External contacts** tab.
3. Click the **Delete** icon for each contact you want to remove.

Creating a personal distribution list

Create a personal distribution list to send packages to a list of contacts. Contacts can be Faspex users or external contacts.

By default, you cannot send packages to a distribution list if any recipient in the list is an invalid user. For example, if the option to send to external users is disabled, you cannot send packages to a distribution list that contains an external user.

Note: The items in the list are not validated until you try to send a package to the list.

1. Click the profile icon (👤) and select **Account settings**.
2. Go to **Distribution lists**.
3. Click **Create new**,
4. Name your distribution list
5. Add contacts or import from a CSV file.

   Faspex reads the first line in the CSV file as user fields. The CSV file supports these fields: `email`, `name`.

   For example, to create two users:

   ```
   email, name
   admin@ibm.com, Admin
   user@example.com, User
   ```
6. Click **Create**.

Check data usage and sender quota limit

If sender quotas are enabled, you can check your sender quota limit and the amount of data you've sent in your personal preferences.

Click the profile icon in the banner and select **Account** from the drop-down menu. Go to **Sender Quota**.

The page shows your remaining available data in the current rolling period and the quota limit set for your account. When the remaining available data drops to zero, you cannot send packages until the rolling period expires and your available data is reset.

Invite someone to send you packages

You can invite someone to send you packages by sending an invitation or by sharing with them your public submission link. Unless required by an admin, external users don't have to log in to send a package using an invitation or URL.

Send an email invitation to allow someone to send you packages

Send an email invitation to allow anyone with the invitation to send you packages without logging in.

1. Go to **My packages** and click the **Invitations** tab.

   You can see all your invitations on this page.

   If you do not see the **Invitations** tab, you do not have permission to send invitations. Ask your admin for permission.
2. Click **New invitation**.
3. Enter the invitee's email address.
4. Choose whether to require the sender to use Connect when sending the package.
5. Set a link expiration policy. If you do not enable **Set invitation link expiration policy**, Faspex uses the server default link expiration setting.

   a) Select **Link expires after a collaborator successfully submits a package** to expire invitation links after invitees have successfully sent a package.

   b) Select **Time-based policy** to expire invitation links after a period of time. For example, if you configure this setting for 10 days, Faspex expires the link 10 days after Faspex sends the email invitation to the invitee.

   You can enable both policies. The link expires whenever either of the conditions are met.
6. Click **Send invitation**.

   Faspex sends an invitation link to the email address. Anyone using the link can send a package to you until the link expires.

Share your public submission link

Send an email invitation to allow anyone with the invitation to send you packages without logging in.

You must first enable public submission links on your account:

1. Click the profile icon (👤) in the top-right of the banner and select **Account** from the drop-down menu.
2. Enable **Enable your submission link to allow anyone with the link to submit packages to you**.
3. Click **Save**.

If you do not see the option to enable your public submission link, you do not have permission to share your public submission link. Ask your admin for permission.

⚠ **Warning:** Anyone with your public submission link can send you a package without logging in.

After enabling public submission links:

1. Go to **Received > My packages**.
2. Click the **Invitations** tab.
3. Copy the public submission link to your clipboard.
4. Share the public submission link with anyone you want to send you packages.

Invite external users to send packages to a shared inbox through an email

If you are a member of a shared inbox that allows regular users to invite members or you are a shared inbox admin, you can send an invitation email to allow someone to send a package to a shared inbox without registering an account or logging in.

Anyone with the URL can send packages to the shared inbox, but cannot see or download shared inbox packages.

Faspex cannot verify the person using the link is the intended collaborator. If this is a concern, set a custom link expiration policy for the invitation link.

1. Go to **Manage shared inboxes** in the sidebar.
2. Select a shared inbox and go to the **Members** tab.
3. Click **Invite with a link**.
4. Enter the email address of the recipient.

   You can send an invitation email to any user, including Faspex users.
5. If allowed, configure security settings. Invitation-link security settings are configured by admins at the server level.
6. Click **Send**.

Invite external users to send packages to a shared inbox through a submission link

Share a public submission link to allow anyone with the URL to send a package to the shared inbox without logging in to Faspex.

1. Select a shared inbox from the **Received** mailbox type.
2. Go to the **Members** tab.
3. Copy the public submission link.

   If you do not see the public submission link, the feature is not available for this shared inbox.

Troubleshooting

Common transfer issues

Connect uses HTTPS instead of FASP for transfers

If Connect is using HTTPS (denoted by `[HTTPS]` when transferring), this may mean your HSTS node does not have port `33001` open. Connect requires port `33001` open to transfer using the FASP protocol.

1. SSH to the HSTS node.
2. Edit the `/etc/ssh/sshd_config` file and add:

   ```
   Port 33001
   ```
3. Restart the **sshd** service.

The transfer completes but the status is Unknown or `Failed`

If your transfer successfully completes in the transfer monitor as reported by Connect or HTTP Gateway but the package status is still Unknown, your *node* may not have activity logging configured. For instructions, see "Enable activity logging on a HSTS node" on page 23.

New packages correctly report the transfer status, though statuses are lost for packages sent while the HSTS node was incorrectly configured.

Changing your password

Change your account password.

1. Click your profile and select **Account settings**.
2. Enter your current password.
3. Enter a new password and confirm it.

   By default, the requirement is a strong password that contains at least six characters (with a minimum of one letter, one number and one symbol).
4. Click **Save**.

External user

An external user is a user that is not associated with a Faspex user account. Faspex users can send public packages to external users with a public submission link. Faspex users can also invite external users to send them a package through an email invitation or with a public submission link.

Send a package using an invitation or public submission link

Send a package using an invitation or through a public submission link. By default, you do not need to log in when using an invitation or public submission link, even if you have a Faspex account.

1. Click the link in the email invitation or go to the public submission link in your browser.
2. When prompted, enter an email address to receive a private submission link..
3. Check your email for the private submission link.
4. Open private submission link to go to the send package form.
5. Fill out the send package form.
6. Check your email for a package status email.

Viewing and downloading packages as an external user

External users do not have a Faspex login. External users view and download packages using links in emails they receive after sending a package or when invited by a Faspex user to download a package.

When a Faspex user sends you a package, as a *public user*, Faspex emails you a download link. By default, external users do not need to create an account to download a public package, but an admin can configure Faspex to require external users to create an account before downloading the package (see "Allowing public packages" on page 105).

You can download the whole package by clicking **Download all** or download individual files using the download icon.

When downloading an encrypted package, Connect prompts you for a passphrase. Connect also prompts for a passphrase if the package contains any **.aspera-env** files within the folder hierarchy, even if the package also contains unencrypted files or files encrypted with different passphrases. If you choose to keep downloaded files encrypted, you do not need to enter a password until you attempt to decrypt the files locally.

Workgroup admin

All the procedures in this section assume that you are a workgroup admin and that you are in the Admin application. To log in to the Admin application, click the application switcher icon (⠿) and select **Admin** from the drop-down menu

Managing workgroup members and SAML groups

Add Faspex users and SAML groups to the workgroup. When a user sends a package to a workgroup, all workgroup members can access the package in their inbox.

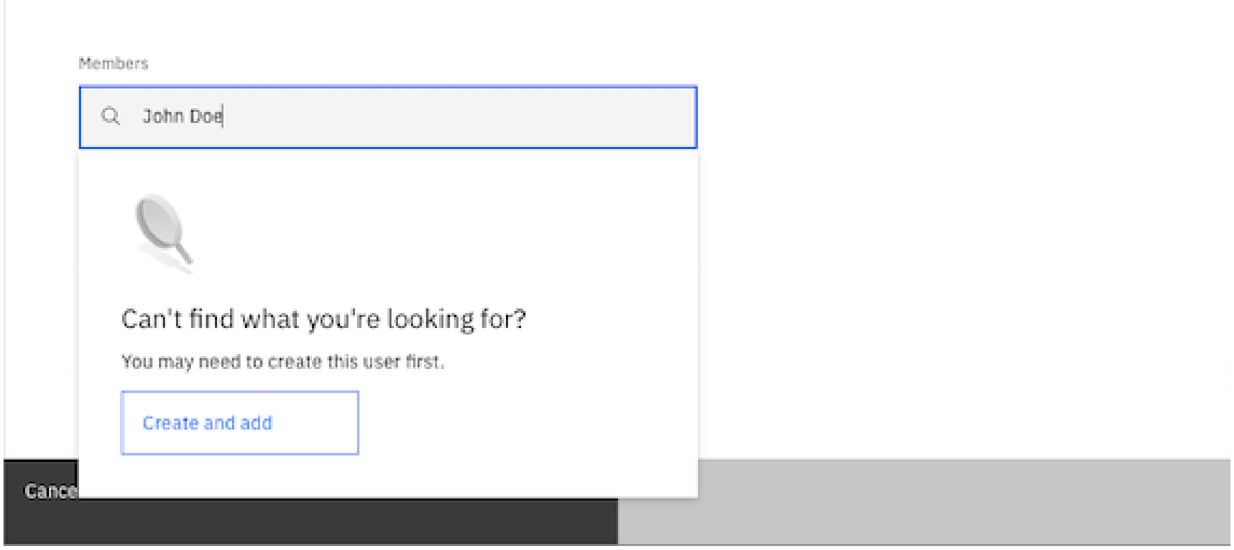
Permissions

Workgroup admins can

Add members

1. After selecting a workgroup, go to the **Members** tab to manage workgroup members.
2. Click **Add member**.
3. Search for and select the Faspex users you want to add.

   If you are a workgroup admin and the workgroup permissions allow you to create and add new users, you can click **Create and add** to create a new user and add the user to the workgroup.



Can't find what you're looking for?

You may need to create this user first.

4. Choose the access permission to grant to these users.
   - **Standard**: These members can see packages sent to the workgroup and can send packages to the workgroup (if permitted by workgroup settings).
   - **Submit only**: These members can send packages to the workgroup (if permitted by workgroup settings).
   - **Workgroup admin**
5. Click **Add**.

Add SAML groups

1. After selecting a workgroup, go to the **Members** tab to manage workgroup members.

2. Select the **SAML groups** toggle.
3. Click **Add group**.
4. Select the group to add.
5. Choose the access permission to grant to these users.
   - **Standard**: These members can see packages sent to the workgroup and can send packages to the workgroup (if permitted by workgroup settings).
   - **Submit only**: These members can send packages to the workgroup (if permitted by workgroup settings).
   - **Workgroup admin**
6. Click **Add**.

Shared inbox admin

All the procedures in this section assume that you are a shared inbox admin and that you have access to **Manage shared inboxes** in the sidebar.

Manage shared inbox members and SAML groups

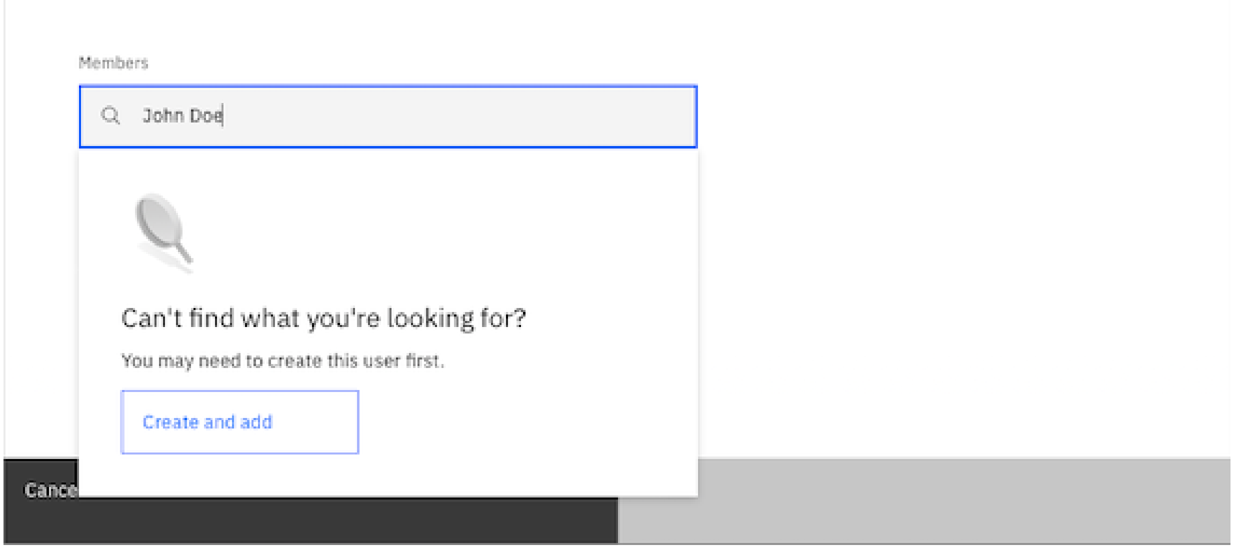
You can add Faspex users and SAML groups to a shared inbox. When a user sends a package to a shared inbox, all shared inbox members can access the package in the shared inbox (under Received).

After selecting a shared inbox, go to the **Members** tab to manage shared inbox members.

Add members

1. Click **Add member**.
2. Search for and select the Faspex users you want to add.

   If you are a shared inbox admin and the shared inbox permissions allow you to create and add new users, you can click **Create and add** to create a new user and add the user to the shared inbox.


Members

John Doe

Can't find what you're looking for?

You may need to create this user first.

3. Choose the access permission to grant to these users.
   - **Standard**
   - **Submit only**
   - **Shared inbox admin**

4. Click **Add**.

Add SAML groups

1. Select the **SAML groups** toggle.
2. Click **Add group**.
3. Select the group to add.
4. Click **Add**.

Invite external users to send packages to a shared inbox through an email

If you are a member of a shared inbox that allows regular users to invite members or you are a shared inbox admin, you can send an invitation email to allow someone to send a package to a shared inbox without registering an account or logging in.

Anyone with the URL can send packages to the shared inbox, but cannot see or download shared inbox packages.

Faspex cannot verify the person using the link is the intended collaborator. If this is a concern, set a custom link expiration policy for the invitation link.

1. Go to **Manage shared inboxes** in the sidebar.
2. Select a shared inbox and go to the **Members** tab.
3. Click **Invite with a link**.
4. Enter the email address of the recipient.

   You can send an invitation email to any user, including Faspex users.
5. If allowed, configure security settings. Invitation-link security settings are configured by admins at the server level.
6. Click **Send**.

Enable public submission links for a shared inbox

Enable a shared inbox's public submission link to allow anyone with the link to send a package to the shared inbox without logging in to Faspex.

1. Go to **Manage shared inboxes** and select a shared inbox.
2. Go to the **Submission link** tab.

   If you do not see the **Submission link** tab, Faspex does not allow shared inbox admins to enable or disable the link.
3. Click **Save**.

Manager

All the procedures in this section assume that you have Faspex manager permissions and that you are in the Admin application. To log in to the Admin application, click the application switcher icon (⠿) and select **Admin** from the drop-down menu

Managing users

Create, configure, and remove Faspex users. For an overview of the different users, see "User roles" on page 3.

User roles

Admins assign user roles to an account when creating a new account or when configuring an account's permissions.

Important: You cannot change your own assigned role.

User accounts can have these user roles:

Role	**Description**
Regular user	Regular users can send and receive packages, as permitted by admin-configured server settings.
Manager	In addition to regular user permissions, managers can manage: • regular users • workgroups • external users • SAML groups Managers can access all shared inboxes and manage shared inbox members and workgroups. Managers cannot create new managers, edit admin accounts, or promote another user to an admin or manager role.
Admin	In addition to manager permissions, admins can adjust server configurations, access all packages and relays, manage all workgroups and shared inboxes, and manage all users.
External user	A external user is a user that is not associated with a Faspex user account. Faspex users can send *public packages* to external users with a *public submission link* and invite external users to send them a package through an email invitation or with a *public submission link*. By default, external users do not need to create an account to download a public package, but an admin can configure Faspex to require external users to create an account before downloading the

Role	Description
	package (see “Allowing public packages” on page 105).

Creating a user

Create a user account in Faspex.

1. Go to **Users > All users**.
2. Click **Create new**.
3. Enter the user's email address. The email must be unique to Faspex.
4. Enter the user's first and last name.
5. Select the user's role:

Role	Description
Regular user	Regular users can send and receive packages, as permitted by admin-configured server settings.
Manager	In addition to regular user permissions, managers can manage: • regular users • workgroups • external users • SAML groups Managers can access all shared inboxes and manage shared inbox members and workgroups. Managers cannot create new managers, edit admin accounts, or promote another user to an admin or manager role.
Admin	In addition to manager permissions, admins can adjust server configurations, access all packages and relays, manage all workgroups and shared inboxes, and manage all users.
External user	A external user is a user that is not associated with a Faspex user account. Faspex users can send *public packages* to external users with a *public submission link* and invite external users to send them a package through an email invitation or with a *public submission link*. By default, external users do not need to create an account to download a public package, but an admin can configure Faspex to require external users to create an account before downloading the package (see “Allowing public packages” on page 105).

6. You can further configure the user by expanding **Account settings** and **Permissions**. See “Reference: User account settings” on page 166 and “Reference: User permissions” on page 168, respectively.
7. Click **Create**.

Faspex sends a welcome email to the email address. Using the link provided in the email, the user creates a password and logs in.

Reactivating an inactive account

A user account becomes inactive if an admin deactivates the user or if the user deactivates due to inactivity.

To reactivate a user:

1. Go to **Users > All users**.
2. Select a user.
3. Enable **Account is active** in the Account settings section.

Deleting a user

Deleting a user does not delete a user's received packages. This means that:

- Admins can still see and download packages sent to a deleted user.
- Users that received a package from a deleted user can still access the package, but the package sender is changed to your account.

Note: Deleting a user does not remove that user from distribution lists. If a user sends a package to a distribution list containing a deleted user, Faspex sends the package to the user's email address as a new external user. If the sender is not allowed to send to external users, sending the package fails.

1. Go to **Users > All users**.
2. Right-click the user you want to delete and select **Delete**.

   You can also delete multiple users at once.
3. Confirm deletion.

Managing workgroups

Workgroups define a group of users that can be sent packages as a collective whole. Use workgroups set the same permissions for a group of users.

A Faspex administrator determines:

- Who has permissions to send packages to a workgroup, including whether workgroup members can see and send packages to other workgroup members.
- Where packages sent to the workgroup are stored.

Creating a workgroup

Create a workgroup and add members to the workgroup.

This procedure covers the minimum requirements to create a workgroup:

1. Go to the **Admin** app from the app switcher.
2. Go to **Workgroups**.
3. Click **Create new**.
4. Name the workgroup.
5. Click **Save**.
6. Select the newly created workgroup.
7. Go to the **Members** tab.
8. To add Faspex users:

   a) Click **Add member**.

   b) Search for and select the Faspex users you want to add.

c) Select the workgroup permissions for the users.

d) Click **Add**.

For more information on the **Permissions** options section, see "Configuring workgroup permissions and privacy settings" on page 58.

Configuring workgroup permissions and privacy settings

Faspex admins can designate workgroup members as workgroup admins. Workgroup admins can have additional permissions granted per workgroup. All Faspex admins are workgroup admins of every workgroup.

To manage workgroup settings, go to the **Settings** tab of any workgroup.

Workgroup admin permissions

There, you can enable workgroup admins to:

- **Add existing IBM Aspera users and remove non-admin members**: Workgroup admins can add any user from the user directory to the workgroup. This must be enabled for workgroup admins to add members.
- **Create and add new IBM Aspera users**: Workgroup admins can create a new user and add that user to the workgroup.
- **Add and remove SAML groups**: Workgroup admins can add any SAML group to the workgroup.

Note:

- All Faspex admins and managers have these permissions for every workgroup.
- IBM Aspera users do not include external users.

Workgroup privacy settings

You should restrict visibility to increase member privacy.

Who can see and send packages to this workgroup

- No one
- Workgroup members only
- Workgroup admins only
- Anyone

Who can see and send packages to other workgroup members

- No one
- Workgroup admins only
- All workgroup members

Important: If you set the permission for both options No one, no one can see or send to the workgroup and workgroup members cannot see or send to each other, so members in the workgroup are members in name only.

Managing workgroup members and SAML groups

Add Faspex users and SAML groups to the workgroup. When a user sends a package to a workgroup, all workgroup members can access the package in their inbox.

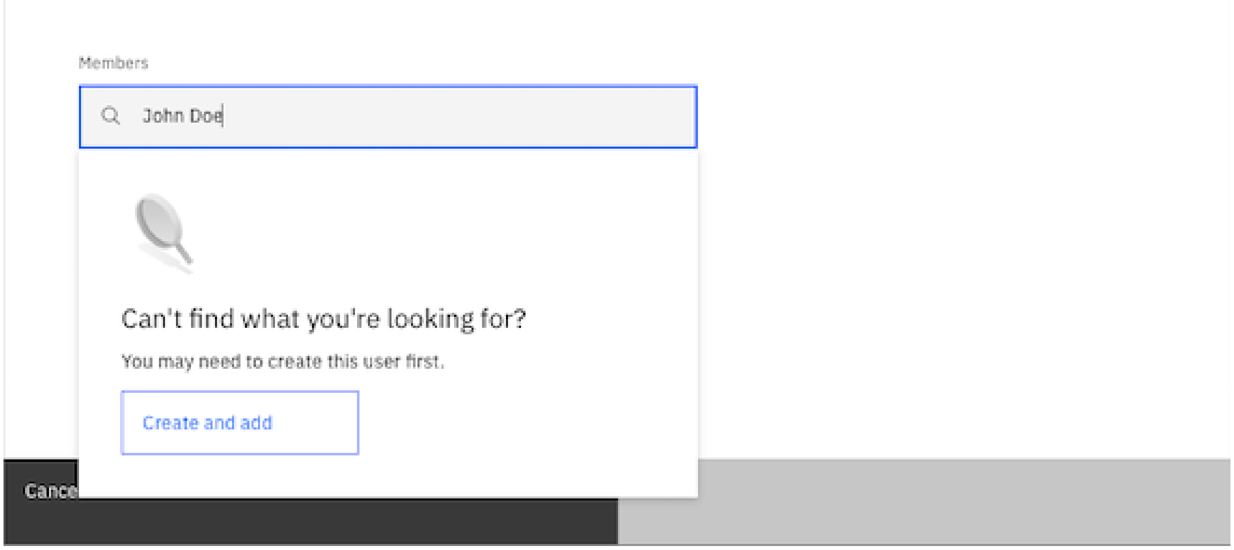
Permissions

Workgroup admins can

Add members

1. After selecting a workgroup, go to the **Members** tab to manage workgroup members.
2. Click **Add member**.
3. Search for and select the Faspex users you want to add.

   If you are a workgroup admin and the workgroup permissions allow you to create and add new users, you can click **Create and add** to create a new user and add the user to the workgroup.

   Can't find what you're looking for?

   You may need to create this user first.

4. Choose the access permission to grant to these users.
   - **Standard**: These members can see packages sent to the workgroup and can send packages to the workgroup (if permitted by workgroup settings).
   - **Submit only**: These members can send packages to the workgroup (if permitted by workgroup settings).
   - **Workgroup admin**
5. Click **Add**.

Add SAML groups

1. After selecting a workgroup, go to the **Members** tab to manage workgroup members.
2. Select the **SAML groups** toggle.
3. Click **Add group**.
4. Select the group to add.
5. Choose the access permission to grant to these users.
   - **Standard**: These members can see packages sent to the workgroup and can send packages to the workgroup (if permitted by workgroup settings).
   - **Submit only**: These members can send packages to the workgroup (if permitted by workgroup settings).
   - **Workgroup admin**
6. Click **Add**.

Setting a custom destination for a workgroup

Choose your workgroup destination inbox. Packages sent to the workgroup are stored at this location.

If you have a group of users that are not permitted to access the server-default storage location, use a workgroup with a custom inbox configured. Packages sent to the workgroup are first uploaded to the server-default storage location, and then relayed to the specified storage location. When the workgroup users download the package, they download the package from the custom inbox location.

When packages are deleted from the default location, they are not automatically removed from the custom location.

Note: When symbolic links are enabled for a storage location, packages in the custom inbox location are stored as actual files, not symbolic links. The default inbox location contains symbolic links, but the custom inbox contains actual files.

1. Select a workgroup and go to the **Nodes and storage** tab.
2. Under inbox destination, select **Custom**.
3. To upload directly to the specified storage location instead of first uploading to the server-default storage location, select **Upload directly to custom inbox**.
4. Select a node from the table and select the storage location you want to use as the custom inbox.
5. Click **Save**.

Forwarding workgroup packages to another storage location

Forward packages sent to a workgroup to another storage location using relays.

Packages sent to a workgroup are first uploaded to the server-default storage location and then relayed to the specified storage locations. When packages are deleted from the default location, they are not automatically removed from the custom location.

1. Select a workgroup and go to the **Nodes and storage** tab.
2. Select **Relay**.
3. Select **Enable relays**.
4. Click **Save**.
5. For each storage location you want to relay packages to:
   a) Select the storage location.
   b) Enable **Forward to this relay**.
   c) If you want to overwrite files if they exist on the destination, enable **Overwrite files**.
   d) If you want to notify users when a relay starts, completes, or errors out, enter contacts in the notification fields.
   e) Click **Save**.

Approving or denying pending registrations

1. Go to **Users > All users** and click the **Pending registrations** tab.
2. Click the **Approve** or **Deny** icon.

Approved users receive the Faspex welcome email and can use the password reset link in the email to set their passwords and log in.

Approved users automatically inherit the permissions of the template user and will become members of a workgroup, if configured to do so. After creation, you can update the permissions and workgroup memberships of these users from the **Users** tab.

Managing shared inboxes

Shared inboxes provide a file submission system for users to share packages. Shared inbox admins can also invite external users (people who don’t have a Faspex account) to send to a shared inbox.

The topics in this section assume that you are an admin, manager, or shared inbox admin. Shared inbox admins can expand access beyond shared inbox members by allowing anyone with a shared inbox-specific public submission link to send packages to the shared inbox.

Common uses

Use several shared inboxes to manage assets for different projects or business processes and require different package metadata for each inbox

Allow external users to send packages without giving them full access to Faspex or requiring them to log in.

Create a shared inbox

1. Go to the **Packages** app from the app switcher.
2. Go to **Manage shared inboxes**.
3. Click **Create new**.
4. Name the shared inbox.
5. Click the **Save** button.
6. Select the newly created shared inbox.
7. Go to the **Members** tab.
8. To add Faspex users:
   a) Click **Add member**.
   b) Search for and select the Faspex users you want to add.
   c) Select the shared inbox permissions for the users.
   d) Click **Add**.

Manage shared inbox members and SAML groups

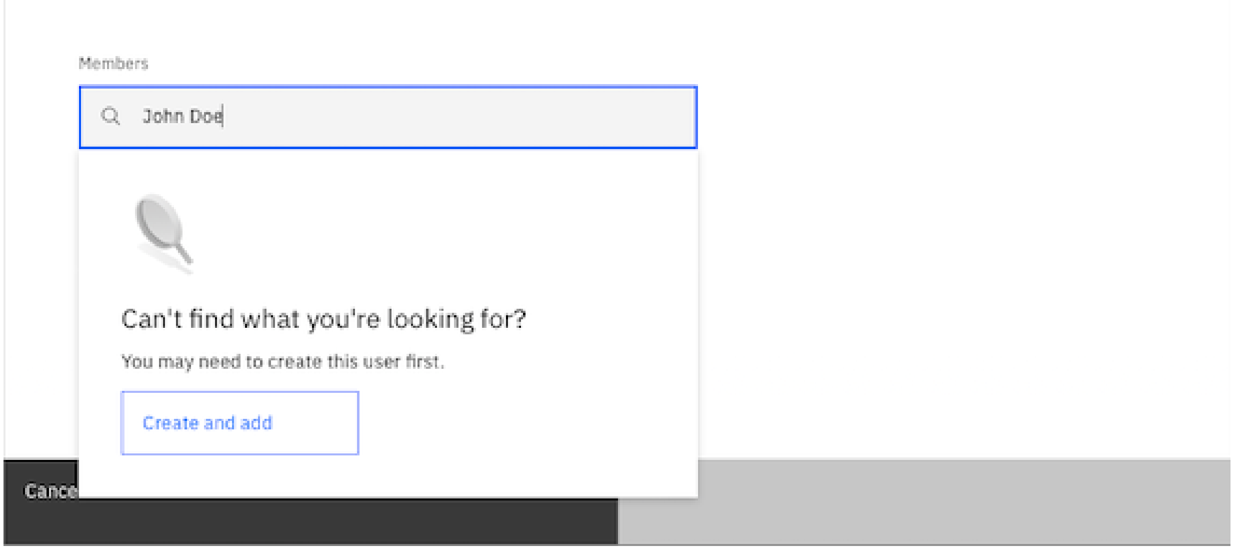
You can add Faspex users and SAML groups to a shared inbox. When a user sends a package to a shared inbox, all shared inbox members can access the package in the shared inbox (under Received).

After selecting a shared inbox, go to the **Members** tab to manage shared inbox members.

Add members

1. Click **Add member**.
2. Search for and select the Faspex users you want to add.

   If you are a shared inbox admin and the shared inbox permissions allow you to create and add new users, you can click **Create and add** to create a new user and add the user to the shared inbox.



Members

John Doe

Can't find what you're looking for?

You may need to create this user first.

3. Choose the access permission to grant to these users.

   - **Standard**
   - **Submit only**
   - **Shared inbox admin**
4. Click **Add**.

Add SAML groups

1. Select the **SAML groups** toggle.
2. Click **Add group**.
3. Select the group to add.
4. Click **Add**.

Invite external users to send packages to a shared inbox through an email

If you are a member of a shared inbox that allows regular users to invite members or you are a shared inbox admin, you can send an invitation email to allow someone to send a package to a shared inbox without registering an account or logging in.

Anyone with the URL can send packages to the shared inbox, but cannot see or download shared inbox packages.

Faspex cannot verify the person using the link is the intended collaborator. If this is a concern, set a custom link expiration policy for the invitation link.

1. Go to **Manage shared inboxes** in the sidebar.
2. Select a shared inbox and go to the **Members** tab.
3. Click **Invite with a link**.
4. Enter the email address of the recipient.

   You can send an invitation email to any user, including Faspex users.
5. If allowed, configure security settings. Invitation-link security settings are configured by admins at the server level.
6. Click **Send**.

Enable public submission links for a shared inbox

Enable a shared inbox's public submission link to allow anyone with the link to send a package to the shared inbox without logging in to Faspex.

1. Go to **Manage shared inboxes** and select a shared inbox.
2. Go to the **Submission link** tab.

   If you do not see the **Submission link** tab, Faspex does not allow shared inbox admins to enable or disable the link.
3. Click **Save**.

Configuring shared inbox admin settings

Faspex admins can designate members as shared inbox admins. Shared inbox admins have additional permissions configured per shared inbox. All Faspex admins are shared inbox admins of every shared inbox.

To manage shared inbox admin permissions, go to the **Settings** tab of any shared inbox. There, you can enable shared inbox admins to:

- **Add existing Faspex users and remove non-admin members**
- **Invite and remove outside submitters**
- **Create, remove and delete new Faspex users**

⚠ **Warning:** This permission allows shared inbox admins to remove Faspex users globally and permanently from the system.

- **Add and remove SAML groups**

Note: All Faspex admins have all these permissions for every shared inbox.

Set the package expiration policy for a shared inbox

Set a package expiration policy.

Setting a package expiration policy overrides the global package expiration setting. If global package expiration is enabled, but you want to disable time-based and download-based package expiration for only this shared inbox, select **None** for downloads-based package expiration and clear **Time-based policy**.

To set a package expiration policy:

1. Go to the **Settings** tab of any shared inbox.
2. Toggle **Set package expiration policy**.
3. Select an option for downloads-based package expiration.

   You can choose these download-based policies:

Option	Description
None	Do not delete the submitted package after it is downloaded.
Delete files after any recipient downloads all files	Delete the package if *any* member downloads all the files in the package. **Note:** When using this policy, forwarding the package to another recipient can prevent the original recipient from downloading the package. If a package is forwarded to and downloaded by another recipient, Faspex deletes the package contents and the original recipient cannot download the package.
Delete files after all recipients download all files	Delete the package if *all* members have downloaded all the files in the package.

4. To set a time-based policy, select **Time-based policy** and enter the number of days the package is available before deleting the package.
5. Click **Save**.

Set the link expiration policy for a shared inbox

Set an invitation link expiration policy. Invitation links allow anyone with the link to send package to the shared inbox.

Setting an invitation link expiration policy overrides the global invitation link expiration setting. If global invitation link expiration is enabled, but you want to disable time-based and sending-based link expiration for only this shared inbox, clear **Expire link after a collaborator successfully submits a package** for downloads-based package expiration and clear **Time-based policy**.

To set an invitation link expiration policy:

1. Go to **Manage shared inboxes**.
2. Select the shared inbox.
3. Go to the **Settings** tab.
4. Toggle **Set invitation expiration policy**.

5. To set sending-based link expiration, select **Expire link after a collaborator successfully submits a package**.

   Note: While the link expires after a collaborator successfully sends a package, it is possible for a collaborator to initiate parallel uploads using a single link to submit multiple packages.
6. To set a time-based policy, select **Time-based policy** and enter the number of days the link stays valid.
7. Click **Save**.

Assigning a metadata profile to a shared inbox

You can apply a metadata profile to a shared inbox. Metadata profiles define additional optional and required fields for users to fill when sending a package.

For instructions on creating a metadata profile, see "Create a metadata profile" on page 106.

1. Go to the **Packages** app from the app switcher.
2. Go to **Manage shared inboxes**.
3. Select the desired shared inbox.
4. Go to **Metadata**.
5. Select a metadata profile from the **Metadata profile** drop-down menu. If the only option is **None**, there are no metadata profiles created.
6. To save the metadata of sent packages to a file in the package, select **Save metadata to file**.

   The metadata is saved to the package's root directory as `aspera-metadata.xml`.
7. Click **Save**.

Setting a custom destination for a shared inbox

Choose your shared inbox destination inbox. Packages sent to the shared inbox are stored at this location. Only Faspex admins can set a custom inbox; shared inbox admins do not have this ability.

When using a custom inbox, packages sent to the shared inbox are first uploaded to the server-default storage location, and then relayed to the specified storage location. When packages are deleted from the default location, they are not automatically removed from the custom location.

Note: When symbolic links are enabled for a storage location, packages in the custom inbox location are stored as actual files, not symbolic links. The default inbox location contains symbolic links, but the custom inbox contains actual files.

1. Select a shared inbox and go to the **Nodes and storage** tab.
2. Under inbox destination, select **Custom**.
3. To upload directly to the specified storage location instead of first uploading to the server-default storage location, select **Upload directly to custom inbox**.
4. Select a node from the table and select the storage location you want to use as the custom inbox.
5. Click **Save**.

Forwarding shared inbox packages to another storage location

Forward packages sent to a shared inbox to another storage location using relays.

When relays are configured, packages sent to the shared inbox are first uploaded to the server-default storage location, and then relayed to the specified storage locations. When packages are deleted from the default location, they are not automatically removed from the custom location.

1. Select a shared inbox and go to the **Nodes and storage** tab.
2. Select **Relay**.
3. Select **Enable relays**.
4. Click **Save**.

5. For each storage location you want to relay packages to:
    a) Select the storage location.
    b) Enable **Forward to this relay**.
    c) If you want to overwrite files if they exist on the destination, enable **Overwrite files**.
    d) If you want to notify users when a relay starts, completes, or errors out, enter contacts in the notification fields.
    e) Click **Save**.

Admin

All the procedures in this section assume that you have Faspex admin permissions and that you are in the Admin application (unless instructed to log in to Faspex Utility). To log in to the Admin application, click the application switcher icon (⋮⋮⋮) and select **Admin** from the drop-down menu

How does Faspex work?

Faspex is a centralized transfer solution that enables users to exchange files with each other using an email-like workflow. User-uploaded files and folders are sent to, stored on, and downloaded from *nodes (also known as Aspera transfer servers)*.

When users send packages to recipients, they upload content to an Aspera transfer server for storage. Faspex notifies the recipients that the package is available. Recipients can then download a copy of the package from the remote server.

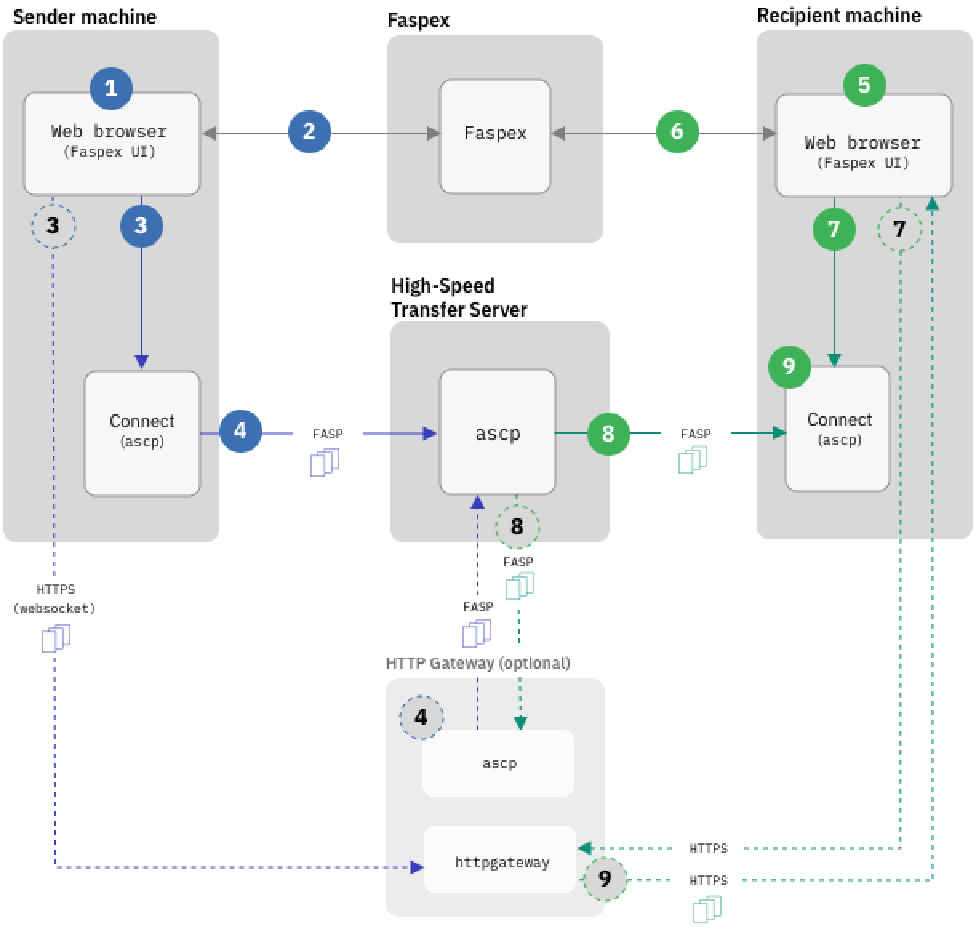
Faspex leverages the IBM Aspera FASP protocol to guarantee fast delivery times with high security over variable network conditions. The FASP protocol requires two **ascp**-enabled machines (client and node) to establish a FASP transfer session. To leverage FASP for transfers, Faspex either:

- requires users to install IBM Aspera Connect, which handles the FASP transfer session with the HSTS node.
- requires users to transfer files (over HTTPS) through the IBM Aspera HTTP Gateway, which handles the FASP transfer session with the HSTS node.

Faspex does not perform the transfer, but connects the user's machine with Connect or HTTP Gateway to perform the transfer.

If permitted, users can also send a package with content from a shared folder. Depending on your configuration, these transfers may involve more than one node in a node-to-node transfer.

How a transfer works



1. The sender uses the Faspex UI to create a package for delivery a recipient.
2. The Faspex UI makes a request to the Faspex server, which passes back a transfer specification used by either Connect or HTTP Gateway to start the transfer.
3. The Faspex UI passes on the information to Connect or HTTP Gateway. In the case of HTTP Gateway, the Faspex UI also sends the package over HTTPS to the HTTP Gateway server.
4. Connect or HTTP Gateway initiates a FASP transfer session with the HSTS node and transfers the package to the HSTS node over FASP.
5. The recipient requests to download a package using the Faspex UI.
6. The Faspex UI makes a request to the Faspex server, which passes back a transfer specification used by either Connect or HTTP Gateway to start the transfer.
7. The Faspex UI passes on the information to Connect or HTTP Gateway.
8. Connect or HTTP Gateway initiates a FASP transfer session with the HSTS node and the HSTS node transfers the package to Connect or HTTP Gateway over FASP.
9. Connect saves the file to the recipient's machine. In the case of HTTP Gateway, HTTP Gateway uses the web browser's download manager to save the file.

Faspex architecture and components

Containers

The Faspex server uses an ecosystem of containers managed by Docker.

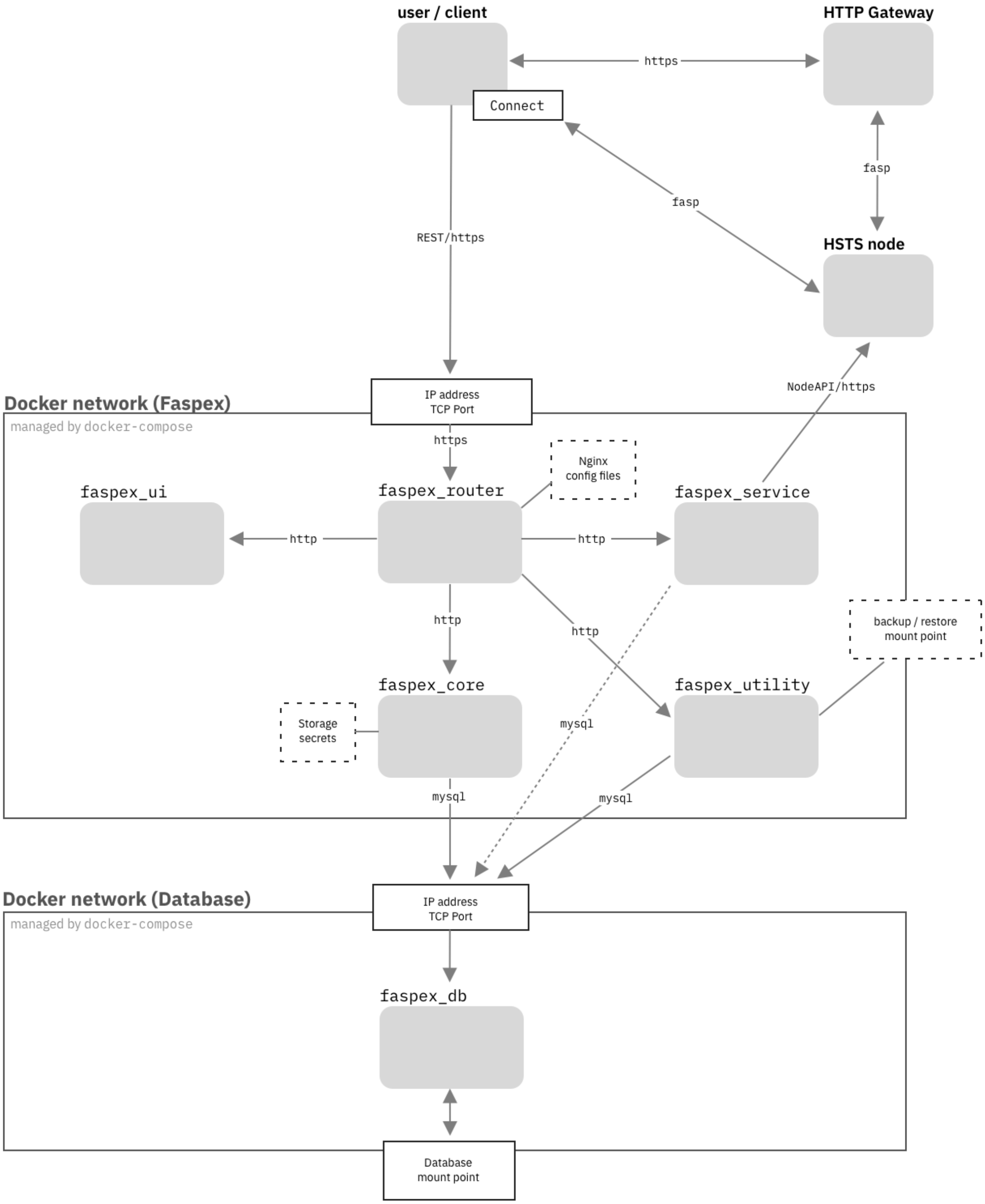
When managing containers with Docker, you must have both Docker and **`docker-compose`** available on your machine. The **`faspexctl`** utility uses the **`docker-compose`** utility to manage Docker containers.

Faspex runs instances of these containers:

Container	Usage	Docker dependencies	Ports used	File system access
`faspex-router`	Services router	These containers must be up and running: • `faspex-core` • `faspex-ui` • `faspex-utility` • `faspex-service`	`80 and 443` These ports are defined by environment variables in `/opt/aspera/faspex/conf/docker/router.env`: • FASPEX_ROUTER_HTTP_PORT (default 80) • FASPEX_ROUTER_HTTPS_PORT (default 443)	The `faspex-router` container requires file system access to store data. `faspex-router` has access to mount locations defined in the `/opt/aspera/faspex/conf/docker/router.env` environment file. `faspex-router` uses file system access to read certificates on the host.
`faspex-ui`	UI web server	These containers must be up and running: • `faspex-core`		
`faspex-core`	API server	These containers must be up and running: • `faspex-db`		
`faspex-service`	Queue handler for background jobs	These containers must be up and running: • `faspex-core`		
`faspex-utility`	Web application for managing database migrations and backups	These containers must be up and running: • `faspex-db`		The `faspex-utility` container requires file system access to save and load db backup data. Define`faspex-utility` has access to mount locations defined in the `/opt/aspera/faspex/conf/docker/`

Container	Usage	Docker dependencies	Ports used	File system access
				`utility.env` environment file.
`faspex-db`	Database The `faspex-`dbcontainer is provided as a convenience, but customers should use their own external database, especially for scalable, high-availability environments. In this case, Faspex removes `faspex-db` from the `faspex-core` dependency list. For instructions see, "Installing Faspex with a remote database" on page 9.			The `faspex-db` container requires file system access to store data. Define the `faspex-db` mount locations in the `/opt/aspera/faspex/conf/docker/db.env` environment file.

The installer provides the **`faspexctl`** utility to manage the containers.



Faspex 5 application file storage

Except for containers, all Faspex application files are stored in `/opt/aspera/faspex`.

If using Docker as your container virtualization system manage, container images are stored by default in `/var/lib/docker`.

Faspex package states and statuses

A package has a both a state and an upload status. The Faspex 5 UI uses a combination of these values to report the package status.

Package statuses

Package status	State	Upload status
Starting	held	submitted
Uploading	held	queued
Uploading (%, ETA)	held	transferring
Paused	held	paused or stopped
Failed + error message	held	timed_out, will_retry, error or failed
Uploaded for processing	held	complete
Processing	transient_held_to_released_start	complete
Checking relays	transient_held_to_released_routing_packages	complete
Relaying	transient_held_to_released_finish	complete
Complete	released	complete
Expiring	transient_released_to_expired	complete
Expired	expired	complete
Deleting	transient_released_to_deleted or transient_expired_to_deleted	complete
Deleted	deleted	complete

A list of package states and upload statuses

Package states:

- held: Faspex is waiting for the package upload to complete or for pending package to be released/
- released: Package contents exist on the server and are available.
- expired: Package contents exist but are unavailable.
- deleted: Package contents no longer exist on the server.
- transient_held_to_released_start: A background job is running post-processing tasks such as email notifications.
- transient_held_to_released_routing_packages: A background job is checking if the package has relays.
- transient_held_to_released_finish: All relay jobs are submitted and Faspex is waiting for relay transfers to complete.
- transient_held_to_expired: A background job is changing the state of this package from held to expired.

- transient_held_to_deleted: A background job is changing the state of this package from held to deleted.
- transient_released_to_expired: A background job is changing the state of this package from released to expired.
- transient_released_to_deleted: A background job is changing the state of this package from released to deleted.
- transient_expired_to_deleted: A background job is changing the state of this package from expired to deleted.

Note: Transient states represent in-between states when transitioning from one state to another.

Upload statuses:

- completed
- error
- failed
- paused
- queued
- stopped
- submitted
- timed_out
- transferring
- will_retry
- pre_initiation

The Faspex Utility web application

Access Faspex Utility at https://*your_faspex_server*/aspera/faspex/utility. Use Faspex Utility to backup and restore the database, migrate the database to new versions, and add or update user accounts.

What do I use Faspex Utility for?

- "Backing up the database" on page 144
- "Restoring the database" on page 144
- "Migrating the database" on page 143
- Creating and updating user accounts without logging in
- Determining duplicate email addresses
- Monitoring database health

Logging in

Faspex Utility uses different credentials than the Faspex web application for logging in.

The credentials for Faspex Utility are set using the /opt/aspera/faspex/conf/docker/utility.env file. Use the values of the FASPEX_UTILITY_USERNAME and FASPEX_UTILITY_PASSWORD variables to log in. If you change these values in the .env file, update Faspex Utility by running:

```
faspexctl setup
```

To log in:

1. Go to https://*your_faspex_server*/aspera/faspex/utility.

2. Log in with the Utility username and password.

Authorized list of IP Addresses

As an Admin you must explicitly authorize at least one source IP address to be able to access the Faspex Utility. Perform the following steps to include specific IP addresses and IP ranges to the authorized list.

1. Modify the `utility.env` file located in the `/opt/aspera/faspex/conf/docker` directory to include specific IP addresses and IP ranges of anyone connecting via browser to the `FASPEX_UTILITY_AUTHORIZED_IPS` authorized list.

   Important: The IP addresses/IP ranges must be in a comma separated list. You can also use wildcards, for example: `1.1.1.1,192.*` opens up your utility to `1.1.1.1` plus every IP address that starts with `192.`
2. Restart the Faspex Utility and router to apply your changes:

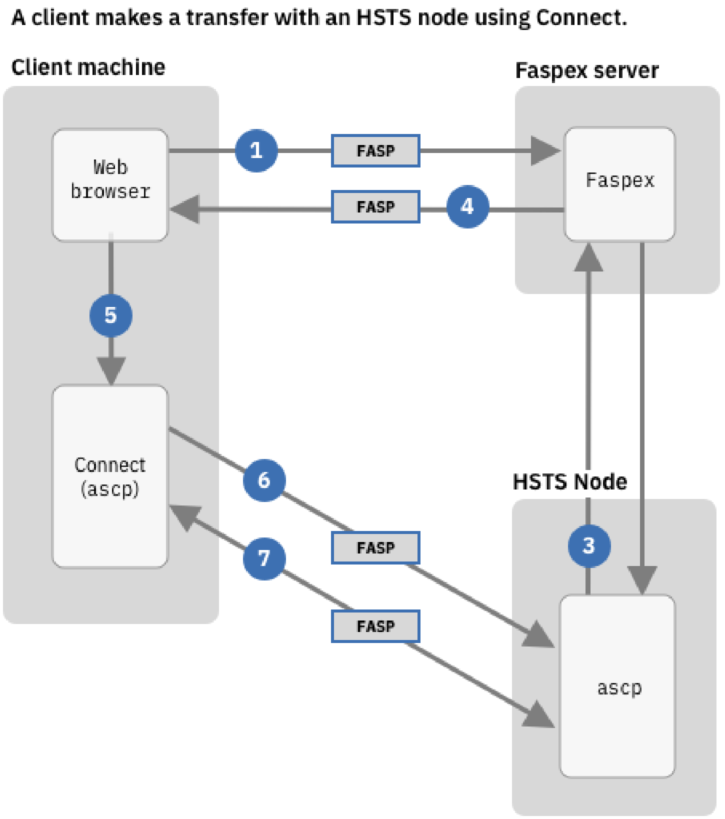
   ```
   faspexctl restart utility
   faspexctl restart router
   ```

IBM Aspera Connect

IBM Aspera Connect is an install-on-demand browser extension and desktop client that facilitates high-speed uploads and downloads with an Aspera transfer server.

When a user first logs in, Faspex checks if Connect has been installed. If Connect is outdated or not installed, Faspex prompts the user to install the browser extension and to download and install the Connect client.

Transfers with Connect



1. The web browser makes a transfer request to the web application.
2. The web application creates a transfer specification and submits it to the HSTS node.
3. The HSTS node returns a transfer token.
4. Faspex inserts the transfer token into the transfer specification and passes the specification to the web browser.
5. The browser passes the transfer specification to Connect.
6. Connect initiates a FASP transfer session with the HSTS node through ascp using the transfer token.
7. Connect performs the transfer with the HSTS node.

Transfers with HTTP Gateway Service

You can choose to use the HTTP Gateway service instead of Connect to make transfers using Faspex. For more information about transferring with HTTP Gateway, see "Sending a package with uploaded content using HTTP Gateway" on page 40.

Self-hosting IBM Aspera Connect

In some situations where Faspex users are unable to install IBM Aspera Connect from the CloudFront CDN, Faspex admins can host the connect-deployer locally. This is typically used in air-gap environments.

By default, the latest versions of IBM Aspera Connect are hosted publicly; for example, at `https://d3gcli72yxqn2z.cloudfront.net/downloads/connect/latest/`

Hosting IBM Aspera Connect privately is an option for customers who:

- Don't allow users to download from an external URL like CloudFront
- Use air-gap or offline environments

Important: To host IBM Aspera Connect locally you will need to install the `connect-deployer` container. Starting with Faspex 5.0.5 the installation of this container is managed by `faspexctl setup`.

Prerequisites

- This procedure assumes that you will host Connect on your Faspex 5 server.

Procedure

1. If you have not already installed the `connect-deployer` container during setup, rerun `faspexctl setup`.
2. When prompted with the `Install the connect-deployer container? [y/n] (default: y):` type y to install it.
3. Verify that all your containers are running including the `connect-deployer` container .

   ```
   faspexctl status
   ```

4. Verify that users can access Connect at `https://`*`<faspex_server>`*`/aspera/connect`.
5. Access the Faspex 5 web interface and direct Faspex 5 to use the locally hosted connect:
   a. Go to **Admin > Configurations > Transfer options** .
   b. Check the box to select **Override Connect installation URL**.
   c. Enter the Connect URL: `https://`*`FQDN`*`/aspera/connect`.
   d. In the **Override Connect minimum version** field, select the version of the Connect app you hosted.
   e. Click **Save**.

Disable Connect for specified user groups

If you have HTTP Gateway enabled, you can default users to use HTTP Gateway by disabling Connect for those users.

1. In the admin app, go to **Configurations > Aspera HTTP Gateway**.
2. Go to the **Connect prompt behavior** tab.
3. Choose the type of users to disable Connect for:

Option	Registered users	External users
No one	Prompt	Prompt
Unauthenticated users	Prompt	Do not prompt
All users	Do not prompt	Do not prompt

Update the minimum required version of IBM Aspera Connect

Update the minimum required version of Connect to ensure your end users are all using a version of Connect with correct compatibility for your version of Faspex.

Use the latest version of Connect.

1. In the Admin app, go to **Configurations > Transfer options**.
2. Update the **Minimum Connect version** field.
3. Click **Save**.

IBM Aspera HTTP Gateway

HTTP Gateway is a standalone product that allows a web application to leverage IBM Aspera technology to transfer data without requiring the end user to install IBM Aspera Connect.

HTTP Gateway acts as a reverse proxy between a supported, modern web browser and an IBM Aspera High-Speed Transfer (HSTS) server, also known as a node. HTTP Gateway allows upload and download operations to the node by leveraging native, web-browser capabilities using HTTP.

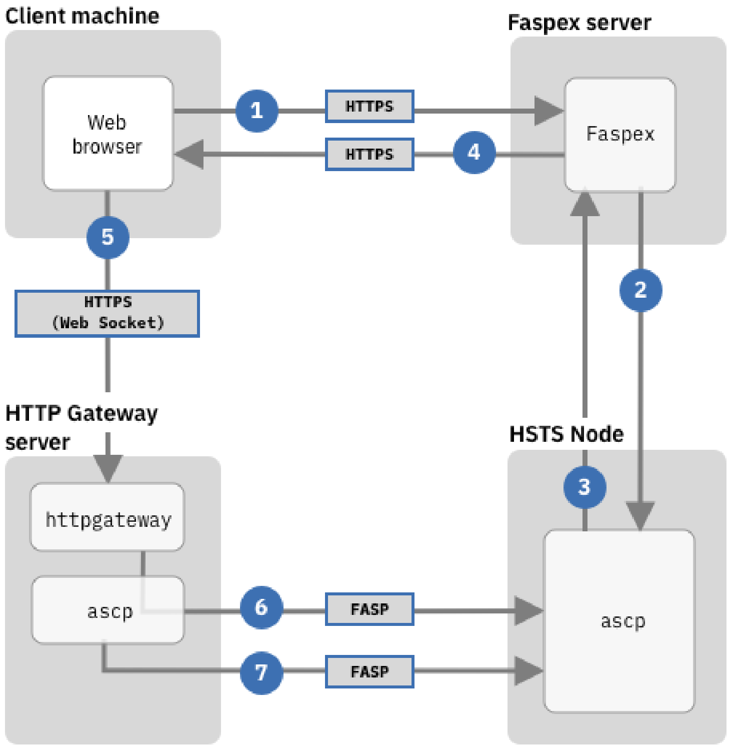
When enabled, Faspex users without Connect send and download packages using HTTP Gateway. Users with Connect can choose to send using HTTP Gateway instead of Connect by disabling Connect in their account preferences.

An admin can also effectively default users to use HTTP Gateway by suppressing the Connect installation prompt. Users can still download and use Connect, but they are not prompted to do so.

HTTP Gateway is a standalone product that has separate documentation. For instructions on installing and configuring HTTP Gateway, see the *IBM Aspera HTTP Gateway Admin Guide*.

Transfers with HTTP Gateway

A client uploads files to an HSTS node using HTTP Gateway.



Limitations

HTTP Gateway-based transfers are limited by the same limitations of HTTP/HTTPS transfers and the native download manager of the client's web-browser.

General Limitations

- Since uploads and downloads leverage HTTP/HTTPS between the web browser and HTTP Gateway, the transfer performance depends on the performance of HTTP/HTTPS, which can be affected by distance between clients and servers, and by other network-related issues. For this reason, follow best practice by deploying HTTP Gateway as close as possible to end users.

- Empty (0-byte) files are not supported for uploads and downloads.
- HTTP Gateway does not support resuming transfers.

Download Limitations

When downloading more than one file, HTTP Gateway bundles the files in-memory. Bundling the files allows end users to download multiple files at once as one archive. Bundling the files also allows preserving a directory structure in the archive.

The total size of the archive cannot be communicated to the web browser, because files are bundled and transferred in-memory. Therefore, the download manager cannot show progress based on the total size.

Upload Limitations

- Since web browsers do not have an upload manager, IBM Aspera provides an upload mechanism through a JavaScript SDK. The upload mechanism allows sending multiple files as chunks, allowing the web page implementing the SDK to send large amounts of data. Because the web page sends the chunked data in the background, the user must stay on the same web page until the upload finishes.

  Note: You can find the JavaScript SDK documentation on the IBM Developer website.
- HTTP Gateway supports uploading only files and not directories.

Enable HTTP Gateway

Enable HTTP Gateway and provide Faspex with the URL to your HTTP Gateway server.

Important: The HTTP Gateway server must be running version 2.2 and above to support uploading folders.

1. Go to **Configurations > Aspera HTTP Gateway**.
2. Enable the **Enable HTTP Gateway** toggle.
3. Enter the URL of your HTTP Gateway server with (with namespace `/aspera/http-gwy/v1`) in the **HTTP Gateway URL** field. For example: `https://gateway.example.com/aspera/http-gwy/v1`.
4. Make sure Faspex can establish a connection with the HTTP Gateway server by clicking **Test connection**.
5. Click **Save**.

Managing nodes and file storage

User-uploaded files and folders are sent to, stored on, and downloaded from configured nodes.

Adding a node to Faspex

A node is a server running an Aspera transfer server product (such as IBM Aspera High-Speed Transfer Server) configured for use with Faspex. User-uploaded files and folders are sent to, stored on, and downloaded from Aspera transfer servers.

Faspex requires one storage location with one default storage to enable users to transfer content, but you can add as many nodes as you want.

A common use case for adding more nodes is to transfer packages to nodes that are closer to distribution servers. For example, instead of doing cross-country transfers with end users, you can choose to transfer content to a node located in the destination country. Users in that country could then download the package from the geo-located node.

All nodes must be configured to interact with Faspex before they can be added to Faspex:

- **Windows**: "Configuring a Windows node for Faspex" on page 82

- **Linux**: “Configuring a Linux node for Faspex” on page 79
- **OS X**: “Configuring a MacOS node for Faspex” on page 85

To add a configured node to Faspex:

1. In the Admin app, go to **Nodes and Storage**.
2. Click **Create node**.
3. Enter a unique name to identify the node.
4. To encrypt the connection to the node using SSL, enable **Use SSL**.
5. To verify the SSL certificate, enable **Verify SSL Certificate**.
6. Set the node type:
   - **Node**: Use the **Node** type when connecting to a standalone HSTS node or to a cluster of HSTS nodes not using a common Redis database.

     If the node host represents a cluster of HSTS nodes not using a common Redis database, Faspex resolves the node host into a list of IP addresses and polls each one of those IP addresses for the full view of all the transfers happening with the cluster.
   - **Cluster**: Use the **Node** type when connecting to an ATS node or to a HSTS cluster with a common Redis database.

     Faspex polls only the node host and expects the host to represent all the transfers happening with the cluster based on these HSTS hosts forming one cluster with a common Redis database.
7. Provide Faspex the information needed to connect to the Node API on the transfer server:

   Host
   The node's hostname or IP address. To avoid connectivity problems, do not specify a hostname that contains underscores.

   Port
   The Node API port number. By default, the port is 9092.

   Username
   The Node API username on the node machine or an access key for cluster.

   Password
   The Node API password on the node machine or the access key secret for a cluster.
8. Click **Create**.

   Faspex checks if Faspex can connect to the HSTS node with the provided information. If Faspex cannot connect, Faspex warns you, but allows you to **Proceed anyway**.
9. To use a node, you must add file storage to the node:
   a) Right-click the node you just created and select **Add file storage** from the drop-down menu.
   b) Click **Create storage location**.
   c) Enter a name for the file storage.
   d) Click **Create**.
10. If this is the first node you are adding to Faspex, make it the default inbox. Faspex requires you set a default inbox before users can send packages. Right-click the storage location you just created and select **Make default inbox** from the overflow menu.

Creating new storage locations

Storage locations are directories on a node. You can use files and folders in storage locations as the source of content for a package. You can also store package content in specific storage locations instead of the server default location. Only registered Faspex users with the **Create packages from remote sources** option enabled can use storage locations as the source or destination of packages.

Setting a default storage location is required to perform basic transfers.

1. In the Admin app, go to **Nodes and storage** and select a node.

2. Go to the **Storage locations** tab.
3. Click **Create storage location**.
4. Enter a name for the file storage.
5. If your node supports it, you can link package files to source files instead of copying source files.

   If the node is running a Linux operating system and you want to enable symbolic links for this file storage, select **Enable linking**. This setting is ignored if the option is not supported by the node (in other words, non-Linux nodes).

   If you are using this file storage as cloud storage, select **Enable cloud referencing**.

   For more information on either option, see "Link package files to source files instead of copying source files" on page 78.
6. Choose the directory for the file storage. Click **Browse**, select a directory, and click **Select**.

   Important: You can browse only within the docroot associated with your Node API credentials. The path / means the docroot, not the root directory of the node.
7. Click **Create**.

Link package files to source files instead of copying source files

Symbolic links and cloud referencing are both features that links package files to source files instead of copying source files to create a package.

The symbolic links and cloud referencing features require that both the package source and the destination inbox are on the same node. This means these features only apply when a user sends a package with content from a shared folder on the same node as the destination inbox.

1. In the Admin app, go to **Nodes and storage** and select a node.
2. Go to the **Storage locations** tab.
3. Select a storage location.
4. Enable linking depending on your node type:

   - **Cloud storage**: Enable **Enable cloud referencing**. The source and destination of a package must be in the same cloud storage attached to an HST Server running in the cloud, or attached to the Aspera on Cloud Transfer Service.

     You must also enable cloud referencing on the HSTS node running on cloud storage. SSH into your machine and edit the `/opt/aspera/etc/trapd/`*`cloud_storage`*`.properties` file:

     ```
     aspera.session.support.symlink = true
     ```

     Enable the configuration changes by running:

     ```
     sudo service asperatrapd restart
     ```

     For a full list of supported cloud storage options, see the IBM Aspera High-Speed Transfer Admin Guide.
   - **Linux**: Enable **Enable symbolic links**. Your node must be running a Linux operating system. This setting is disabled if the option is not supported by the node (in other words, non-Linux nodes).
5. Enable specific users to create packages from remote sources.

   Go to **Users > All users** and click the name of the user. Under the Permissions section, select **Send packages from a remote source** to enable the feature for that user.

Setting a default storage location

Whenever a user uploads content to send a new package, the user uploads that content to the server's default storage location. Setting a default storage location is required to perform basic transfers.

To change the default storage location:

1. In the Admin app, go to **Nodes and storage**.
2. Select the node with the storage location you want to set as the server default.
3. Right-click the storage location and select **Make default inbox** from the overflow menu.

Making a storage location available as a content source

Change a storage location's read permissions to public to make it available for Faspex users to use it as a content source. Any Faspex user with the **Send packages from a remote sources** option enabled in their accounts can send packages using this storage location as a content source.

Permissions:

- **Private**: No one can use this storage location as a remote source.
- **Public**: Any user with the **Send packages from a remote source** permission can use this storage location as a remote source.
- **Limited**: Set a list of users who can use this storage location as a remote source. Users must have the **Send packages from a remote source** permission to use the file storage as a source.

1. In the Admin app, go to **Nodes and storage** and select a node.
2. Go to the **Storage locations** tab.
3. Select a storage location.
4. Go to the **Read permissions** tab and set Read permission to **Public**.
5. Click **Save**.

Configuring a Linux node for Faspex

A *node* is any server running IBM Aspera High-Speed Transfer Server (HSTS). Aspera web applications, such as Faspex, communicate with a node through the IBM Aspera Node API.

The instructions below assume you have already installed HSTS 4.3+ on your server. For instructions on installing *IBM Aspera High-Speed Transfer Server Admin Guide: Installing HSTS*.

1. Set up the node as the `root` user. If you do not have access to the `root` user, you must give the current system user permissions to make changes to the `/opt/aspera/etc/aspera.conf` configuration file.

   Change ownership of the `aspera.conf` file to the current system user:

   ```
   chown system_user:root /opt/aspera/etc/aspera.conf
   ```

2. Verify that the node is running HSTS with a valid Connect Server license on your transfer server:

   Run the following command:

   ```
   ascp -A
   ```

   In the resulting output, look for the following phrase:

   ```
   Connect Server License max rate
   ```

   If you need to update your transfer server license, follow the instructions in *IBM Aspera High-Speed Transfer Server Admin Guide: Updating Product License.*

3. Create the `faspex` system user account on the node.

   Run the following commands to create the system user faspex.

   ```
   groupadd -r faspex
   useradd -r faspex -g faspex
   ```

4. Set up `aspshell` as the default shell for the transfer user.

   ```
   usermod -s /bin/aspshell faspex
   ```

5. Create and configure the `faspex_packages` directory.

   Run the following commands to create the `faspex_packages` directories and configure the `faspex` user directories:

```
mkdir -p /home/faspex/faspex_packages
chown faspex:faspex /home/faspex/
chown faspex:faspex /home/faspex/faspex_packages
```

The **`asconfigurator`** utility modifies the `aspera.conf` configuration file, located at: `/opt/aspera/etc/aspera.conf`.

6. Add the user to `aspera.conf` and set the *docroot*.

   The directory you choose for the docroot is the absolute path for the transfer user. When this node is added to Faspex, users cannot access files or folders outside of the docroot.

   CAUTION: Do not use spaces in your docroot. If your docroot contains spaces, you may not receive all email notifications relating to transfer activity.

   Run the following **`asconfigurator`** command with the transfer username and the docroot path:

```
asconfigurator -x "set_user_data;user_name,username;absolute,/docroot/path"
```

   For example:

```
asconfigurator -x "set_user_data;user_name,faspex;absolute,/home/faspex/faspex_packages"
```

7. Set up token authorization for the user in `aspera.conf`.

   a) Run the following **asconfigurator** commands to set the encryption key for the user:

```
#
asconfigurator -x
"set_user_data;user_name,username;authorization_transfer_in_value,token"

#
asconfigurator -x
"set_user_data;user_name,username;authorization_transfer_out_value,token"
```

   For example:

```
#
asconfigurator -x "set_user_data;user_name,faspex;authorization_transfer_in_value,token"

#
asconfigurator -x "set_user_data;user_name,faspex;authorization_transfer_out_value,token"
```

   b) Configure dynamic key generation:

```
sudo asconfigurator -x "set_node_data;token_dynamic_key,true"
```

   c) Set a Redis primary key using **`askmscli`**. The master key must be a unique random 256-bit key. The example below uses **`openssl`** to generate the key. This Redis primary key will be used to encrypt the dynamic token encryption key.

```
echo -n "$(openssl rand -base64 32)" | sudo askmscli -s redis-primary-key
```

   d) Initialize the transfer user's keystore:

```
sudo askmscli -i -u username
```

   e) Set the store for the `asperadaemon` user that runs **`asperanoded`**:

```
sudo askmscli -i -u asperadaemon
```

   For more information, see the *IBM Aspera High-Speed Transfer Server:Secrets Management with askmscli* section.

8. Set the IP address or hostname for the node in the `aspera.conf` file with the following **`asconfigurator`** command:

   ```
   asconfigurator -x "set_server_data;server_name,ip_or_hostname"
   ```

   For example:

   ```
   asconfigurator -x "set_server_data;server_name,aspera.example.com"
   ```

9. Configure the node for HTTP and HTTPS fallback.

   HTTP fallback serves as a backup transfer method when Aspera FASP transfers (UDP port 33001, by default) is unavailable. When HTTP fallback is enabled and UDP connectivity is lost or cannot be established, the transfer continues over HTTP or HTTPS. By default, Faspex requires you to enable HTTP and HTTPS and use the ports 8080 and 8443, respectively:

   ```
   asconfigurator -x "set_http_server_data;enable_http,true"
   asconfigurator -x "set_http_server_data;http_port,8080"
   asconfigurator -x "set_http_server_data;enable_https,true"
   asconfigurator -x "set_http_server_data;https_port,8443"
   ```

   Restart the **asperahttpd** service.

   ```
   service asperahttpd restart
   ```

   Or on an OS running `systemd`:

   ```
   systemctl restart asperahttpd
   ```

10. Enable activity logging on the node:

    ```
    asconfigurator -x "set_server_data;activity_logging,true"
    ```

    If you do not enable activity logging, Faspex cannot retrieve package information and your users cannot download packages.

11. You must restart the `asperanoded` service for changes made using the **`asconfigurator`** utility (which modifies the `aspera.conf` configuration file) to take effect:

    ```
    service asperanoded restart
    ```

    Or on an OS running `systemd`:

    ```
    systemctl restart asperanoded
    ```

12. Configure a HSTS transfer user account with a Node API username and password.

    Faspex communicates to the HSTS transfer user account through the Node API to start transfers on the node.

    For instructions on adding users to HSTS, see the *IBM Aspera High-Speed Transfer Server Admin Guide: Setting Up Users*.

    a) Set up the Node API user:

    ```
    /opt/aspera/bin/asnodeadmin -a -u node_api_username -p node_api_passwd -x system_username
    ```

    Note: Use different names for the system user account and transfer user account in order to minimize confusion when tracing transactions and events.

    For example:

    ```
    /opt/aspera/bin/asnodeadmin -a -u node_user -p XF324cd28 -x faspex
    ```

    b) Run the following command to check the system user was successfully added to **`asnodeadmin`**:

    ```
    /opt/aspera/bin/asnodeadmin -l
    ```

Given a node user named `node_user` and a system user named `faspex`, the output should be:

```
                user       system/transfer user                    acls
====================    =======================    ====================
           node_user                  faspex    []
```

13. Copy the IBM Aspera Connect public key to `authorized_keys` to allow Connect to connect to Faspex.

    a) If the `.ssh` folder does not already exist in the `faspex` system user's home directory, run the following command to create the folder:

    ```
    mkdir -p /home/username/.ssh
    ```

    For example:

    ```
    mkdir -p /home/faspex/.ssh
    ```

    b) If the `authorized_keys` file does not already exist, add the `aspera_tokenauth_id_rsa.pub` public key to the file by running:

    ```
    cat /opt/aspera/var/aspera_tokenauth_id_rsa.pub >> /home/username/.ssh/authorized_keys
    ```

    For example:

    ```
    cat /opt/aspera/var/aspera_tokenauth_id_rsa.pub >> /home/faspex/.ssh/authorized_keys
    ```

    c) Transfer the `.ssh` folder and `authorized_keys` file ownership to the system user by running the following commands:

    ```
    chown -R username:username /home/username/.ssh
    chmod 600 /home/username/.ssh/authorized_keys
    chmod 700 /home/username
    chmod 700 /home/username/.ssh
    ```

    For example:

    ```
    chown -R faspex:faspex /home/faspex/.ssh
    chmod 600 /home/faspex/.ssh/authorized_keys
    chmod 700 /home/faspex
    chmod 700 /home/faspex/.ssh
    ```

You can now add this server to Faspex as a node.

Configuring a Windows node for Faspex

A *node* is any server running IBM Aspera High-Speed Transfer Server (HSTS). Aspera web applications, such as Faspex, communicate with a node through the IBM Aspera Node API.

The instructions below assume you have already installed HSTS 4.3+ on your server. For instructions on installing *IBM Aspera High-Speed Transfer Server Admin Guide: Installing HSTS*.

1. Verify that the node is running HSTS with a valid Connect Server license on your transfer server:

   Run the following command:

   ```
   ascp -A
   ```

   In the resulting output, look for the following phrase:

   ```
   Connect Server License max rate
   ```

   If you need to update your transfer server license, follow the instructions in *IBM Aspera High-Speed Transfer Server Admin Guide: Updating Product License.*

2. Create the `faspex` system user account on the node.

Click **Control Panel User Accounts** and add a new account named `faspex`. This system user account is associated with the Node API account in the steps below.

After creating a Windows user account, log in as that user at least once for Windows to set up the user's home folder.

3. Create the `faspex_packages` directory.

```
cd C:\
mkdir faspex_packages
```

The **`asconfigurator`** utility modifies the `aspera.conf` configuration file, located at: `C:\Program Files (x86)\Aspera\Enterprise Server\etc\aspera.conf`.

4. Add the transfer user to `aspera.conf` and set the *docroot*.

   The directory you choose for the docroot is the absolute path for the transfer user. When this node is added to Faspex, users cannot access files or folders outside of the docroot.

   CAUTION: Do not use spaces in your docroot. If your docroot contains spaces, you may not receive all email notifications relating to transfer activity.

   Run the following **`asconfigurator`** command with the transfer username and the docroot path:

```
asconfigurator -x "set_user_data;user_name,username;absolute,\docroot\path"
```

   For example:

```
asconfigurator -x "set_user_data;user_name,faspex;absolute,\faspex_packages"
```

5. Set up token authorization for the user in `aspera.conf`.

   a) Run the following **asconfigurator** commands to set the encryption key for the user:

```
#
asconfigurator -x
"set_user_data;user_name,username;authorization_transfer_in_value,token"

#
asconfigurator -x
"set_user_data;user_name,username;authorization_transfer_out_value,token"
```

   For example:

```
#
asconfigurator -x "set_user_data;user_name,faspex;authorization_transfer_in_value,token"

#
asconfigurator -x "set_user_data;user_name,faspex;authorization_transfer_out_value,token"
```

   b) Configure dynamic key generation:

```
asconfigurator -x "set_node_data;token_dynamic_key,true"
```

   c) Set a Redis master key using **`askmscli`**. The master key must be a unique random 256-bit key. The example below uses **`openssl`** to generate the key. This Redis master key will be used to encrypt the dynamic token encryption key.

```
#
openssl rand -base64 32 > redis_primary_key_file

#
type redis_primary_key_file | askmscli.exe -s redis-primary-key
```

   d) Initialize the transfer user's keystore:

```
askmscli.exe -u username -i -L- -DD
```

   e) Set the store for the user that runs **asperanoded**. The default user is `svcAspera`. The username might change if you are not using the default user:

```
askmscli.exe -i -u svcAspera
```

For more information, see the *IBM Aspera High-Speed Transfer Server:Secrets Management with askmscli* section.

6. Set the IP address or hostname for the node in the `aspera.conf` file with the following **`asconfigurator`** command:

   ```
   asconfigurator -x "set_server_data;server_name,ip_or_hostname"
   ```

   For example:

   ```
   asconfigurator -x "set_server_data;server_name,aspera.example.com"
   ```

7. Configure the node for HTTP and HTTPS fallback.

   HTTP fallback serves as a backup transfer method when Aspera FASP transfers (UDP port 33001, by default) is unavailable. When HTTP fallback is enabled and UDP connectivity is lost or cannot be established, the transfer continues over HTTP or HTTPS. Faspex requires you to enable HTTP and HTTPS and use the ports 8080 and 8443, respectively:

   ```
   asconfigurator -x "set_http_server_data;enable_http,true"
   asconfigurator -x "set_http_server_data;http_port,8080"
   asconfigurator -x "set_http_server_data;enable_https,true"
   asconfigurator -x "set_http_server_data;https_port,8443"
   ```

   Restart the `asperahttpd` service. Go to **Control Panel Administrative Tools Computer Management Services and Applications Services**, click **IBM Aspera HTTPD**, and click **Restart**.

8. Enable activity logging on the node:

   ```
   asconfigurator -x "set_server_data;activity_logging,true"
   ```

   If you do not enable activity logging, Faspex cannot retrieve package information and your users cannot download packages.

9. You must restart the `asperanoded` service for changes made using the **`asconfigurator`** utility (which modifies the `aspera.conf` configuration file) to take effect. Go to **Control Panel > Administrative Tools > Computer Management Services and Applications Services**, click **IBM Aspera NodeD**, and click **Restart**.

10. Configure a HSTS transfer user account with a Node API username and password.

    Faspex communicates to the HSTS transfer user account through the Node API to start transfers on the node.

    For instructions on adding users to HSTS, see the *IBM Aspera High-Speed Transfer Server Admin Guide: Setting Up Users*.

    a) Set up the Node API user:

    ```
    asnodeadmin -a -u node_api_username -p node_api_passwd -x system_username
    ```

    Note: Use different names for the system user account and transfer user account in order to minimize confusion when tracing transactions and events.

    For example:

    ```
    asnodeadmin -a -u node_user -p XF324cd28 -x faspex
    ```

    b) Run the following command to check that the system user was successfully added to **`asnodeadmin`**:

    ```
    asnodeadmin -l
    ```

Given a node user named node_user and a system user named faspex, the output should be:

```
                user      system/transfer user                   acls
====================    =======================    ====================
           node_user                   faspex    [ ]
```

11. Copy the IBM Aspera Connect public key to authorized_keys to allow Connect to connect to Faspex.
    a) If the .ssh folder does not already exist in the system user's home directory, run the following commands to create the folder:

    ```
    cd "C:\Documents and Settings\username"
    mkdir .ssh
    ```

    b) If the authorized_keys file does not already exist, use a text editor to create or edit the following file: C:\Documents and Settings*username*\.ssh\authorized_keys.

    c) Copy the contents of the aspera_tokenauth_id_rsa.pub (C:\Program Files (x86)\Aspera\Enterprise Server\var\aspera_tokenauth_id_rsa.pub) public key to the file.

    Note: When you copy the contents inside *aspera_tokenauth_id_rsa.pub* remove /bin/ from the command="/bin/aspshell -t" to make it be command="aspshell -t",...., or use the full path command="\"C:\Program Files\Aspera\Enterprise Server\bin\aspshell.exe\" -t",...

    The file must be named "authorized_keys" without file extensions. Some text editors add a .txt extension to the filename automatically. Be sure to remove the extension if it was added to the filename.

You can now add this server to Faspex as a node.

Configuring a MacOS node for Faspex

A *node* is any server running IBM Aspera High-Speed Transfer Server (HSTS). Aspera web applications, such as Faspex, communicate with a node through the IBM Aspera Node API.

The instructions below assume you have already installed HSTS 4.3+ on your server. For instructions on installing *IBM Aspera High-Speed Transfer Server Admin Guide: Installing HSTS*.

1. Verify that the node is running HSTS with a valid Connect Server license on your transfer server:

   Run the following command:

   ```
   ascp -A
   ```

   In the resulting output, look for the following phrase:

   ```
   Connect Server License max rate
   ```

   If you need to update your transfer server license, follow the instructions in *IBM Aspera High-Speed Transfer Server Admin Guide: Updating Product License.*
2. Create the faspex system admin account on the node.
   a) Go to **System Preferences Users & Groups**.
   b) Click the lock button and enter your admin credentials to make changes.
   c) Click the add button.
   d) Name the user faspex.
   e) Select **Administrator** from the New Account drop-down menu.
   f) Name the account.
   g) Enter and verify a password for the account.
   h) Click **Create User**.

i) Click **Login Options** in the users panel.

j) Click the **Join** button next to Network Account Server.

k) Click **Open Directory Utility**.

l) In the Directory Utility window, click the lock button and enter an administrator account and password to make changes.

m) From the menu bar, select **Edit Enable Root User**.

n) Enter and verify the password.

o) Click **OK**.

The **asconfigurator** utility modifies the `aspera.conf` configuration file, located at: `/Library/Aspera/etc/aspera.conf`.

3. Add the user to `aspera.conf` and set the *docroot*.

   The directory you choose for the docroot is the absolute path for the transfer user. When this node is added to Faspex, users cannot access files or folders outside of the docroot.

   CAUTION: Do not use spaces in your docroot. If your docroot contains spaces, you may not receive all email notifications relating to transfer activity.

   Run the following **asconfigurator** command with the transfer username and the docroot path:

   ```
   asconfigurator -x "set_user_data;user_name,username;absolute,/docroot/path"
   ```

   For example:

   ```
   asconfigurator -x "set_user_data;user_name,faspex;absolute,/project1"
   ```

4. Set up token authorization for the user in `aspera.conf`.

   a) Configure dynamic key generation:

   ```
   sudo asconfigurator -x "set_node_data;token_dynamic_key,true"
   ```

   b) Set a Redis primary key using **askmscli**. The master key must be a unique random 256-bit key. The example below uses **openssl** to generate the key. This Redis primary key will be used to encrypt the dynamic token encryption key.

   ```
   echo -n "$(openssl rand -base64 32)" | sudo askmscli -s redis-primary-key
   ```

   c) Initialize the transfer user's keystore:

   ```
   sudo askmscli -i -u username
   ```

   d) Set the store for the `asperadaemon` user that runs **asperanoded**:

   ```
   sudo askmscli -i -u asperadaemon
   ```

   For more information, see the *IBM Aspera High-Speed Transfer Server:Secrets Management with askmscli* section.

5. Set the IP address or hostname for the node in the `aspera.conf` file with the following **asconfigurator** command:

   ```
   asconfigurator -x "set_server_data;server_name,ip_or_hostname"
   ```

   For example:

   ```
   asconfigurator -x "set_server_data;server_name,aspera.example.com"
   ```

6. Configure the node for HTTP and HTTPS fallback.

   HTTP fallback serves as a backup transfer method when Aspera FASP transfers (UDP port 33001, by default) is unavailable. When HTTP fallback is enabled and UDP connectivity is lost or cannot be established, the transfer continues over HTTP or HTTPS. Faspex requires you to enable HTTP and HTTPS and use the ports `8080` and `8443`, respectively:

```
asconfigurator -x "set_http_server_data;enable_http,true"
asconfigurator -x "set_http_server_data;http_port,8080"
asconfigurator -x "set_http_server_data;enable_https,true"
asconfigurator -x "set_http_server_data;https_port,8443"
```

Restart the **asperahttpd** service by running the following commands:

```
sudo launchctl stop com.aspera.asperahttpd
sudo launchctl start com.aspera.asperahttpd
```

7. Enable activity logging on the node:

```
asconfigurator -x "set_server_data;activity_logging,true"
```

If you do not enable activity logging, Faspex cannot retrieve package information and your users cannot download packages.

8. You must restart the `asperanoded` service for changes made using the **`asconfigurator`** utility (which modifies the `aspera.conf` configuration file) to take effect:

```
sudo launchctl stop com.aspera.asperanoded
sudo launchctl start com.aspera.asperanoded
```

9. Configure a HSTS transfer user account with a Node API username and password.

Faspex communicates to the HSTS transfer user account through the Node API to start transfers on the node.

For instructions on adding users to HSTS, see the *IBM Aspera High-Speed Transfer Server Admin Guide: Setting Up Users*.

a) Set up the Node API user:

```
/Library/Aspera/bin/asnodeadmin -a -u node_api_username -p node_api_passwd -x
system_username
```

Note: Use different names for the system user account and transfer user account in order to minimize confusion when tracing transactions and events.

For example:

```
/Library/Aspera/bin/asnodeadmin -a -u node_user -p node_user -p XF324cd28 -x faspex
```

b) Run the following command to check the system user was successfully added to **`asnodeadmin`**:

```
/Library/Aspera/bin/asnodeadmin -l
```

Given a node user named `node_user` and a system user named `faspex`, the output should be:

```
                user       system/transfer user                    acls
====================    =======================    ====================
           node_user                     faspex    []
```

10. Copy the IBM Aspera Connect public key to `authorized_keys` to allow Connect to connect to Faspex.

a) If the `.ssh` folder does not already exist in the system user's home directory, run the following commands to create the folder:

```
mkdir "/Users/username/.ssh"
```

For example:

```
mkdir "/Users/faspex/.ssh"
```

b) If the `authorized_keys` file does not already exist, use a text editor to create or edit the following file: `/Users/`*`username`*`/.ssh/authorized_keys`.

c) Copy the contents of the `aspera_tokenauth_id_rsa.pub` (`/Library/Aspera/Enterprise Server/var/aspera_tokenauth_id_rsa.pub`) public key to the file.

The file must be named `authorized_keys` without file extensions. Some text editors add a `.txt` extension to the filename automatically. Be sure to remove the extension if it was added to the filename.

You can now add this server to Faspex as a node.

Managing shared inboxes

Shared inboxes provide a file submission system for users to share packages. Shared inbox admins can also invite external users (people who don't have a Faspex account) to send to a shared inbox.

The topics in this section assume that you are an admin, manager, or shared inbox admin. Shared inbox admins can expand access beyond shared inbox members by allowing anyone with a shared inbox-specific public submission link to send packages to the shared inbox.

Common uses

Use several shared inboxes to manage assets for different projects or business processes and require different package metadata for each inbox

Allow external users to send packages without giving them full access to Faspex or requiring them to log in.

Create a shared inbox

1. Go to the **Packages** app from the app switcher.
2. Go to **Manage shared inboxes**.
3. Click **Create new**.
4. Name the shared inbox.
5. Click the **Save** button.
6. Select the newly created shared inbox.
7. Go to the **Members** tab.
8. To add Faspex users:

   a) Click **Add member**.

   b) Search for and select the Faspex users you want to add.

   c) Select the shared inbox permissions for the users.

   d) Click **Add**.

Manage shared inbox members and SAML groups

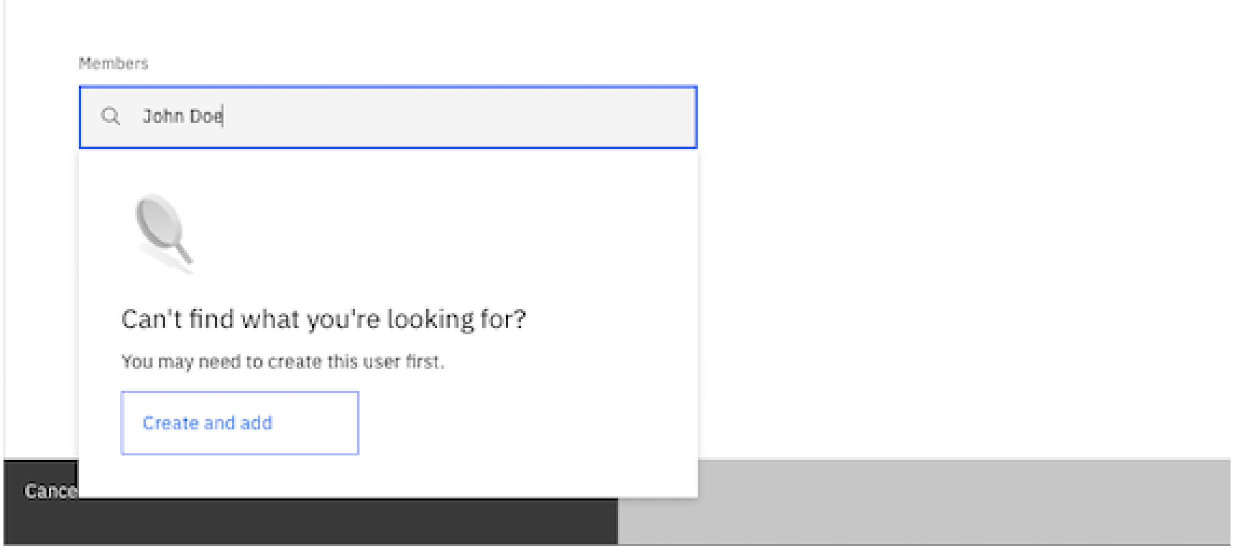
You can add Faspex users and SAML groups to a shared inbox. When a user sends a package to a shared inbox, all shared inbox members can access the package in the shared inbox (under Received).

After selecting a shared inbox, go to the **Members** tab to manage shared inbox members.

Add members

1. Click **Add member**.
2. Search for and select the Faspex users you want to add.

   If you are a shared inbox admin and the shared inbox permissions allow you to create and add new users, you can click **Create and add** to create a new user and add the user to the shared inbox.


Members

John Doe

Can't find what you're looking for?

You may need to create this user first.

3. Choose the access permission to grant to these users.
   - **Standard**
   - **Submit only**
   - **Shared inbox admin**
4. Click **Add**.

Add SAML groups

1. Select the **SAML groups** toggle.
2. Click **Add group**.
3. Select the group to add.
4. Click **Add**.

Invite external users to send packages to a shared inbox through an email

If you are a member of a shared inbox that allows regular users to invite members or you are a shared inbox admin, you can send an invitation email to allow someone to send a package to a shared inbox without registering an account or logging in.

Anyone with the URL can send packages to the shared inbox, but cannot see or download shared inbox packages.

Faspex cannot verify the person using the link is the intended collaborator. If this is a concern, set a custom link expiration policy for the invitation link.

1. Go to **Manage shared inboxes** in the sidebar.
2. Select a shared inbox and go to the **Members** tab.
3. Click **Invite with a link**.
4. Enter the email address of the recipient.

   You can send an invitation email to any user, including Faspex users.
5. If allowed, configure security settings. Invitation-link security settings are configured by admins at the server level.
6. Click **Send**.

Enable public submission links for a shared inbox

Enable a shared inbox's public submission link to allow anyone with the link to send a package to the shared inbox without logging in to Faspex.

1. Go to **Manage shared inboxes** and select a shared inbox.
2. Go to the **Submission link** tab.

   If you do not see the **Submission link** tab, Faspex does not allow shared inbox admins to enable or disable the link.
3. Click **Save**.

Configuring shared inbox admin settings

Faspex admins can designate members as shared inbox admins. Shared inbox admins have additional permissions configured per shared inbox. All Faspex admins are shared inbox admins of every shared inbox.

To manage shared inbox admin permissions, go to the **Settings** tab of any shared inbox. There, you can enable shared inbox admins to:

- **Add existing Faspex users and remove non-admin members**
- **Invite and remove outside submitters**
- **Create, remove and delete new Faspex users**

  ⚠ **Warning:** This permission allows shared inbox admins to remove Faspex users globally and permanently from the system.
- **Add and remove SAML groups**

Note: All Faspex admins have all these permissions for every shared inbox.

Set the package expiration policy for a shared inbox

Set a package expiration policy.

Setting a package expiration policy overrides the global package expiration setting. If global package expiration is enabled, but you want to disable time-based and download-based package expiration for only this shared inbox, select **None** for downloads-based package expiration and clear **Time-based policy**.

To set a package expiration policy:

1. Go to the **Settings** tab of any shared inbox.
2. Toggle **Set package expiration policy**.
3. Select an option for downloads-based package expiration.

   You can choose these download-based policies:

Option	Description
None	Do not delete the submitted package after it is downloaded.
Delete files after any recipient downloads all files	Delete the package if *any* member downloads all the files in the package. **Note:** When using this policy, forwarding the package to another recipient can prevent the original recipient from downloading the package. If a package is forwarded to and downloaded by another recipient, Faspex deletes the package

Option	Description
	contents and the original recipient cannot download the package.
Delete files after all recipients download all files	Delete the package if *all* members have downloaded all the files in the package.

4. To set a time-based policy, select **Time-based policy** and enter the number of days the package is available before deleting the package.
5. Click **Save**.

Set the link expiration policy for a shared inbox

Set an invitation link expiration policy. Invitation links allow anyone with the link to send package to the shared inbox.

Setting an invitation link expiration policy overrides the global invitation link expiration setting. If global invitation link expiration is enabled, but you want to disable time-based and sending-based link expiration for only this shared inbox, clear **Expire link after a collaborator successfully submits a package** for downloads-based package expiration and clear **Time-based policy**.

To set an invitation link expiration policy:

1. Go to **Manage shared inboxes**.
2. Select the shared inbox.
3. Go to the **Settings** tab.
4. Toggle **Set invitation expiration policy**.
5. To set sending-based link expiration, select **Expire link after a collaborator successfully submits a package**.

   Note: While the link expires after a collaborator successfully sends a package, it is possible for a collaborator to initiate parallel uploads using a single link to submit multiple packages.
6. To set a time-based policy, select **Time-based policy** and enter the number of days the link stays valid.
7. Click **Save**.

Assigning a metadata profile to a shared inbox

You can apply a metadata profile to a shared inbox. Metadata profiles define additional optional and required fields for users to fill when sending a package.

For instructions on creating a metadata profile, see "Create a metadata profile" on page 106.

1. Go to the **Packages** app from the app switcher.
2. Go to **Manage shared inboxes**.
3. Select the desired shared inbox.
4. Go to **Metadata**.
5. Select a metadata profile from the **Metadata profile** drop-down menu. If the only option is **None**, there are no metadata profiles created.
6. To save the metadata of sent packages to a file in the package, select **Save metadata to file**.

   The metadata is saved to the package's root directory as `aspera-metadata.xml`.
7. Click **Save**.

Setting a custom destination for a shared inbox

Choose your shared inbox destination inbox. Packages sent to the shared inbox are stored at this location. Only Faspex admins can set a custom inbox; shared inbox admins do not have this ability.

When using a custom inbox, packages sent to the shared inbox are first uploaded to the server-default storage location, and then relayed to the specified storage location. When packages are deleted from the default location, they are not automatically removed from the custom location.

Note: When symbolic links are enabled for a storage location, packages in the custom inbox location are stored as actual files, not symbolic links. The default inbox location contains symbolic links, but the custom inbox contains actual files.

1. Select a shared inbox and go to the **Nodes and storage** tab.
2. Under inbox destination, select **Custom**.
3. To upload directly to the specified storage location instead of first uploading to the server-default storage location, select **Upload directly to custom inbox**.
4. Select a node from the table and select the storage location you want to use as the custom inbox.
5. Click **Save**.

Forwarding shared inbox packages to another storage location

Forward packages sent to a shared inbox to another storage location using relays.

When relays are configured, packages sent to the shared inbox are first uploaded to the server-default storage location, and then relayed to the specified storage locations. When packages are deleted from the default location, they are not automatically removed from the custom location.

1. Select a shared inbox and go to the **Nodes and storage** tab.
2. Select **Relay**.
3. Select **Enable relays**.
4. Click **Save**.
5. For each storage location you want to relay packages to:
   a) Select the storage location.
   b) Enable **Forward to this relay**.
   c) If you want to overwrite files if they exist on the destination, enable **Overwrite files**.
   d) If you want to notify users when a relay starts, completes, or errors out, enter contacts in the notification fields.
   e) Click **Save**.

Configuring transfer options

By default, Faspex uses the transfer settings from the Aspera Central Server section. Select **Override default settings** to set user-specific transfer settings, which take precedence over the server-wide settings.

Aspera Connect client settings

Field	Description
Override Connect installation URL	The default installation URL is on IBM Cloudfront. For users who cannot access the internet or download software, configure an accessible URL from which users can download and install Aspera Connect.
Override Connect minimum version	The default minimum version is always the latest released version. To suppress the system prompt to upgrade, configure an alternative minimum

Field	Description
	version of Aspera Connect that can be used to transfer with Faspex. The version must be in the form "X.Y.Z" (for example, 0.0.0.).
Communicate with Connect over HTTP	[Chrome only] Force the browser to use HTTP to communicate with the Aspera Connect client instead of using the browser extension.

Downloads during transfers

Field	Description
Enable downloads during transfers	When enabled, users can download files from packages in an ongoing transfer.
Enable HTTP fallback	When enabled, HTTP fallback serves as a secondary transfer method when the Internet connectivity required for Aspera FASP transfers is unavailable. When HTTP fallback is enabled and UDP connectivity is lost or cannot be established, the transfer will continue over the HTTP protocol. HTTP fallback is not supported if you are using HTTP gateway. **Important:** HTTP fallback requires HSTS 4.4.3 or later.

Uploads

Field	Description
Timeout rate	Define how long the system waits before the package upload operation times out.

Default transfer rates

Field	Description
Upload rate	Specify the target transfer rate for user-to-server transfers.
Download rate	Specify the target transfer rate for server-to-user transfers.
Lock minimum rate and policy	Prevent clients from adjusting the transfer policy or minimum transfer rate.

Default maximum allowed rate

Field	Description
Upload rate	Specify the maximum transfer rate for user-to-server transfers.
Download rate	Specify the maximum transfer rate for server-to-user transfers.

Outgoing bandwidth

Field	Description
Server-to-server relay transfer settings	Set the outgoing bandwidth for local server to remote server transfers.

Configuring the web server

In the admin app, go to **Configurations > Web Server**.

Field	Description
Server's external address or name	The Faspex server's primary IP address or domain name. Faspex uses this field to generate correct links in email notifications and to generate correct branding links. If you don't properly configure the server address, you may experience: • Broken links in emails • Missing custom images in areas with branding, such as the log in page.
HTTP port	The HTTP port number to use for HTTP fallback.
HTTPS port	The HTTPS port number to use for HTTP fallback.

Configuring alternate addresses

Configure alternate addresses for groups of users that should access Faspex from an alternate IP address or domain name.

1. Verify with your IT department that the alternate address resolves to your Faspex server's hostname in your DNS.
2. Go to **Configurations > Web server**.
3. Select **Enable alternate address** and click **Add address**.
4. Enter an address name.

   You can specify multiple subnets or a specific range of addresses. For example:

   ```
   198.51.100.24,192.168.0.0/18,10.0.0.*
   ```

   Note: Alternate addresses support comma-delimited Classless Inter-Domain Routing (CIDR).
5. Enter description to include in email notifications.
6. Enable **Show in emails** to make this alternate address available as a variable to include in email notification templates.

   This variable goes by `ALTERNATE_ADDRESS_#`.
7. Click **Update** to finish.

Configure display settings

You can upload a custom logo and background photo and configure login page instructions.

Go to **Configuration > Display settings**

Custom logo and background photo

Click the **Add file** button to replace the default logo in the menu bar with your custom logo.

Login page

You can configure the login page text using the **Login page header** and **Local login instructions** options. The header is the title of the login form and the instructions appear above the local login option.

Setting custom ports

Setting up custom ports in Faspex 5 involves configuring unique communication channels to facilitate the transfer of data.

1. Edit the `/opt/aspera/faspex/conf/docker/router.env` file to replace the default port values with the desired custom ports.

   ```
   FASPEX_ROUTER_HTTP_PORT=80
   FASPEX_ROUTER_HTTPS_PORT=443
   ```

2. Run `faspexctl setup`. When prompted for the URL, make sure to include the port. For example: `9.30.189.165:8443`
3. After the setup is complete, you can access the Faspex app in the browser with the new port: `https://[faspex-server-ip]:[https-port]`

Managing packages on the server

Configuring package expiration policies

Delete package contents based on time-based or download-based expiration policies.

Go to **Configurations > Package storage** to set server package expiration policies enforced for all users. When a package expires, Faspex deletes the contents of the original package.

To allow users to override the server setting and configure the expiration policy when sending a package, enable **Users can set package deletion policies on a package-by-package basis**.

Download-based policy

Configure whether Faspex deletes package contents after recipients have downloaded the package. In this context, downloaded means either downloaded the entire package or all the individual contents of the package in one operation. Downloading only some of the package contents does not expire the package.

You can choose these download-based policies:

Option	Description
None	Do not delete the submitted package after it is downloaded.
Delete files after any recipient downloads all files	Delete the package if *any* member downloads all the files in the package. **Note:** When using this policy, forwarding the package to another recipient can prevent the original recipient from downloading the package. If a package is forwarded to and downloaded by another recipient, Faspex deletes the package contents and the original recipient cannot download the package.
Delete files after all recipients download all files	Delete the package if *all* members have downloaded all the files in the package.

Time-based policy

Configure whether Faspex deletes package contents after a period of time. For example, if you configure this setting for 10 days, Faspex deletes package contents 10 days after the package becomes available to the intended recipients.

Deleting package content

As an admin, you can delete package content from the **Packages** page in the Admin application. Deleting package content does not remove the package metadata, just the files and folders stored on the HSTS node. Senders and recipients can still view the package from the **Full history** page.

You can delete package content by clicking the trash can icon by a package or by using the **Free up space** button, which you can use to delete all package content older than a specified number of days.

Free up space

Delete old packages to free up space on your server.

Managing global distribution lists

Users with access can send packages to a global distribution list, which is a list of email addresses and Faspex users. The items in the list are not validated until a user tries to send a package to the list. Admins can configure whether all users can see these lists or whether admins have to grant access to individual users.

Configure access to global distribution lists

By default, all users can access global distribution lists. You can disable access by going to **Security > Users** and disabling the **Users can see global distribution lists by default** option. If disabled, an admin must manually grant a user access to global distribution lists.

In **Security > Users**, you can also enable **Ignore invalid recipients** to allow users to send package to distribution lists including invalid recipients. If enabled, Faspex skips any invalid user and delivers the package to all valid recipients in the list.

To create a new global distribution list:

1. Go to **Configurations > Global distribution lists** and click **Create new**.
2. Name your distribution list
3. Add emails from contacts or import from a CSV file.

   Faspex reads the first line in the CSV file as user fields. The CSV file supports these fields: `email`, `name`.

   For example, to create two users:

   ```
   email, name
   admin@ibm.com, Admin
   user@example.com, User
   ```
4. Click **Create new**.

Configuring email notifications

Notify users about various Faspex events, such as resetting a password, receiving a package, and so on.

Connecting Faspex to an email server for notifications

Connect Faspex to a SMTP email server.

1. In the Admin application, go to **Configurations > Email configuration**.
2. Choose **Login** or **Open** authentication. If you choose **Login** authentication, you are required to enter login credentials for the SMTP server.
3. Enter your **SMTP mail Server** and its **Server Port**.
4. To enable TLS, select **Use TLS if available**.

Important: Faspex confirms whether the name in your TLS security certificate matches your mail server's configured address (fully qualified domain name or IP address). If it does not, Faspex displays an error.

5. Enter the domain of the SMTP server.
6. If you chose **login** authentication, enter your login credentials.
   - **User**: The email account that you are sending the notification from (be sure to include the domain).
   - **Password**: The password for the email account.
7. Configure email details:
   - **Faspex "From" name**: The "From" name that appears on Faspex-generated emails.
   - **Faspex "From" email**: The "From" email address that appears on Faspex-generated emails.
   - **Package Recipient "from"**: The "From" name that appears on Faspex-generated emails related to packages.

Option	Package email notification received from
Faspex	"Faspex"
Sender via Faspex	Sender's name and "via Faspex".
Sender	Sender's full name
Sender email	Sender's email

8. Click **Save**.
9. Test your SMTP server settings:
   a) Click **Test email**.
   b) Enter your email address.
   c) Click **Send**.

   You should receive a confirmation email titled "Email settings test".

   Note: If you are running into errors, follow the `faspex-service` container logs and send a test email to see if `faspexservice` reports errors sending an email:

```
docker logs --tail 100 -f faspex-service
```

Customizing email notification templates

Customize Faspex email notifications. You can edit a template by using the template editor or by editing plain HTML.

Faspex provides a template editor for each email notification type. For more customization, you can edit the email template in plain HTML.

The form editor and HTML editor modify two separate templates, but only one template can be used at a time. When you click **Save**, Faspex uses the template of the current editor. If you switch editors and click **Save**, you lose any unsaved changes made in the other editor.

Scenario	Steps	Form template saved?	HTML template saved?	Template used
Save only after editing both templates.	1. Make changes in the form editor. 2. Make changes in the HTML editor.	No	Yes	HTML template

Scenario	Steps	Form template saved?	HTML template saved?	Template used
	3. Click **Save**.			
Save each time you edit a template.	1. Make changes in the form editor. 2. Click **Save**. 3. Make changes in the HTML editor. 4. Click **Save**.	Yes	Yes	HTML template

1. In the Admin app, go to **Notifications > Customize email templates**.
2. Select the email template you want to modify.
3. Use either the template editor or the HTML editor.

   You can click **View and copy variables** to see a list of variables you can use in the email.
4. Click **Save** when done.

Securing Faspex

This section covers important security practices for your Faspex application.

Aspera security considerations

Faspex file transfers do not include anti-virus and malware scanning before or after the transfer. If your business security requirements include anti-virus and anti-malware scanning, you must implement a separate process for the scan before you transfer files using Faspex.

If your business security requirements include restrictions for uploads and transfers by file extension type, you must create your own policies in the High Speed Transfer Server `aspera.conf` file. See the aspera.conf - Filters to include and exclude files section for instructions.

Installing an SSL certificate from an authorized Certificate Authority

For a secure installation, you should replace the Faspex self-signed certificate. The key and certificate (or certificate bundle) must be provided by an authorized, globally trusted Certificate Authority (CA).

Note: Use a certificate included in the Mozilla CA bundle.

1. Back up the default self-signed certificate and private key.

   ```
   mv /opt/aspera/faspex/conf/nginx/cert.pem /opt/aspera/faspex/conf/nginx/cert.pem.bak
   mv /opt/aspera/faspex/conf/nginx/key.pem /opt/aspera/faspex/conf/nginx/key.pem.bak
   ```

2. Retrieve the `fullchain` and `private` key from your certificate authority. This will vary depending on the issuing certificate authority.

   ```
   ll
   total 20
   -rw-r--r--. 1 root root 5599 Aug 18 21:13 fullchain.pem
   -rw-------. 1 root root 1704 Aug 18 21:13 privkey.pem
   ```

3. Copy the CA-issued certificate bundle (`fullchain`.pem) and private key (`privkey`.pem) to the directory `faspex nginx`; overwrite the default `cert`.pem and `key`.pem files.

Note: `Faspex-router` expects the name of the certificate to be `cert.`pen and the private key to be named `key.pem`.

```
cp -iv /<PATH_TO>/fullchain.pem /opt/aspera/faspex/conf/nginx/cert.pem
cp -iv /<PATH_TO>/privkey.pem /opt/aspera/faspex/conf/nginx/key.pem
```

4. Modify and verify the ownership/group ownership is `aspweb:aspweb` on the `cert.pem` and `key.pem`.

```
chown aspweb:aspweb /opt/aspera/faspex/conf/nginx/cert.pem
chown aspweb:aspweb /opt/aspera/faspex/conf/nginx/key.pem
```

```
ls -alh /opt/aspera/faspex/conf/nginx/
total 44K
drwxrwx--- 3 aspweb aspweb  242 Mar 29 11:04 .
drwxrwx--- 8 aspweb aspweb  187 Mar 29 12:13 ..
-rw-r--r-- 1 aspweb aspweb  118 Jul  5  2022 assets.conf
-rw-r--r-- 1 aspweb aspweb 6.2K May 26  2022 cert.pem
drwxr-xr-x 3 aspweb aspweb  178 Mar 29 16:28 custom
-rw-r--r-- 1 aspweb aspweb  769 Jul  5  2022 dhparam.pem
-rw-r--r-- 1 aspweb aspweb 3.2K May 26  2022 key.pem
-rw-r--r-- 1 aspweb aspweb 3.9K Aug  9  2022 nginx.conf
-rw-r--r-- 1 aspweb aspweb 3.8K Jul  5  2022 nginx.conf.bk
-rw-r--r-- 1 aspweb aspweb 4.6K Mar 29 16:28 nginx.conf.copy
-rw-r--r-- 1 aspweb aspweb 3.9K Aug  9  2022 nginx.conf.template
-rw-r--r-- 1 aspweb aspweb 3.8K May 26  2022 nginx.conf.template.501GA
```

5. Restart the `faspex-router`.

```
faspexctl restart router
```

6. Verify that the router is healthy by checking the status.

```
faspexctl status
```

```

CONTAINER ID   IMAGE
COMMAND                  CREATED        STATUS
PORTS                                         NAMES
8009fffa1337   icr.io/ibmaspera-test/faspex-router:5.0.5_release-d3fea4d    "/bin/sh -c 'rm
-rf …"   22 hours ago   Up About a minute (healthy)   0.0.0.0:80->8080/tcp, 0.0.0.0:443-
>8443/tcp   faspex-router
249905dde790   icr.io/ibmaspera-test/faspex-service:5.0.5_release-58ded11   "/bin/sh
-c 'sed -i …"   22 hours ago   Up 18 hours
(healthy)                                                   faspex-service
c55a15c2b2f4   icr.io/ibmaspera-test/faspex-ui:5.0.5_release-b56da15        "/bin/sh
-c 'rm -rf …"   22 hours ago   Up 18 hours
(healthy)                                                   faspex-ui
2f7254b9f8d0   icr.io/ibmaspera/connect-deployer:4.2.5                      "/bin/sh
-c 'rm -rf …"   22 hours ago   Up 18 hours
(healthy)                                                   connect-deployer
9b448259470b   icr.io/ibmaspera-test/faspex-core:5.0.5_release-4c6192d      "/bin/sh
-c 'if [ ! …"   22 hours ago   Up 18 hours
(healthy)                                                   faspex-core
bca81ada3e35   icr.io/ibmaspera-test/faspex-utility:5.0.5_release-ad77b09   "/bin/sh
-c 'if [ ! …"   22 hours ago   Up 18 hours
(healthy)                                                   faspex-utility
6f49f8becd62   icr.io/ibmaspera-test/faspex-db:5.0.5_release-9f266f2        "/bin/sh
-c 'if [ ! …"   22 hours ago   Up 18 hours
(healthy)                                                   faspex-db
```

Installing an SSL certificate using Let's Encrypt

Let's Encrypt allows you to automatically generate and configure SSL certificates within the `faspex-router` container for Red Hat, CentOS, and Amazon Linux2 operating systems.

Important: You need to have previously configured a DNS. For Amazon Linux2 you can use the ROUTE 53 service. For more information on this, visit the AWS documentation.

1. Use the tool `dig` to verify that your domain has a public DNS record.

   To install `dig`, run this command: `yum -y install dig`.

For example, the output below indicates that `faspex5.demotest.com` has a valid CNAME record pointing to an AWS load balancer.

```
dig faspex5.demotest.com

;; QUESTION SECTION:
;faspex5.demotest.com.    IN    A

;; ANSWER SECTION:
faspex5.demotest.com.    60    IN    CNAME    faspex5-ha-lb-xxxxxxxxxxxx-
west-2.elb.amazonaws.com.
faspex5-ha-lb-xxxxxxxxxxxx-west-2.elb.amazonaws.com. 60 IN A 3x.xxx.xxx.xxx
faspex5-ha-lb-xxxxxxxxxxxx-west-2.elb.amazonaws.com. 60 IN A 3x.xxx.xxx.xxx
```

2. To generate the certificates Certbot will take care of the authentication and validation of the server.
3. Install the certificate.

```
faspexctl install_letsencrypt
```

Important: You may be prompted to manually download packages depending on the operating system. In the specific case of Red Hat and CentOS, if you encounter the `NOTICE: Unable to find epel-release package installed` message, run the following command to install the `epel-release` package:

```
yum install -y https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
```

After the package is downloaded Certbot will request the DNS of the server that will run Faspex 5. Once this request is complete, the server will have the certificates configured.

Masking file names in packages

For security reasons, you may want to mask the names of files in your packages so that the original file names are not visible in the Faspex UI or in logs.

Faspex performs obfuscation when:

- A user initiates a transfer through Connect.
- A user initiates a transfer through the HTTP Gateway.
- A user initiates a transfer from a remote source (file storage on tethered nodes).

If enabled, Faspex obfuscates the file names of all uploaded files. In the case of a directory file structure, Faspex also obfuscates the folder name, the names of all files within the folder, and the names and files of any nested directories.

Faspex does not mask the file extensions of files. For example, after masking, a file with the `.txt` extension may be named `Nqu7ORqTEC2R9GHK8ISFw.txt`.

Note: File name masking in Faspex is irreversible.

Configure Global File Name Obfuscation

Go to **Server > Content security**. In the Obfuscation section, you can set the global option to:

- **Always**
- **Never**
- **Optional**

If set to **Optional**, users can choose whether to obfuscate file names at package creation time.

File encryption

Use Faspex and IBM Aspera High-Speed Transfer Server together to encrypt files before they are transferred, encrypt files at the destination, and encrypt data transferred over the network.

Encryption Options

Option	Description	Use Case	Instructions
Client-Side Encryption-at-Rest (CSEAR)	CSEAR provides end-to-end encryption on uploaded packages. When enabled, Faspex requires users to set an encryption password when uploading packages using IBM Aspera Connect. Connect encrypts the files with that password and transfers the packages to Faspex. Encrypted files are given the `.aspera-env` extension. When a package recipient downloads these `.aspera-env` files, they must use the password to decrypt the files and access their contents. The sender must give the recipient the password.	Give the sender complete control over who has access to the data.	Go to **Security > Content security** and set **Use encryption-at-rest** to **Always**.
Server-Side Encryption-at-Rest (SSEAR)	SSEAR is not a Faspex feature, but a HSTS feature. When a user sends a package, the HSTS encrypts the transferred files at the destination using a password defined in the `aspera.conf` configuration file.	Protect data on untrusted storage (for example, cloud storage connected to HSTS).	See *IBM Aspera High-Speed Transfer Server Admin Guide: Server-Side Encryption-at-rest (EAR)*.
Encryption-in-transit	Encrypt transfers using an encryption standard.	Protect data transfer through an untrusted or insecure network.	Go to **Security > Content security** and enable **Encrypt transfers**. Choose the **FASP cipher** to use for encryption.

Require external users to create an account to download packages

By default, external users do not need to create an account to download a public package, but an admin can configure Faspex to require external users to create an account before downloading the package (see "Allowing public packages" on page 105).

For more information on external users, see "Allowing package submission from external users" on page 103.

1. Go to **Security > Advanced collaboration**.
2. Under **External user registration policy** enable **External user must register an IBM Aspera account to download packages**.
3. Click **Save**.

Keeping user directories private

You can globally prevent a Faspex user (even if they have permissions to send to all Faspex users) from being able to see the entire user directory when sending packages and when adding users to workgroups, shared inboxes, distribution lists, and so on.

Users can still see workgroups, shared inboxes, and distribution lists. You can override this setting on a user-by-user basis by editing their permissions.

Important: When the privacy setting is turned on (set to **Yes**), users who have been assigned the role of **Workgroup admin** can still view the entire list of Faspex users via the **Workgroups > Members** page.

Configuring invitation link expiration policies

Set expiration policies for invitation links.

1. Go to **Security > Advanced collaboration** and enable **Set invitation link expiration policy for both personal and shared inbox invitations**.
2. Set a link expiration policy.

   a) Select **Link expires after a collaborator successfully submits a package** to expire invitation links after invitees have successfully sent a package.

   b) Select **Time-based policy** to expire invitation links after a period of time. For example, if you configure this setting for 10 days, Faspex expires the link 10 days after Faspex sends the email invitation to the invitee.

   You can enable both policies. The link expires whenever either of the conditions are met.

You can enable both policies. The link expires whenever either of the conditions are met.

3. Click **Save**.

Enabling FIPS on the host OS

This section provides steps on how to set up Red Hat 8 to be FIPS compliant. This will automatically set up the containers to inherit the FIPS configuration from the OS.

Attention: Configure HSTS with an HTTPS reverse proxy that supports Transport layer security (TLS) 1.3.

Red Hat 8

To enable FIPS on Red Hat 8 follow the official documentation available at: Switching the system to FIPS mode.

Important: You must run the initial `faspexctl` setup with FIPS disabled. Enable FIPS on Red Hat 8 after the Faspex 5 installation is complete. Reboot your system after enabling FIPS.

Enabling TLS to connect to the database

The following step-by-step instructions explain how to enable TLS encryption to establish a secure connection between the database and Faspex 5.

1. Modify the `db.env` file and set the `FASPEX_DB_USE_TLS` variable to `true`.
2. Place your `ca-cert.pem` file in the `/opt/aspera/faspex/conf/db_ssl_public/ca-cert.pem` directory.
3. Run `faspexctl setup`.

Allowing package submission from external users

Configure Faspex to allow Faspex users to invite external users to send them packages. Faspex users can, if allowed, send an invitation or share a user-specific, public submission link. Unless you require it, external users don't have to log in to send a package using an invitation or URL.

Public submission links

A public submission link allows anyone to send packages to the associated Faspex user or shared inbox without logging in.

Each user has a unique public submission link. When enabled at the server level, users can decide whether to enable their public submission links in their account preferences.

Public submission links are enabled by default. Anyone with a public submission link can send a package to the associated user or shared inbox, even if they are not the intended users of the public submission link. You can increase security by disabling public submission links at a user or shared-inbox level or by disabling public submission links entirely.

Disabling public submission links for a user

1. In the admin application, go to **Users > All users**.
2. Select the user.
3. Override **Allow public submission URLs** and choose **Deny**.
4. Click **Save**.

Allowing shared inbox admins to toggle public submission links

1. In the admin application, go to **Security > Advanced collaboration**.
2. Enable **Shared inboxes admins can enable or disable shared inbox public submission links**.
3. Click **Save**.

Disabling public submission links for a shared inbox

1. In the packages application, go to **Manage shared inboxes**.
2. Select the shared inbox.
3. Go to the **Submission link** tab.
4. Override default settings and choose **Deny**.
5. Click **Save**.

Disabling all public submission links on the server

1. In the admin application, go to **Security > Advanced collaboration**.
2. Disable **Faspex users and shared inboxes can generate public submission links**.
3. Click **Save**.

External user invitations

When enabled at the server level, users can invite a external user to send packages to their account or to a shared inbox by sending them an email invitation.

External user invitations are enabled by default. Anyone with an invitation can send a package to the associated user or shared inbox, even if they are not the intended users of the invitation. You can increase security by setting a global invitation link expiration policy, disabling invitations at a user or shared-inbox level, or by disabling all invitations entirely.

For instructions on sending an invitation, see "Send an email invitation to allow someone to send you packages" on page 49.

Disabling invitations on the server

1. Go to **Security > Advanced collaboration**.
2. Disable **Faspex users can invite external users to send packages to them**.
3. Click **Save**.

Disabling invitations for a specific user

1. Go to **Users > All users**.
2. Select the user.
3. Override **Allow inviting external users** and choose **Deny**.
4. Click **Save**.

Admins can enable or disable the Public submission link feature for specific users despite global settings by going to **Accounts**, selecting the user, going to the Permissions section, and choosing **Allow**, or **Deny** for **Allow public submission urls**.

Setting a global, sending-based invitation-expiration policy

1. Go to **Security > Advanced collaboration**.
2. Enable **Set invitation link expiration policy for both personal and shared inbox invitations**.
3. Select **Link expire after a collaborator successfully submits a package**.

   Note: While the link expires after a collaborator successfully sends a package, it is possible for a collaborator to initiate parallel uploads using a single link to submit multiple packages.
4. Click **Save**.

Setting a global, time-based invitation-expiration policy

1. Go to **Security > Advanced collaboration**.
2. Enable **Set invitation link expiration policy for both personal and shared inbox invitations**.
3. By default, users can override the global time-based link expiration policy. You can disable this ability by toggling off **Allow users to set custom link expiration policy**.

   Note: Shared inbox admins always have the ability to configure a shared inbox's invitation-expiration policy.
4. Select **Time-based policy** and enter the number of days the link stays valid.

   Note: You can enforce the number of days a max value by selecting **Always enforce as maximum duration for expiration policy**.
5. Click **Save**.

Allowing public packages

A public package is a package that recipients can download without logging in. Making a package public only affects Faspex users. Recipients without an account are still required to log in before downloading the package.

Go to **Security > Advanced collaboration** and enable **Senders can allow IBM Aspera users to download their packages without logging in**.

Metadata profiles

Metadata refers to the additional information that an Faspex user can include when sending a package. A metadata profile is a set of additional requirements that an admin can apply to all new packages or to specific shared inboxes.

Metadata example

For example, in the context of a media project, to require your audio engineers include additional metadata about the project files they send to you:

1. Create a metadata profile that requires senders to specify additional field such as sample rate, bit depth, and compression.
   - Sample rate (text input field)
   - Bit Depth (option list that includes 8-bit, 16-bit and 24-bit)
   - Compression (text input field)
   - Date Created (date picker)
2. Create a shared inbox and apply the newly created metadata profile.
3. Invite your audio engineers to send packages to the shared inbox.

You can view the metadata by viewing the package in Faspex. You can also save the metadata as an XML file in the package (`aspera-metadata.xml`) using the **Save metadata to file** option (available globally and at a per-shared-inbox level).

Forwarding packages with metadata

When you forward a package, the original metadata is preserved in the **Note** field. The preserved metadata in the **Note** does not change even if the applied metadata profile has been changed. No new **aspera-metadata.xml** file is created, even if **Save metadata to file** is enabled for the metadata.

Metadata and IBM Aspera Console reporting

If a Faspex instance is added to IBM Aspera Console as a managed node, Console monitors transfer details of transfers in Faspex. Custom metadata fields are included as metadata tags in the transfer details and as transfer cookies for Console to use in running reports.

A Faspex transfer cookie is formatted in the following way:

```
{"aspera":
  {"faspex":
    { "key1":"val1", … , "key3":"val3"}
  }
}
```

The corresponding JSON match value is shown below:

```
[aspera][faspex][key1]val1
```

Create a metadata profile

Require users to provide additional information when sending a package.

1. Go to **Metadata profiles** and click **Create new**.
2. Name the metadata profile and set illegal characters and max lengths for the package title and package note.

   Note: You can disable the ability to add a note to a package by disabling the **Enabled** toggle.
3. Click **Add field** to add metadata fields to the profile:

   Each field option has its own template:

Type	**Description**
Text field	Create a single-line text field.
Text area	Create a multi-line text field.
Option list	Create a drop-down options list.
Date field	Create a date picker.

   When done click **Save**.

   Faspex displays each new field below the **Add field** button. You can edit, delete, and reorder fields.
4. Add additional fields until you are done.
5. Click **Create**.

Selecting a default metadata profile

You can chose one metadata profile to be the default profile. Faspex applies that profile automatically to all packages sent to users, distributions lists, and workgroups. The default profile does not apply to packages sent to a shared inbox. Besides any custom metadata you include in the profile, can also configure the default profile to generate an XML file containing package metadata.

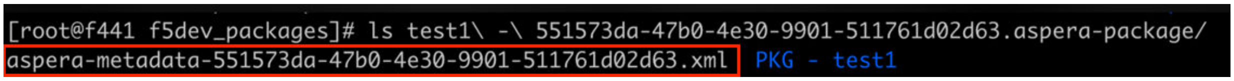
Like all metadata profiles, a default profile adds fields to the Send files form.

To select a default metadata file, you must create at least one metadata profile (see "Create a metadata profile" on page 106).

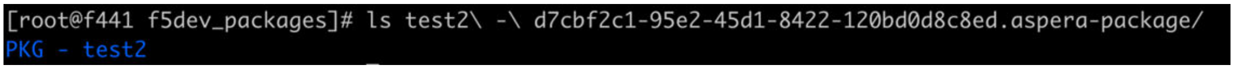
1. Go to **Metadata profiles > Default settings**.
2. Select a profile from default profiles drop-down menu.
3. To generate an XML sidecar file with each normal sent package, set the toggle labeled **Include metadata in a sidecar file** to **On**.

   Faspex uses the following naming format for the metadata file: **aspera-metadata-*package_uuid*.xml**. For example: **aspera-metadata-42dfda4c-ff05-4f61-8d82-f89c0523d799.xml**.

   Unless you complete the next step, Faspex saves the XML sidecar file in the root package directory. For example:

   ```
   [root@f441 f5dev_packages]# ls test1\ -\ 551573da-47b0-4e30-9901-511761d02d63.aspera-package/
   aspera-metadata-551573da-47b0-4e30-9901-511761d02d63.xml  PKG - test1
   ```
4. To save the XML file inside the package directory rather than in the root package directory, set the toggle labeled **Store sidecar file in package directory** to **On**.

   ```
   [root@f441 f5dev_packages]# ls test2\ -\ d7cbf2c1-95e2-45d1-8422-120bd0d8c8ed.aspera-package/
   PKG - test2
   ```
5. Click **Save**.

Managing users

Create, configure, and remove Faspex users. For an overview of the different users, see "User roles" on page 3.

User roles

Admins assign user roles to an account when creating a new account or when configuring an account's permissions.

Important: You cannot change your own assigned role.

User accounts can have these user roles:

Role	Description
Regular user	Regular users can send and receive packages, as permitted by admin-configured server settings.
Manager	In addition to regular user permissions, managers can manage: • regular users • workgroups • external users • SAML groups Managers can access all shared inboxes and manage shared inbox members and workgroups. Managers cannot create new managers, edit admin accounts, or promote another user to an admin or manager role.
Admin	In addition to manager permissions, admins can adjust server configurations, access all packages and relays, manage all workgroups and shared inboxes, and manage all users.
External user	A external user is a user that is not associated with a Faspex user account. Faspex users can send *public packages* to external users with a *public submission link* and invite external users to send them a package through an email invitation or with a *public submission link*. By default, external users do not need to create an account to download a public package, but an admin can configure Faspex to require external users to create an account before downloading the package (see "Allowing public packages" on page 105).

Creating a user

Create a user account in Faspex.

1. Go to **Users > All users**.
2. Click **Create new**.
3. Enter the user's email address. The email must be unique to Faspex.

4. Enter the user's first and last name.
5. Select the user's role:

Role	Description
Regular user	Regular users can send and receive packages, as permitted by admin-configured server settings.
Manager	In addition to regular user permissions, managers can manage: • regular users • workgroups • external users • SAML groups Managers can access all shared inboxes and manage shared inbox members and workgroups. Managers cannot create new managers, edit admin accounts, or promote another user to an admin or manager role.
Admin	In addition to manager permissions, admins can adjust server configurations, access all packages and relays, manage all workgroups and shared inboxes, and manage all users.
External user	A external user is a user that is not associated with a Faspex user account. Faspex users can send *public packages* to external users with a *public submission link* and invite external users to send them a package through an email invitation or with a *public submission link*. By default, external users do not need to create an account to download a public package, but an admin can configure Faspex to require external users to create an account before downloading the package (see "Allowing public packages" on page 105).

6. You can further configure the user by expanding **Account settings** and **Permissions**. See "Reference: User account settings" on page 166 and "Reference: User permissions" on page 168, respectively.
7. Click **Create**.

Faspex sends a welcome email to the email address. Using the link provided in the email, the user creates a password and logs in.

Adding global user properties

Add additional properties to user accounts. Adding additional properties adds fields to new user and self-registration forms.

SAML configurations also inherit global user properties. If you add fields and set them as required, make sure to adjust settings in each SAML configuration so that the required fields do not prevent SAML users from logging in.

You can add additional properties by going to **Users > User properties**. You can add up to five properties. Each field can be required or optional.

Reactivating an inactive account

A user account becomes inactive if an admin deactivates the user or if the user deactivates due to inactivity.

To reactivate a user:

1. Go to **Users > All users**.
2. Select a user.
3. Enable **Account is active** in the Account settings section.

Deleting a user

Deleting a user does not delete a user's received packages. This means that:

- Admins can still see and download packages sent to a deleted user.
- Users that received a package from a deleted user can still access the package, but the package sender is changed to your account.

Note: Deleting a user does not remove that user from distribution lists. If a user sends a package to a distribution list containing a deleted user, Faspex sends the package to the user's email address as a new external user. If the sender is not allowed to send to external users, sending the package fails.

1. Go to **Users > All users**.
2. Right-click the user you want to delete and select **Delete**.

   You can also delete multiple users at once.
3. Confirm deletion.

Review accounts with duplicate email addresses

If there is more than one user account with the same email address, Faspex does not allow anyone to log in with that email address.

User accounts may share email addresses if you upgraded from Faspex 4.X and before, since Faspex did not require unique email addresses in those versions. These users can still log in to their accounts using their usernames, but they cannot log in using their email addresses. To allow users to log in with their email addresses, you must manually resolve accounts with duplicated email addresses by changing emails or by deleting accounts until the email address is unique.

1. Log in to Faspex Utility.
2. Go to **Manage users**.
3. Take note of the accounts with duplicated email addresses.
4. Log in to Faspex as a user with admin permissions.
5. Click the application switcher icon (⋮⋮⋮) and select **Admin** from the drop-down menu.
6. Go to **Users > All users**.
7. Choose the user to keep and edit or delete the rest.

Managing external users

An external user is a user that is not associated with a Faspex user account. Faspex users can send public packages to external users with a public submission link and invite external users to send them a package through an email invitation or with a public submission link.

Allow Faspex users to send packages to external users

Allow Faspex users to send packages to external email addresses not associated with a Faspex account.

1. Go to **Security > Advanced collaboration**

2. Enable **IBM Aspera users can send packages to external users** and set the default to **Allow**.

   Choosing **Allow** enables all users to send to external email addresses by default. An admin override this setting for specific users.
3. Click **Save**.

When a user sends to an external email address that is not associated with an existing Faspex user, Faspex creates a new external user with that email address.

Removing external users from Faspex

Whenever a user sends to an email address not associated with a user account, Faspex creates an external user. Remove an external user when the external user should no longer have access to received packages.

1. Go to **Users > All users** and click **External users**.
2. Right-click the user you want to delete and select **Remove**.

   You can also delete multiple users at once.
3. Confirm removal.

Managing self-registration

You can allow users without an account to request an account from the login page. Ensure that proper security settings have been configured before allowing self-registration.

Enabling self-registration

Allow anyone to request accounts from the Faspex login page. You must ensure that proper security settings have been put into place before allowing self-registration.

Self-registration relieves the workload of admins and managers. The self-registration feature is turned off by default.

New user accounts inherit the permissions of the configured template user and automatically becomes a member of designated workgroups. To configure the template user, go to **Accounts > Pending Registrations** and click the **template user** link.

⚠ **Warning:** If self-registration is enabled, a user can use it to find out whether a certain account exists on the server. If a user attempts to self-register a duplicate account, then the user receives a prompt stating that the user already exists.

1. Go to **Security > Account registration** and enable self-registration.
2. Choose whether to require approval for account registration.

   If you allow self-registration, use the **Approval required** setting for security purposes. An admin or manager must approve or deny the account before the user can log in. For more information on approving or denying accounts, see "Approving or denying pending registrations" on page 60.

   If you choose to require approval, you can choose which users are notified through email of a new request.

   Note: These email addresses are not validated against existing Faspex admins or managers, but only admins and managers can approve account requests.
3. If you choose to require approval, you can also configure these settings:

Configuration	**Description**
Terms of service	If text is set, Faspex requires users to accept the terms of service in order to register an account.

Configuration	Description
Self-registered users can send packages to other self-registered users	Allow self-registered users to send packages to other self-registered users.
Blocked list	New users are not allowed to register accounts using emails from these email domains.
Require external users to register	Require external users to register a Faspex account to download packages.

4. Add any registration instructions you want to provide for someone creating or requesting an account.
5. Click **Save**.

Approving or denying pending registrations

1. Go to **Users > All users** and click the **Pending registrations** tab.
2. Click the **Approve** or **Deny** icon.

Approved users receive the Faspex welcome email and can use the password reset link in the email to set their passwords and log in.

Approved users automatically inherit the permissions of the template user and will become members of a workgroup, if configured to do so. After creation, you can update the permissions and workgroup memberships of these users from the **Users** tab.

Configure self-registration user defaults

New users created by approved self-registration requests inherit self-registration user defaults. You can configure these defaults by going to **Security > Account registration** and clicking **Configure defaults**.

Account details and security

Setting	Description
Account expires	Select to set an expiration date for the user. The user becomes inactive on the specified date. **Note:** Admin accounts do not expire.
Account is active	Select to activate this account so that the user can log into Faspex. Clear to disable the account. **Note:** Admin accounts are always active.

Custom password policy

Setting	Description
Override default password policies	Enable to set password expiration and password reuse limits.
Password expires and **Days until password expires**	Expire passwords a specified number of days after they are set.
Prevent password reuse and **Number of previous passwords**	Prevent users from reusing a specified number of previously used passwords.

User permissions

Setting	Description
Upload packages	The user can upload packages to a node. Disabling this does not prevent the user from sending a package from a shared folder, if **Create packages from a remote source** is enabled.
Download packages	The user can download received packages. When disabled, the user can still receive packages, but cannot download packages or the packages' files.
Forward packages	The user can forward received packages to other users.
Send packages from a remote source	The user can include content from a shared inbox when sending a package. You must first add remote sources to Faspex to see the **Source** drop-down menu. This setting must be set on a per-user basis. There is no global option to enable this setting for all users.
Send normal packages	If disabled, the user can only send packages to shared inboxes. You can change the server default by doing to **Security > Users**.
Invite external senders	You must enable this option globally to override this feature. For more information, see “Allow Faspex users to send packages to external users” on page 109. Override the server default and select **Allow** to enable this user to invite anyone to send a package to this user.
Share submission links	You must enable this option globally to see this feature. For more information, see “Public submission links” on page 103. Override the server default and select **Allow** to enable users to share their public submission link. Anyone with a public submission link can submit packages to this user through this submission link. **Note:** Even if the public submission link feature is enabled for registered Faspex users, they can override the feature for their own account by going to their user preferences. See “Share your public submission link” on page 49.
Send packages to external emails	Override the server default and select **Allow** to enable the user to send packages to external email addresses. For more information, see “Allow Faspex users to send packages to external users” on page 109 Select **Allow** to enable this user to send packages to external.
Send to all Faspex users	Override the server default and select **Allow** to allow users to send packages to all Faspex users. If this feature is enabled, all existing Faspex users appear in the contact list. By default, users can send packages to: • Users in their contacts • Personal distribution lists • Shared inboxes • Workgroups they are a part of (if workgroup settings allow) • Members of workgroups they are a part of (if workgroup settings allow) • Global distribution lists (if allowed to see)

Setting	Description
See global distribution lists	Override the server default and select **Allow** to enable the user to see and send to global distribution lists.
Keep user directory private	Override the server default and select **Yes** to prevent users from being able to see the entire user directory, even if they have permissions to send to all Faspex users.

IP permissions

Important: Restricting access by IP address requires the requesting client's IP address be preserved through the TCP/IP network until the request reaches Faspex. If you have a network configuration that passes the requesting client's IP address through the network with the `X-Forwarded-For` HTTP Header, you will not be able to restrict access to Faspex via IP addresses.

Tip: For all of these settings, you can use a wildcard (*) to allow a range of options. For example, specifying `192.0.2.*` allows a user to login from `192.0.2.1`, `192.0.2.2`, `192.0.2.3`, and so on. Separate multiple IP addresses with commas (,).

Setting	Description
User can log in from only these IP addresses	Specify the IP addresses that a Faspex user can login from.
User can download only when downloading from these IP addresses	Specify the IP addresses that the user must access Faspex from to download packages.
User can send packages only when sending from these IP addresses	Specify the IP addresses that the user must access Faspex to upload packages.

Working with SAML

Configure Faspex as a service provider (SP) to connect with your SAML identity provider (IdP) to authenticate users. Authenticated users can then use Faspex to access secure content.

With SAML enabled, Faspex redirects a user to the IdP sign-on URL. The user signs in with the IdP and the IdP sends a SAML assertion back to Faspex. When a SAML user logs in to Faspex for the first time, Faspex automatically creates a new user account based on the information provided by the SAML response. Any changes subsequently made to the account on the DS server are not automatically picked up by Faspex. For more information about user provisioning for SAML users, see "Reference: User accounts provisioned by Just-In-Time (JIT) provisioning" on page 170.

If you want to use directory services with Faspex, configure your SAML IdP to act as a front-end for the directory service.

Multiple SAML configurations

Faspex supports multiple SAML configurations on the same server. Faspex redirects users to the default SAML IdP, but if no default is specified, Faspex directs users to the local login page where users can choose to log into publicly visible SAML configurations or log in locally.

To configure multiple SAML configurations in Faspex, first create a new SAML configuration (see "Creating a new SAML configuration in Faspex" on page 115) and then configure an alternate address for the configuration (see "Configuring a SAML alternate address" on page 117).

Multi-factor authentication (MFA)

As a Faspex 5 SAML user you can enforce the use of multi-factor authentication (MFA) through a supported SAML identity provider (IdP). Multi-factor authentication is not supported for Faspex 5 local users.

Identity provider (IdP) requirements

IdP requirements

Make sure your IdP meets these requirements:

- Supports SAML 2.0
- Able to use an HTTP POST Binding
- Not configured to use pseudonyms
- Can return assertions to Faspex that include the entire contents of the signing certificate
- If prompted, set to sign the SAML response

Uploading a Faspex SAML configuration metadata file to configure the IdP

If the IdP is capable of reading SAML XML metadata for a service provider, you can upload a saved XML metadata file to configure the IdP.

To access the XML metadata:

1. Go to **Server > Authentication > SAML**.
2. Right-click the SAML configuration and select **Metadata** from the overflow menu.

IdP metadata tags

Configure IdP tags for Faspex:

Tag	Format
NameID Format	Supported formats: • `urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified` • `urn:oasis:names:tc:SAML:1.1:nameid-format:transient` • `urn:oasis:names:tc:SAML:1.1:nameid-format:persistent` • `urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress`
Entity ID	`https://`*`faspex_ip`*`/aspera/faspex/api/v5/samls/`*`saml_id`*`/saml_metadata`
Binding	`urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST`
Callback URL	`https://`*`faspex_ip`*`/aspera/faspex/api/v5/samls/`*`saml_id`*`/callback`

SAML assertion requirements

Faspex expects assertion from an IdP to contain the following elements:

Default Attribute	Faspex User Field	Required
NameID	Username	Yes
email	Email address	Yes
given_name	First name	Yes

Default Attribute	Faspex User Field	Required
surname	Last name	Optional
member_of	SAML group	Necessary for SAML groups

Note: Some IdPs may refer to the **NameID** attribute as **SAML_SUBJECT**.

Creating a new SAML configuration in Faspex

Before configuring SAML in Faspex, make sure you have properly configured your SAML IdP (see "Identity provider (IdP) requirements" on page 114).

1. Go to **Authentication > SAML** and click **Create new**.
2. Enter the configuration information.

   You can import the SSO URL, fingerprint, and certificate from metadata (URL or saved file) or by manually entering the information.

Field	Description	Example
Name	Name to identify the SAML configuration in Faspex, used to differentiate between multiple SAML configurations.	`Shibboleth SAML`
SSO target URL	The IdP Single Sign-On URL.	`https://shib-idp-01.example.com/idp/profile/SAML2/Redirect/SSO`
Fingerprint	The SAML fingerprint used to authenticate to the SAML IdP.	`4D:88:C3:51:94:94:16:EE:51:3D:28:89:6C:C1:E5:43:A8:24:EA:C2`
Fingerprint algorithm	The algorithm type for the SAML fingerprint.	SHA1

Certificate	The SAML certificate used to authenticate to the SAML IdP. If you are providing a fingerprint, you do not need to provide a certificate (the fingerprint is a digest of the certificate).	`-----BEGIN CERTIFICATE-----`
		`MIIDUDCCAjigAwIBAgIVANkYVO1`
		`LBB6MuViBihCDECq8XoAxMA0GCS`
		`qGSIb3DQEBBQUAMCQxIjAg`
		`BgNVBAMTGXNoaWItaWRwLTAxLmR`
		`ldi5hc3BlcmEudXMwHhcNMTMxMT`
		`A2MjIzODAzWhcNMzMxMTA2`
		`MjIzODAzWjAkMSIwIAYDVQQDExl`
		`zaGliLWlkcC0wMS5kZXYuYXNwZX`
		`JhLnVzMIIBIjANBgkqhkiG`
		`9w0BAQEFAAOCAQ1AMIIBCgKCAQE`
		`Akk7e5VrTJpcmeQTbNQXlgBTgpe`
		`Wkkhx+8t4zpEh4UbQr8sXh`
		`so9GtDQjVhasWMfGPAO+Mlp112e`
		`XVvT8uQQMBh2Ce7qSx1aXl4ZsJw`
		`+mPfuRf6xIZDk5sVNfY801`
		`SxXbeVvPSGXN6lTPV7/0/`
		`dd4s+IMIeG6NfIdfpFbYa4F2QaJ`
		`D28ergf3KELzHkrBWti55NH8Np4`
		`9`
		`rk5Iq0fk56YR1KuETHI2pS3vvVI`
		`OJMwIhOvOrsNxHu0O6oohFmLM5k`
		`+yHQqur1Lk0mV9GFZnwDWC`
		`lwPcLKvJ6gTv8k4hUkI0fhWUVOE`
		`NcleyyEc9acnMXCrnM424eW4QnK`
		`E1H8u8xO6DcwIDAQABo3kw`
		`dzBWBgNVHRDDTzBNghlzaGliLWl`
		`kcC0wMS5kZXYuYXNwZXJhLnVzhj`
		`BodHRwczovL3NoaWItaWRw`
		`LTAxLmRldi5hc3BlcmEudXMvaWR`
		`wL3NoaWJib2xldGgwHQYDVR0OBB`

Field	Description	Example
		YEFPZq25rft0WK+9WvL+Wl +W+knKH2MA0GCSqGSIb3DQEBBQU AA4IBAQAhCICuALkaLW1glDVtp8 YuYB3FZqBn0Y3ekt/OUXIU uGwXDYhR8FdumXhGIGdUaPlQHd3 MnZRIVougy7fS/Qyg8V/ C8ALa5g7K/2sTOi/ RtMjRQZK+vOlO oxneqotk4BPGp3an+m1pdnxjJvp hL4kX/ZPuCcvkyzoDnelv/c+dE/ +Yz6IzmL1j/drsxRL8etPc jpgGjIF4TDGTNDDhleOyLP3yN2a NPqEpF/ Y8WOVhejrkux2YKwH6SQVKdSgod D6EVsUs13F1atvB BRRwBWgG2lFBnVRl01r3LOjH0Vt FK/ Hms3V3L9jE7ucR+qDbWNdPEmWvB Y2aHr0EQU/NscQl -----END CERTIFICATE-----
Name ID format	The Name ID format must match the format used with your IdP. Faspex supports these formats: • Unspecified • Transient • Persistent • Email Address When set to "Unspecified", any Name ID format returned by the IdP is accepted.	

3. Configure the default profile fields. These fields must map to attributes in your SAML IdP's SAML response. Enter the **SAML Name** for each of the required fields: **username**, **email**, **first_name**, and **last_name**.

   Important: Once you set the value for **username**, do not change it. If **username** is changed, existing SAML users can no longer log into their existing Faspex accounts, but are instead given new accounts with new usernames.

4. Optional: Configure local custom profile fields.

   These are custom user attributes that only apply to this IdP. **Name** is the name of the attribute displayed in Faspex. **SAML Name** is the name of the attribute as configured in the IdP. To add a field, click **Add Local Profile Field**. For more information, see "Setting up custom SAML profile fields" on page 119.

   Note: If you've configured custom attributes (**Server > User Profile**), these fields show up as Global Custom Profile Fields that, if required, you must map to valid SAML names. For more information about custom attributes, see "Setting up custom SAML profile fields" on page 119.

5. Click **Create**.

   After creating a new SAML configuration, Faspex redirects you to the SAML Configurations page and displays the existing SAML configurations.

Users can now access Faspex through SAML instead of going through the local login page. For information about bypassing the SAML redirect, see "Bypassing SAML redirection" on page 38.

Configuring a SAML alternate address

SAML alternate addresses allow users to directly access a SAML IdP. A user may use a SAML alternate address to bypass the default SAML IdP if the user is not a member of that IdP (for example, `shibboleth.faspex.example.com`).

Configuring a SAML alternate address requires you to:

1. Allow users to access Faspex from an alternate address.
2. Update your IdP with an XML metadata file generated for the alternate address.
3. Enable the SAML alternate address in Faspex.

Allow access from Faspex from an alternate address:

1. Verify with your IT department that the alternate address resolves to your Faspex server's hostname in your DNS.
2. Allow the alternate address in Faspex (see “Configuring alternate addresses” on page 94).

Update your IdP with an XML metadata file generated for the SAML alternate address. Faspex generates the XML metadata file based on the URL used to access the server. To generate the XML metadata for your SAML alternate address, you need to first log in to the SAML alternate address.

3. Go to the SAML alternate address in your browser and log in as your admin user.
4. Update your IdP with the metadata for the SAML alternate address:

   a) Go to **Server > Authentication > SAML**.

   b) Right-click the SAML configuration and select **Metadata** from the overflow menu.

   c) Copy the metadata and update your IdP following your IdP's instructions.

Enable the SAML alternate address in Faspex.

5. In Faspex, edit the SAML configuration, click the **Authentication rules** tab, and enter the alternate hostname in the **SAML alternate address** text field.

   You do not need to enter a full URL. For example, you can use `idp.faspex.com` instead of `https://idp.faspex.com`.
6. Test that the SAML alternate address redirects you to your SAML IdP and that the SAML IdP allows you to log in.

Creating SAML groups

SAML groups manage permissions for all SAML users that are members of the group.

Note: SAML groups in Faspex are called "Faspex SAML groups" and SAML groups in the SAML Identity Provider (IdP) are called "IdP groups".

Faspex SAML groups are associated with IdP groups by the IdP group distinguished name (DN). Faspex SAML groups are created in one of two ways:

- Automatically: When a SAML user with IdP group membership logs in, Faspex automatically creates new Faspex SAML groups for each of that user's IdP groups that do not yet exist in Faspex. Faspex adds the SAML user to the newly created Faspex SAML groups.

  Note: If an admin enables the **Restrict access to known groups** feature for the SAML configuration, only members of existing Faspex SAML groups can log in. This also means that new SAML groups are not automatically created when SAML users log in. For more information about SAML configuration options, see “Configuring authentication rules for a SAML configuration” on page 118.
- Manually: Faspex must have at least one enabled SAML configuration. An admin creates a Faspex SAML group and associates it with an IdP group by the IdP group's DN. When a SAML user that is a part of that IdP group logs into Faspex, Faspex adds the SAML user to the Faspex SAML group.

On the SAML groups page, you can activate, deactivate, or remove existing groups from the **Actions** drop-down menu. If a user belongs to only one group and that group is deactivated, the user cannot login anymore. If a user belongs to multiple groups and at least one of these groups is active, the user can log in.

To manually create a SAML group in Faspex:

1. Go to **Authentication groups > SAML groups** and click **Create a SAML group**.
2. Enter the group name. This is the IdP group's distinguished name (Identifier of the group).

   Note: For Azure Active Directory use the Object ID of the group.
3. Select the **SAML Configuration** the IdP group belongs to.
4. Expand **Permissions** and **Advanced settings** to configure settings such as account permissions and package deletion parameters.
5. Click **Create**.

Configuring authentication rules for a SAML configuration

To configure an existing SAML IdP, go to **Authentication > SAML** and click the name of the IdP.

Option	Description
Publicly visible	Determine whether Faspex allows users to choose this IdP as an option from the local login page. If selected, Faspex displays this IdP as a login option. If not selected, Faspex does not display this IdP and users must access the IdP using a domain URL.
Restrict access to known groups	Prevent SAML users that are not members of existing Faspex SAML groups from logging into this IdP. If a user is a member of multiple groups, the user can log in as long as one of those groups exists in Faspex. If this feature is enabled, Faspex does not create new groups for users that are a member of multiple SAML groups. For more information about automatically creating new groups, see “Reference: User accounts provisioned by Just-In-Time (JIT) provisioning” on page 170. For more information about SAML groups, see “Creating SAML groups” on page 117.
Restrict access to known users	Prevent users that are not existing Faspex SAML users from logging into this IdP.
Redirect to SAML upon logout	When SAML users log out of Faspex, they are redirected to the SAML logout page instead of the local login page. From the SAML logout page, users can log back into Faspex with SAML.
Default SAML configuration	Determine if accessing the Faspex URL redirects users to this IdP or to the local Faspex login page. If selected, accessing the Faspex URL directs them to this IdP. If not selected, users arrive at the local login page instead. If the admin does not specify a default SAML configuration, Faspex automatically redirects users to the local login page. For more information on bypassing the SAML redirect, see “Logging in with SAML” on page 38. **Note:** The default SAML configuration does not affect client applications (IBM Aspera Connect) connecting to Faspex. When adding Faspex to Connect as a connection, Connect users are not redirected to the default configuration. Connect users can instead choose from all enabled SAML configurations.
Allowable clock drift (ms)	Configure the milliseconds allowed for clock drift between Faspex and the SAML IdP.

Setting up custom SAML profile fields

You can add additional profile fields to import additional information from a SAML Identity Provider (IdP)'s SAML response. These fields must be correctly mapped to the SAML IdP. You can import different profile fields for each individual IdP.

Faspex maps user fields to fields in the SAML response. There are three types of profile fields:

- Default fields: All SAML responses must provide these fields. You can edit the SAML name, but not the Faspex field name.
- Global fields: SAML inherits any global profile fields set for the server. You can edit the SAML name, but not the Faspex field name. If a global fields is required but you do not want to map it to a SAML field, make it optional.
- Local fields: You can create additional fields a particular SAML configuration.

To add a local SAML field:

1. In your SAML IdP, add new SAML fields in your SAML IdP.
2. Go to **Authentication > SAML** and select the associated SAML configuration.
3. Go to the **Profile values** tab.
4. Scroll down to Local custom profile fields and click **Add field**.

   The **SAML Name** must be correctly mapped to your SAML fields in IdP. If the names are incorrectly mapped, Faspex does not allow SAML users to log in. If the names are correctly mapped but the SAML response does not include required attributes, Faspex does not allow the SAML user to log in.

Disabling SAML configurations

Faspex does not allow you to delete SAML configurations. To stop allowing users to log in using a SAML configuration, disable it instead.

1. Go to **Server > Authentication > SAML**.
2. Right-click the SAML configuration and select **Disable** from the overflow menu.

Managing workgroups

Workgroups define a group of users that can be sent packages as a collective whole. Use workgroups set the same permissions for a group of users.

A Faspex administrator determines:

- Who has permissions to send packages to a workgroup, including whether workgroup members can see and send packages to other workgroup members.
- Where packages sent to the workgroup are stored.

Creating a workgroup

Create a workgroup and add members to the workgroup.

This procedure covers the minimum requirements to create a workgroup:

1. Go to the **Admin** app from the app switcher.
2. Go to **Workgroups**.
3. Click **Create new**.
4. Name the workgroup.
5. Click **Save**.
6. Select the newly created workgroup.
7. Go to the **Members** tab.
8. To add Faspex users:

a) Click **Add member**.

b) Search for and select the Faspex users you want to add.

c) Select the workgroup permissions for the users.

d) Click **Add**.

For more information on the **Permissions** options section, see "Configuring workgroup permissions and privacy settings" on page 58.

Configuring workgroup permissions and privacy settings

Faspex admins can designate workgroup members as workgroup admins. Workgroup admins can have additional permissions granted per workgroup. All Faspex admins are workgroup admins of every workgroup.

To manage workgroup settings, go to the **Settings** tab of any workgroup.

Workgroup admin permissions

There, you can enable workgroup admins to:

- **Add existing IBM Aspera users and remove non-admin members**: Workgroup admins can add any user from the user directory to the workgroup. This must be enabled for workgroup admins to add members.
- **Create and add new IBM Aspera users**: Workgroup admins can create a new user and add that user to the workgroup.
- **Add and remove SAML groups**: Workgroup admins can add any SAML group to the workgroup.

Note:

- All Faspex admins and managers have these permissions for every workgroup.
- IBM Aspera users do not include external users.

Workgroup privacy settings

You should restrict visibility to increase member privacy.

Who can see and send packages to this workgroup

- No one
- Workgroup members only
- Workgroup admins only
- Anyone

Who can see and send packages to other workgroup members

- No one
- Workgroup admins only
- All workgroup members

Important: If you set the permission for both options No one, no one can see or send to the workgroup and workgroup members cannot see or send to each other, so members in the workgroup are members in name only.

Managing workgroup members and SAML groups

Add Faspex users and SAML groups to the workgroup. When a user sends a package to a workgroup, all workgroup members can access the package in their inbox.

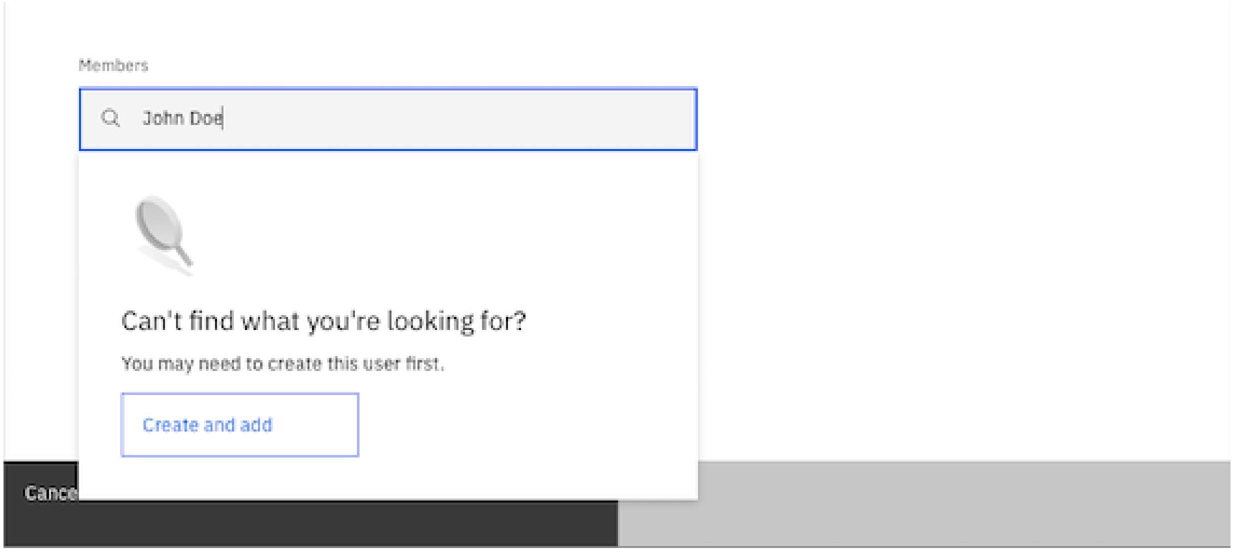
Permissions

Workgroup admins can

Add members

1. After selecting a workgroup, go to the **Members** tab to manage workgroup members.
2. Click **Add member**.
3. Search for and select the Faspex users you want to add.

   If you are a workgroup admin and the workgroup permissions allow you to create and add new users, you can click **Create and add** to create a new user and add the user to the workgroup.



Members

John Doe

Can't find what you're looking for?

You may need to create this user first.

4. Choose the access permission to grant to these users.
   - **Standard**: These members can see packages sent to the workgroup and can send packages to the workgroup (if permitted by workgroup settings).
   - **Submit only**: These members can send packages to the workgroup (if permitted by workgroup settings).
   - **Workgroup admin**
5. Click **Add**.

Add SAML groups

1. After selecting a workgroup, go to the **Members** tab to manage workgroup members.
2. Select the **SAML groups** toggle.
3. Click **Add group**.
4. Select the group to add.
5. Choose the access permission to grant to these users.
   - **Standard**: These members can see packages sent to the workgroup and can send packages to the workgroup (if permitted by workgroup settings).
   - **Submit only**: These members can send packages to the workgroup (if permitted by workgroup settings).

- **Workgroup admin**

6. Click **Add**.

Setting a custom destination for a workgroup

Choose your workgroup destination inbox. Packages sent to the workgroup are stored at this location.

If you have a group of users that are not permitted to access the server-default storage location, use a workgroup with a custom inbox configured. Packages sent to the workgroup are first uploaded to the server-default storage location, and then relayed to the specified storage location. When the workgroup users download the package, they download the package from the custom inbox location.

When packages are deleted from the default location, they are not automatically removed from the custom location.

Note: When symbolic links are enabled for a storage location, packages in the custom inbox location are stored as actual files, not symbolic links. The default inbox location contains symbolic links, but the custom inbox contains actual files.

1. Select a workgroup and go to the **Nodes and storage** tab.
2. Under inbox destination, select **Custom**.
3. To upload directly to the specified storage location instead of first uploading to the server-default storage location, select **Upload directly to custom inbox**.
4. Select a node from the table and select the storage location you want to use as the custom inbox.
5. Click **Save**.

Forwarding workgroup packages to another storage location

Forward packages sent to a workgroup to another storage location using relays.

Packages sent to a workgroup are first uploaded to the server-default storage location and then relayed to the specified storage locations. When packages are deleted from the default location, they are not automatically removed from the custom location.

1. Select a workgroup and go to the **Nodes and storage** tab.
2. Select **Relay**.
3. Select **Enable relays**.
4. Click **Save**.
5. For each storage location you want to relay packages to:
   a) Select the storage location.
   b) Enable **Forward to this relay**.
   c) If you want to overwrite files if they exist on the destination, enable **Overwrite files**.
   d) If you want to notify users when a relay starts, completes, or errors out, enter contacts in the notification fields.
   e) Click **Save**.

Working with relays

Relays transfer copies of uploaded files to specified destinations. There are two types of relays: package relays and file relays.

Overview

There are three types of transfers to Faspex file destinations: direct uploads, package relays, and file relays.

<table>
<tr><th>Transfer type</th><th>Description</th><th>Starts when</th><th>Transferred files directory structure on the destination node</th></tr>
<tr><td>Direct upload</td><td>A direct upload transfers files to the default inbox or to a custom inbox with direct upload configured.
Faspex makes these files available to the designated recipients. Faspex also uses these files when performing a relay.</td><td>A user sends a package to a recipient, workgroup, or shared inbox.</td><td><i>package_title</i> - <i>package_id</i>.aspera-package/
|-- PKG - <i>package_title</i>
    |-- <i>folder1</i>
        |-- <i>file1</i>
    |-- <i>file2</i></td></tr>
<tr><td>Package relay</td><td>A package relay transfers copies of files uploaded to a source node to specified nodes. Package relays preserve the package directory structure.</td><td>A user sends a package to a workgroup or shared inbox with a custom inbox that does not have direct upload configured.
A user sends a package with relay metadata (see “Using metadata fields to set relay destinations” on page 125).</td><td><i>package_title</i> - <i>package_id</i>.aspera-package/
|-- PKG - <i>package_title</i>
    |-- folder1
        |-- file1
    |-- file2</td></tr>
<tr><td>File relay</td><td>A file relay transfers copies of files uploaded to a source node to specified nodes. File relays do not preserve the package directory structure.</td><td>A user sends a package to a workgroup or shared inbox with relays configured.</td><td>folder1
|-- file1
file2</td></tr>
</table>

Hide relay information

By default, the package details panel displays relay information. For additional security, you can hide relay information.

1. Go to **Admin > Relays**.
2. Click **Display settings** to open the Relay display settings window.
3. To hide relay information, click the toggle to **On**.
4. Click **Save**.

Order of Operations

If a transfer triggers both package and file relays, Faspex performs transfers in this order:

1. Direct upload
2. Package relay
3. File relay

If the direct package upload fails, Faspex does not relay the package to custom inboxes or file relay destinations.

If the direct package upload succeeds, Faspex performs both relays. If the package relays fail, Faspex still performs the file relays.

Package relays

A package relay transfers copies of files uploaded to a source node to specified nodes. Package relays preserve the package directory structure.

When a user uploads files to a custom inbox with direct upload or to the default inbox, Faspex creates the following package directory structure on the destination node:

```
package_title - package_id.aspera-package/
|-- PKG - package_title
    |-- folder1
        |-- file1
    |-- file2
```

Once the initial transfer to the initial destination node has completed, then Faspex starts the package relay to copy that entire directory structure to the new destination. The new destination can be:

- A workgroup or shared inbox custom inbox without direct upload configured.
- Destinations specified by package metadata.

File relays

A file relay transfers copies of files uploaded to a source node to specified nodes. File relays do not preserve the package directory structure. File relays transfers files to storage in a flat structure. This is ideal for ingesting files without having to parse nested data from package structure (such as providing files for API consumption).

File relay options

When you configure file relays on a workgroup or shared inbox, you can set the file relay to overwrite existing files with the same name at the destination. By default, Faspex skips files with the same name that exist at the destination node.

You can also configure email notifications for file relays. In the **Server > Notifications** section, you can use the **Relay Started CC** email template to notify users when package forwarding is started, a **Relay Finished CC** email template to let users know when package forwarding is completed, and a **Relay Error CC** email template to notify users when package forwarding has failed. For details see "Reference: Email notification templates" on page 152.

Comparing file relays to package relays

When a user uploads files to a custom inbox with direct upload or to the default inbox, Faspex creates the following package directory structure on the destination node:

```
package_title - package_id.aspera-package/
|-- PKG - package_title
    |-- folder1
        |-- file1
    |-- file2
```

File relays do not preserve this directory structure, but transfers the contents of the package in a flat structure to the destination node:

```
folder1
|-- file1
file2
```

If the destination node already had the `existing_file1` and `existing_folder1` files, the resulting directory snapshot would be:

```
existing_file1
existing_folder1
```

```
|-- existing_file2
folder1
|-- file1
file2
```

Tracking relay progress and status

You can track the progress of a relay by going to **Relays**. You can also track the relay progress for a specific package on the package details page.

Faspex reports these relay statuses:

- `Completed`
- `Failed`
- `Incomplete`
- `In progress`
- `Timed out`

If a transfer triggers both package and file relays, Faspex first reports the package relay status. If the package relay succeeds, Faspex then reports the file relay status. However, if the package relay fails, Faspex reports the error and does not report the file relay status.

Note: Canceling a relay with an `Incomplete` status may not work if the destination node has not received the job. Once the relay status changes to `In progress`, canceling the relay works.

Using metadata fields to set relay destinations

Use the `SenderShareId`, `RecipientShareIds`, `OverrideShareIds` metadata fields to configure relays for a package upload.

Metadata field names for relay destinations use the term *share*. A *share* in this context is the storage location used as a relay destination.

You use the `share_id` of a storage location to designate it as a relay destination. To determine the `share_id` of a file relay destination:

1. Go to **Nodes and storage**.
2. Right-click the node and select **Add file storage**.
3. Select the storage location.
4. Find the `share_id` in the page URL. For example, if the page URL is `https://faspex.aspera.us/aspera/faspex/admin/nodes-storage/4/storage/1/profile`, the `share_id` is 1.

Metadata field	Description	Format	Example
`SenderShareId`	Defines the storage location destination (defined by *share_id*) for the initial upload of a new package. If set, override the default inbox storage setting in Faspex with the storage location destination. If not set, use the default inbox destination as the storage location destination. Use `SenderShareId` to control where the sender uploads to and downloads the sent package from. When the sender downloads the package from the **Sent** mailbox,	*`share_id`*	3

Metadata field	Description	Format	Example
	Faspex transfers the package to the sender from this storage location.		

RecipientShareIds	Defines extra recipients (defined by *user_name*) and their respective storage locations (defined by *share_id*). Faspex performs a package relay transfer from the initial transfer destination to the targets defined in the metadata.
	Use RecipientShareIds to control where recipients download the package from. When recipients download the package from the **Received** mailbox, Faspex transfers the package to the recipients from the specified storage locations.
		{
		  "*user_name*": *share_id*,
		  "*another_user*: *share_id*,
		  ...
		}
		Note: Value must be valid JSON.
			{
			  "admin": 4,
			  "other_user": 5
			}

OverrideShareIds	Defines additional file relays from the initial transfer destination host to designated storage locations (defined by *share_id*).	[*share_id*, ...] **Note:** Value must be valid JSON.	[1, 2, 3]

Example

share_id	Node for storage location with specified share_id
1	node1 (default inbox)
2	node2
3	node3
4	node4
5	node5

The sender (sender_user) sends a package to the recipients (recipient_user1, recipient_user2, and recipient_user3) and configures file transfers using the metadata:

- **SenderShareId** = 2
- **RecipientShareIds** = recipient_user1: 3, recipient_user2: 4
- **OverrideShareIds** = 5

Faspex performs the following transfers:

1. Faspex uploads the package directly to node2.
2. Faspex performs a package relay from node2 to node3 and node4.
3. Faspex performs a file relay from node2 to node5.

When a user downloads the uploaded package, Faspex uses the metadata to determine from which node to serve the content:

User	Downloading from	Package source node
sender_user	**Sent** mailbox	node2 (share_id: 2)
recipient_user1	**Received** mailbox	node3 (share_id: 3)

User	Downloading from	Package source node
recipient_user2	**Received** mailbox	node4 (share_id: 4)
recipient_user3	**Received** mailbox	node2 (share_id: 2)

In this scenario, the sender uploads the package to the node2 as defined by the SenderShareId, and not the server-default inbox. When recipient_user3 (who is not defined in RecipientShareIds) downloads the package, the user downloads from node2, since there is no package in the server default inbox. In this scenario, Faspex treats the share configured with **SenderShareId** as the default inbox.

Setting sender quotas

Sender quotas allow Faspex admins to control the maximum volume of data that specific Faspex users can send to specific recipients over a rolling period, based on settings at the global and user account levels.

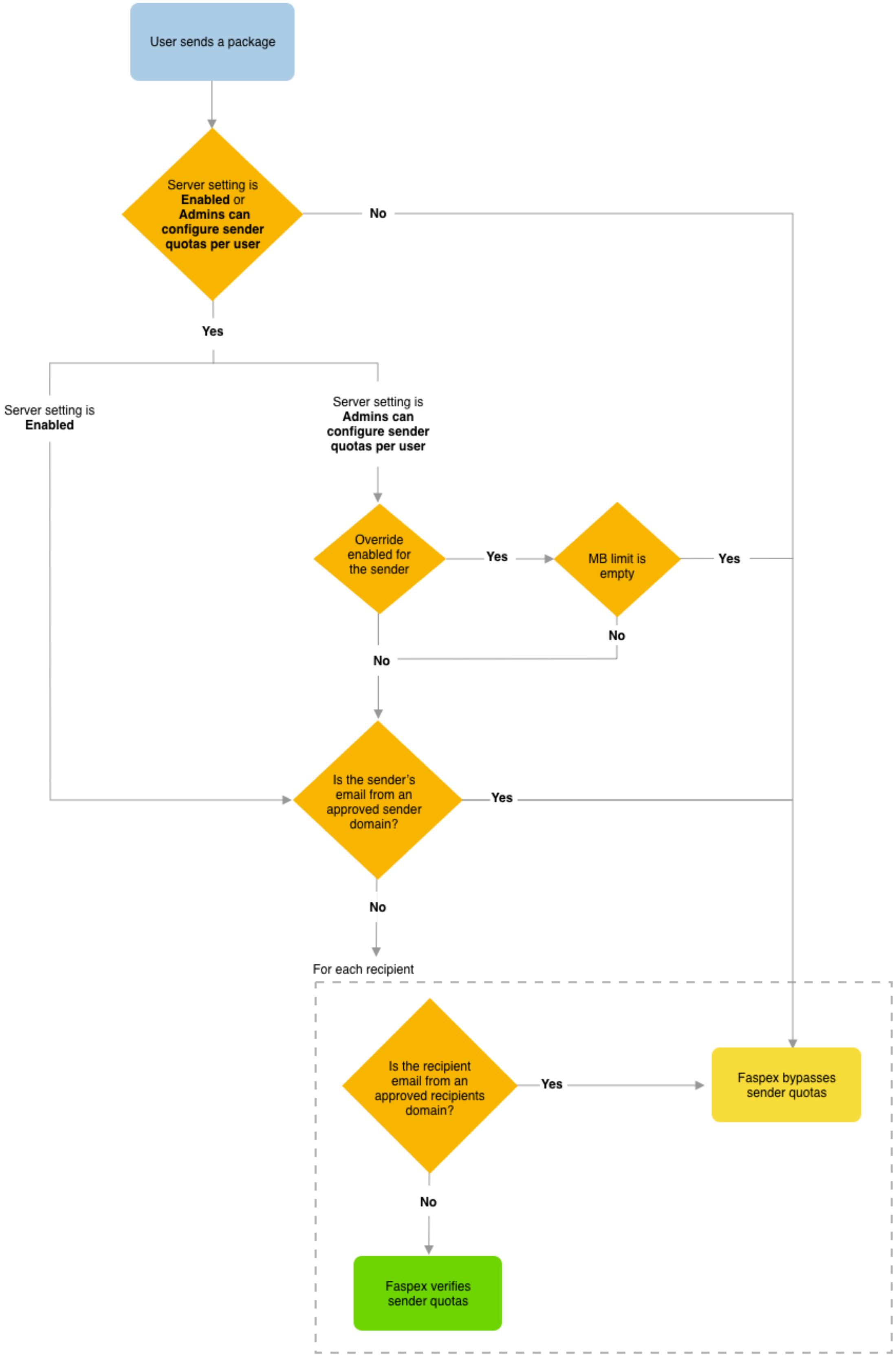
For example, admins can prevent all Faspex users from individually sending more than 25 GB over a twenty-four hour period, while allowing some of those users to send an unlimited amount of data.

Sender quotas can be applied based on who is sending, who is receiving, or both.

Senders and recipients can be exempted from sender quota enforcement based on their email address domain name.

Verification logic

Faspex uses this logic to determine whether to verify sender quotas for a package being sent:


User sends a package
Server setting is Enabled or Admins can configure sender quotas per user
No
Yes
Server setting is Enabled
Server setting is Admins can configure sender quotas per user
Override enabled for the sender
Yes
MB limit is empty
Yes
No
No
Is the sender's email from an approved sender domain?
Yes
No
For each recipient
Is the recipient email from an approved recipients domain?
Yes
Faspex bypasses sender quotas
No
Faspex verifies sender quotas

Data allowed for a new package when enforcing sender quota

If the sender quota is being verified for a package, then the maximum number of bytes (in MB) allowed for the package is defined by the global sender quota configuration and possibly the sender's sender quota configuration:

Global and user-level sender quota configurations	Package maximum size
Global sender quota = **Enabled**	Global default sender quota minus bytes already sent by that user to non-exempted recipients within the current rolling period
Global sender quota = **Admins can configure sender quotas per user** Sender's **Override default sender quota settings** is enabled and the **MB limit enforced for sending new packages** has a value greater than `0` MB	User account sender quota minus bytes already sent by that user to non-exempted recipients within the current rolling period
Global sender quota = **Admins can configure sender quotas per user** Sender's **Override default sender quota settings** is selected and the **MB limit enforced for sending new packages** has a value of `0` MB	`0` MB (no package can be sent)
Global sender quota = **Admins can configure sender quotas per user** Sender's **Override default sender quota settings** is selected and the **MB limit enforced for sending new packages** is empty	Unlimited (normally based on file storage)

Note: Faspex takes into account all user data sent within the current period (duration configured by **Hours in the rolling window** option), even if sender quotas were not yet enabled. For example, a user sends 100 MB of data in the current period. An admin then enables sender quotas with a maximum of 4000 MB. The user has 3900 MB available in the current period.

Over-Quota warnings and enforcement

When users start the process of sending a new package, Faspex warns users as soon as possible that the package might go over quota:

- If the user is over quota, Faspex notifies the user when they attempt to create a new package that they cannot send more data until the end of the current rolling period.

  Note:

  Faspex cannot determine the size of the package before its files are uploaded. Instead, Faspex relies on the HST Server node to pre-calculate the package size. Faspex uses that information to determine if senders will reach or exceed their sender quotas. The node must have the `pre_calculate_job_size` enabled (set to `yes` or `any`) for Faspex to use this information.

  If the option is disabled (set to no), Faspex determines the size of the package as the transfer goes and stops the transfer when it determines the sender has reached or exceeded the sender quota. In the worst-case scenario, Faspex might not stop a transfer until most of the package is transferred.

  Therefore, enable the `pre_calculate_job_size` setting on the HST Server node (by default, set to `any`).

- If the user is still under quota, Faspex allows the user to initiate a transfer. When the package size reaches or exceeds the sender quota, Faspex notifies the user that the user went over quota. Faspex then cancels the transfer and deletes the package.

  Note: During a transfer, Faspex checks a sender's available quota every 5 seconds. The polling frequency can allow some packages to go beyond the quota limit if Faspex does not check the in-transfer package at least once. For example, a user creates a new package while under quota and the transfer completes in under 5 seconds due to a small, package size and a high, transfer speed. Nevertheless, that package size is accounted for in the user's sender quota for any, future quota verification.

Sender quota exceptions

Admins can exempt individual accounts from sender quota enforcement by overriding the account sender quota and leaving the sender quota value empty. That requires the global sender quota setting to be set as **Admins can configure sender quotas per user**.

Admins can also choose to safelist email domains to exempt senders or recipients from sender quota enforcement based on their email addresses.

Configuring global sender quotas

Enable sender quotas globally to limit the amount of data users can send in a specified rolling window.

1. Go to **Security > Sender quota**.
2. Set the **Sender quota option** option to **Enabled** or **Default**.

Option	Sender Enforcement
Enabled	Faspex enforces the default sender quota for user accounts not included in an approved list, even if an admin sets a specific sender quota for the account.
Disabled	Faspex does not enforce sender quotas for any user accounts in Faspex, even if an admin sets a specific sender quota for the account.
Admins can configure sender quotas per user	Faspex enforces the default sender quota for all user accounts not included in an approved list. If an admin sets a specific sender quota for the account, the specific sender quota is enforced instead of the default.

3. Set the **Hours in the rolling window** in hours. Faspex resets a user account's sent bytes (in MB) after the specified duration. The value must be a positive number between `1 - 9999` hours.
4. Set the **MB limit enforced during the rolling window** to a value in megabytes (MB). The value must be a positive number.

   If the limit set to `0`, users cannot send packages.

   The default is 4000 MB.

   If the **Sender quota** option is **Admins can configure sender quotas per user**, admins can configure sender quotas for specific account.
5. Define a list of domains to exempt from sender quota enforcement. Faspex does not enforce sender quotas for:

   - Recipients with email addresses in the **Approved recipient domains**.
   - Senders with email addresses in the **Approved recipient domains**.

   For example: "@aspera.com; @ibm.com".

Note: Approved lists do not support individual email addresses. You can only exempt domain names. For example: "@aspera.com; @ibm.com".

6. Click **Save**.

Set a specific sender quota for a user account

If the **Sender quota** option is **Admins can configure sender quotas per user**, admins can configure sender quotas for specific account. You can change the quota limit, but the quota rolling period is defined globally.

1. Go to **Users > All users**.
2. Select a user.
3. In the Package sending section, enable **Override default sender quota settings**.
4. Set the **MB limit enforced for sending new packages** to a value in megabytes (MB). The value must be a positive number.

   If the limit set to `0`, users cannot send packages.

   The default is 4000 MB.

Job health monitoring

Monitor the status of package, email, SAML, node, and relay jobs.

You can check the state of your background processes by going to **Job health > Job health**.

You can check the state of your background jobs by going to **Job health > Job queues**. On this page, you can delete jobs in queue.

If you are a system admin, you can check the state of your database by logging into the Faspex Utility application.

Configuring API clients

Faspex requires your application use OAuth 2 to authorize your application to access protected Faspex resources.

OAuth 2 methods

Before your application can authorize to Faspex using OAuth 2, you must first register an API client for your application. Faspex supports these OAuth 2 methods:

OAuth 2 with SAML
Web applications requiring users to authenticated through a configured SAML identity provider (SAML IdP). For details, see “Working with SAML” on page 113.

OAuth 2 PKCE (Proof Key for Code Exchange)
Web and mobile applications requiring users to enter credentials into a user login page, which then authenticates to the Faspex server. For details, see “Configuring OAuth 2 for user-based workflows” on page 192.

Note: The Faspex UI acts as an OAuth 2 client to authenticate to the Faspex API server. The Faspex UI is a pre-registered OAuth 2 client.

OAuth 2 JWT (JSON Web Token Grant)
Non-web applications that do not require access to user-protected endpoints, such as an application that monitors background jobs. For details, see “Configuring OAuth 2 for non-user-based workflows (JWT)” on page 195.

To register an OAuth 2 client:

1. Go to **Configurations > API clients**.

2. Click **Create new**.
3. Fill out the form.

Field	**Description**	**Required for OAuth 2**	**Required for OAuth 2 with SAML**	**Required for OAuth 2 PKCE**	**Required for OAuth 2 JWT**
Name	Name to differentiate the API client in Faspex.	X	X	X	X
Allow implicit grant	Allow an application to get an access token without an intermediate code exchange step.	X			Not available when **Enable JWT grant type** is enabled.
Redirect URIs	List of allowed URIs that the API client can redirect an application to (designated in application API call)	X	X	X	
Origins	List of URIs or protocol-host-port of the client app initial login page, from which the user must arrive to the authentication flow.	X			
Access token expiration	The maximum duration of an active session unless the refresh token duration (see following parameter) is configured to extend the session. Without refresh tokens, users must re-authenticate when the login token expires.	X	X	X	X

Field	Description	Required for OAuth 2	Required for OAuth 2 with SAML	Required for OAuth 2 PKCE	Required for OAuth 2 JWT
All users can use this client to access the API	Enable to allow the API client to request access as any Faspex user. Disable to allow the API client to request access as specifically listed users only.				
Allow refresh token	Defines the maximum duration that an active login session can be extended.				
Key	Public key (in .pem format) used to verify a JWT payload.	X	X		X

OTFV-Validator

OTFV is a Golang service that works as a file validator. It processes files to verify and validate their content. Examples of this could include virus scanning, file name checking or package names.

Setting up the OTFV API client

Instructions for setting up the OTFV API client in Faspex 5.

1. Install Faspex 5.0.7.
2. Complete the Faspex 5.0.7 setup, add the HSTS nodes, and confirm transfers are working properly.
3. Create an API client public/private key par for OTFV:

```
openssl genrsa -out private_key.pem 4096
openssl rsa -in private_key.pem -pubout -out public_key.pem
```

4. Create a JWT API client for OTFV and add the `public_key.pem` that was just generated. For more information, see the "Configuring API clients" on page 131 section.

Installing OTFV

Step-by-step instructions on how to install OTFV.

1. Download the `ibm-aspera-otfv-validator.x86_64.rpm` package and install it. You can install OTFV in the same machine where HSTS is installed.

   Warning: OTFV is not compatible with previous versions of Faspex. You should perform a new install of OTFV 1.0.4. Upgrading from a previous OTFV version is not supported.

2. Verify that there are no previous versions of OTFV installed:

```
rpm -qa | grep -n validator
```

To uninstall a previous version of OTFV run:

```
rpm -e ibm-aspera-otfv-validator
```

3. Edit the `/opt/aspera/aspera-validator/config/validator.yaml` file and fill out the values under `Faspex Settings`. For more information refer to the Table 3 on page 135 table in the Configuration reference section.

   - Set the `LuaFilePath` to the path of the Lua script that Validator runs on every transferred file.

     Important: Although OTFV validator includes example scripts for guidance, it is the user's responsibility to provide the Lua script. The example scripts are not meant to be run out-of-the-box without proper configuration for your environment and your use case.
4. Add the certificate authority (CA) to the machine running OTFV. You have to add it to the CA-Bundle on the Linux host in order to be trusted. You must do this for OTFV to work properly.
5. Start the OTFV validator service. If you want to view or edit the validator logs you can find the log file under

   ```
   service aspera-validator start
   ```

   ⚠ **Attention:** Verify that there are no HTTPS / certificate issues during startup. If you encounter any, resolve them before enabling files processing in the system's UI.
6. To view the validator tail logs run:

   ```
   tail -f /var/log/validator.log
   ```

Configuration reference

Reference information, descriptions and configuration values.

validator.yaml

Table 1. Global settings

Value	**Description**	**Default**
Debug	Set to `true` to output more information.	false
LogConfig		
Filename	Specifies the file path and name for the log file generated by the validator.	`/var/log/validator.log`
EnableLogRotation	Determines whether log rotation is enabled for the log file.	true
MaxSizeMB	Sets the maximum size, in megabytes (MB), that the log file can reach before triggering a rotation.	100
MaxFilesRetained	Defines the maximum number of rotated log files that are retained before the oldest files are deleted.	10

Table 1. Global settings (continued)

Value	Description	Default
MaxAgeDays	Specifies the maximum number of days that a log file is retained before it is deleted.	60
Compress	Determines whether the rotated log files are compressed to save disk space.	true
UseLocalTime	Specifies whether the timestamps in the log files are based on the local system time or UTC.	true
Format	Defines the format of the log entries in the log file.	text
FileQueueLimit	The number of files that will be requested from Faspex at once.	25
UpdateBatchSize	The number of file processing results that will be sent back in one call to Faspex.	25

Table 2. Lua Settings

Value	Description	Default
MaxRoutines	The number of Lua Script workers that will run at once (Lua threads).	8
LuaFilePath	Set the path to the Lua script Validator runs on every transferred file.	`/opt/aspera/aspera-validator/examples/basic.lua`

Table 3. Faspex Settings

Value	Description	Default	Example
FaspexFileProcessingEnabled	Determines whether the file processing functionality is enabled.	true	
FaspexURL	Specifies the URL of the Faspex server or platform that the validator will interact with.		HTTPS://FQDN.COM
FaspexAPIClientID	The ID obtained after registering the OAuth Client through the Faspex UI.		“<OTFV_API_CLIENT_ID>”
FaspexAPIClientPrivateKeyFile	Key from registering the Faspex API Client Id. Stored as a path to the key file.		“<FULL_PATH_TO_PRIVATE_KEY>”

Table 4. Env variable

Variable	Description
VALIDATOR_CONFIG_PATH	Points to the directory for the `validator.yaml` and `node_servers.yaml` file configurations.

node_servers.yaml

List of servers to poll for validation. This is optional.

Table 5. Servers

Value	Description
${NODE_NAME}	Name of the node
${NODE_URL}	URL of the node
${NODE_USER}	Username to access the node
${NODE_PASSWORD}	Password to access the node

Lua file

This section provides information and details about functions, environments and processing statistics related to the Lua script interaction with the Faspex 5 API.

Table 6. Global settings

Function	Description	Additional information
update_status(status, bytes_processed, err_msg)	Sends the status of a package	
update_status("completed", bytes_processed)	Mark a package as completed	You can omit `err_msg` on success
update_status("error", bytes_processed, "Custom error message")	Mark a package as failed	If an error occurs during the processing of the file, the file will be marked as failed and the error message will be used as the custom error message
aslog(string)	Takes one argument and logs the string to `validator.log`	
url_path(path)	Removes the domain and the extra parameters from the URL path.	

Table 6. Global settings (continued)

Function	Description	Additional information
pkg_env(pkg_id)	Takes one argument which is the packageID. Calling pkg_env(pkg_id) could potentially present some delay since it makes an HTTP request. **Note:** Shared Inboxes and Work Groups are not currently supported in lua scripts package details.	local f = env["Path"] local bytes_processed = env["Size"] local penv = pkg_env(env.PackageId) for i,v in pairs(penv) do print(i,v) end if f:match(".*failme.*") then local err_msg = "bad file name" update_status("error", bytes_processed, err_msg) else update_status("completed", bytes_processed) end
External Libraries		
lfs = require("lfs")	LuaFileSystem complements the set of functions related to file systems offered by the standard Lua distribution.	local lfs = require 'lfs' for f in lfs.dir '.' do print(f) end
socket = require("socket")	LuaSocket is an extension library that is composed by two parts: a C core that provides support for the TCP and UDP transport layers, and a set of Lua modules that add support for functionality commonly needed by applications that deal with the Internet.	local http = require("socket.http") print(http.request("http:// example.com"))

json = require("json")	Instructs Lua to load and include the JSON library into the script.	local json = require("json")
		local jstr =
		json.encode({example="yes"})
		local data =
		json.decode(jstr)
		print(data.example)

mysql = require("mysql")	Instructs Lua to load and include the MySQL library into the script.	local mysql = require('mysql') local c = mysql.new() local ok, err = c:connect({ host = '127.0.0.1', port = 3306, database = 'test', user = 'user', password = 'pass' }) if ok then local res, err = c:query('SELECT * FROM mytable LIMIT 2') dump(res) end

Table 7. Env

Attribute	Value type	Description
BytesContiguous	number	
ByteLost	number	Number of bytes lost during the transfer
BytesWritten	number	Number of bytes written
BytesProcessed	number	Number of bytes Processed
Elapsed	number	
Errorcode	number	Error code when transferring should be `0`
FileId	string	Unique ID for a file in a package
NodeId	string	The ID of the node the file has been transferred to
Path	string	Path to the file
SessionUuid	string	The UUID of the session
Size	number	The size of the current file being processed
StartOffset	number	
Status	string	Status of the package should be:`transient_file_processing`
Message	string	
UpdatedAt	string	Time of last updated formatted: `yyyy-mm-ddThh:mm:ssZ (2024-01-10T18:49:25Z)`
FileName	string	Name of the file being processed
ErrorDescription	string	
PackageId	string	Id of the package that the file is part of. Used in `pkg_env` function
PackageUuid	string	UUID of the package

Table 8. PENV

Attribute	Value type	Description
ID	string	ID of the package
Title	string	Name of the package
ReleasePolicy	string	
ReleaseDate	string	Formatted: `yyyy-mm-ddThh:mm:ssZ (2024-01-10T18:49:25Z)`

Table 8. PENV (continued)

Attribute	Value type	Description
State	string	State of package should be: `transient_file_processing_needed`
TotalBytes	number	Total Bytes in the package
TotalFiles	number	Total Files in the pacakge
Message	string	
UploadStatus	string	Upload status for the package can be: `completed(` or `uploading`
CreatedTime	string	Formatted: `yyyy-mm-ddThh:mm:ssZ` `(2024-01-10T18:49:25Z)`
Sender	table	Refer to sender table
PackageUUID	string	UUID of the package
ExpirationPolicy	string	What the expiration policy is
ErrorDesc	string	
FilesOnServer	string	If the files are or aren't on the server. Can be: `yes` or no
EarEnabled	boolean	
FileProcessingStatus	string	Status of File Processing should be: `to_be_processed`
FileProcessingMsg	string	
Metadata		Table that contains custom data with a variable number of fields. It's equal to the value set in the UI for metadata.
Recipients		List of recipients
PrivateRecipients		List of private recipients
Recipient_table		
Recipient_type	string	Indicates the type of recipient
ID	number	Identifier of the recipient
Name	string	Username of the recipient
FirstName	string	First name of the recipient
LastName	string	Last name of the recipient
Email	string	Email address associated with the recipient

Table 9. Sender table

Attribute	Value type	Description
ID	string	
LastLoginTime	string	Formatted: `yyyy-mm-ddThh:mm:ssZ (2024-01-10T18:49:25Z)`
Status	string	If the sender is active or not. Should be `active`
Name	string	Name of the sender, can be an email
FirstName	string	First name of the sender
LastName	string	Last name of the sender
Email	string	Senders Email
AccountActivated	boolean	If the account is active or not
CanSendNormalPackages	string	If this sender can send packages: most likely `yes` or `default`
Role	string	The role of the sender
Type	string	The type of user

Table 10. Unprotect library

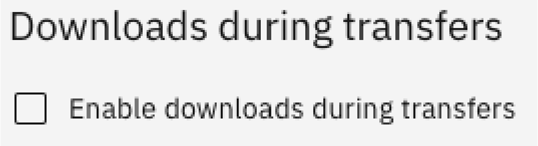
Value	Description
unprotect	Module required for file decryption
faspex_package_path	Local path to the packages
password	Password for file decryption
path	The combined path of the Faspex package directory and the file's URL path
err	Variable that stores error messages
file	File handle returned by `unprotect.open` on success it returns: `file_handle, nil`, on error it returns: `nil,error_msg`
content	The content read from the decrypted file

Enabling file processing

When file processing is enabled Faspex puts packages into a transient state (the UI shows files as `uploaded for file processing`) and waits for a third-party application (including IBM Aspera otfv-validator, IBM Aspera Orchestrator, or any other application that can integrate with Faspex 5 API) to successfully process all the files; if file processing on a package fails for one or more files, Faspex will not deliver the package. File processing occurs simultaneously with file uploads to ensure efficient data management.

Warning: Do not proceed with these steps until OTFV is properly configured and running with no errors. For instructions on how to install and configure OTFV see the "Installing OTFV" on page 133 section. File processing requires HSTS 4.4.3 patch level 3 or later, download and apply the patch before enabling this feature.

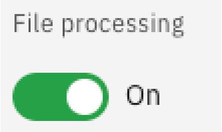
1. Go to **Admin > Configurations > Transfer options** and disable the **Enable downloads during transfers** option.



Downloads during transfers

□ Enable downloads during transfers

2. Go to **Admin > Package processing > File processing**.
3. Switch the **Files processing** toggle to **On**. **Timeout settings** are displayed.

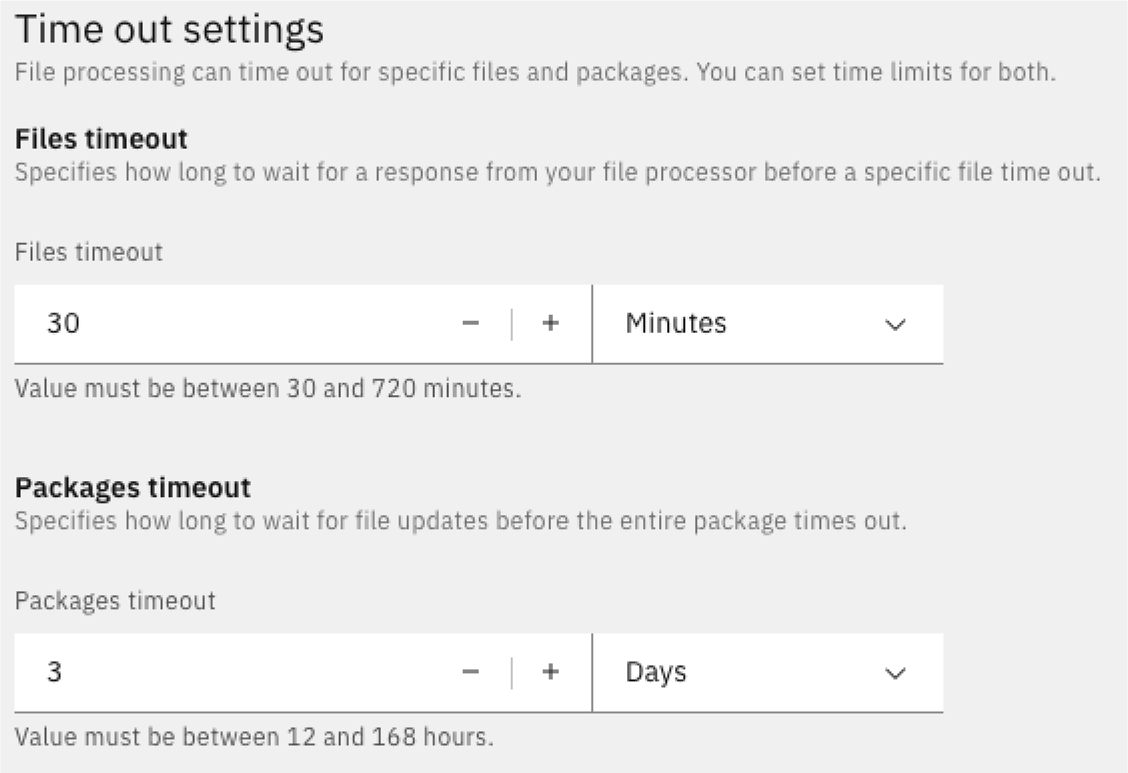


File processing

On

4. Set the timeout duration. For files timeout the valid values are between 30 and 720 minutes. For Packages timeout the valid values are between 12 and 168 hours.

   Faspex must receive an update from the third-party system to the status of at least one file in the package before the configured timeout duration.



Time out settings

File processing can time out for specific files and packages. You can set time limits for both.

Files timeout

Specifies how long to wait for a response from your file processor before a specific file time out.

Files timeout

30 − | + Minutes

Value must be between 30 and 720 minutes.

Packages timeout

Specifies how long to wait for file updates before the entire package times out.

Packages timeout

3 − | + Days

Value must be between 12 and 168 hours.

5. Click **Save**.

Configuring package processing webhooks

Admins can create one or more webhooks to notify external systems (using a POST call) when a package is uploaded. For example, a webhook request can initiate an IBM Aspera Orchestrator workflow. Configure filters to execute the applicable webhook for each uploaded package. Faspex does not process to any response to the POST request.

CAUTION: If a Faspex Administrative account is compromised, package processing can be a serious threat to your server security. Update all administrative user permissions to prevent unauthorized users from executing package processing by restricting the IP addresses from which a user can log in to an admin account. For more information, see Configure User Settings.

Here is a sample POST request.

```
{
"faspex_pkg_directory":"/package test - 61ae5292-d8ef-496e-8b15-8cdc0379fc4a.aspera-package/PKG
- package test",
"faspex_pkg_name":"Test files",
"faspex_pkg_note":"",
"faspex_pkg_id":506,
"faspex_pkg_delivery_id":506,
"faspex_recipient_list":"aspera-recipient-test@ibm.com",
"faspex_recipient_count":1,
"faspex_sender_id":30,
"faspex_sender_name":"aspera-sender-test@ibm.com",
"faspex_sender_email":"aspera-sender-test@ibm.com",
"faspex_pkg_total_bytes":4633,
"faspex_pkg_total_files":1,
"faspex_pkg_uuid":"61ae5299-d8ea-496e-8b15-8cdc0379fc4a",
"faspex_metadata_fields": "{\"_pkg_uuid\":\"61ae5292-
d8ef-496e-8b15-8cdc0379fc4a\",\"_pkg_name\":\"Test
files\",\"_created_utc\":\"2022-1209T17:47:03.545Z\"}",
"faspex_recipient_0":"aspera-sender-test@ibm.com"
}
```

1. Identify the endpoint of the external system for the webhook to notify.
2. In the Faspex web UI, go to **Admin > Package processing > Create new webhook**.
3. Configure the webhook.

Field name	Description
Name	A unique, descriptive label for this webhook.
Webhook endpoint URI	The full URI to the external system.
HTTPS	Create a secure connection to the URI.
Webhook connection timeout	Defines (in seconds) how long the system should wait for URI connection before the attempt times out. In the event of a connection timeout, Faspex retries the connection once. Valid values: 1 to 600; default 60.
Active	To enable this webhook, click the toggle to **On**. To disable, click the toggle to **Off**.

4. Configure the filters.

   All configured filters must match the uploaded package attributes for the webhook to activate for that package. All fields in this section are optional. When you select **Exact match**, the package attribute must match the configured filter exactly to activate the webhook. If you don't select **Exact match**, the text you enter can match any part of the package attribute.

Filter name	Apply the webhook when...
Package name	The package name matches the string.
Sender username	The sender username matches the string.
Sender email	The sender email address matches the string.
Recipient usernames	The recipient usernames match the string. Workgroups and shared inboxes are not supported as recipients.
Recipient emails	The recipient email addresses match the string. Workgroups and shared inboxes are not supported as recipients.
Package message	The package note matches the string.

Filter name	Apply the webhook when...
Package start date Package end date	The package date falls into the configured range.
Minimum package size Maximum page size	The package size (in bytes) falls into the configured range.

5. Click **Create**.

Monitor package processing

Click **Jobs** for an activity list. Filter as required per webhook or status.

Go to **Admin > Job health > Webhook jobs** for details. See also "Job health monitoring" on page 131.

Managing the Faspex database

Use Faspex Utility to manage the Faspex database

Migrating the database

After an upgrade or restoring the database from a backup, you may need to migrate your database to the latest schema.

1. Log in to Faspex Utility.
2. Go to **Manage database**.

In the **Faspex Core schema state** area:

3. Select the migration you want to update to and click **Migrate**, or click **Migrate latest**.

In the **Faspex Service schema state** area:

4. Select the migration you want to update to and click **Migrate**, or click **Migrate latest**.

Seeding the database

You can seed the database when performing the following actions:

- Creating basic Faspex data
- Creating an initial admin user
- Updating the Faspex redirect URLs

Note: You don't need to run the seed operation in utility after setting up Faspex. However, it can be useful for users that need to update the `FASPEX_CORE_URL` variable to match the value provided in the `core.env` file.

1. Log in to Faspex Utility.
2. Go to **Manage database**.

In the **Database Seeding** area:

3. If the Faspex database has not been seeded or set up properly before, provide the **Email**, **Password** and **Password confirmation**.

   Note: Database Seeding is allowed to create insecure passwords.
4. Click **Seed**

Backing up the database

Use the Faspex Utility application to manually back up your database or use a Docker command to do it from the command line.

Automated regular backups

You can run a Docker command to dump the Faspex database onto your computer:

```
docker exec faspex-utility /usr/bin/mysqldump faspex > /tmp/backup.sql
```

Set up a **cron** job to backup your database regularly.

To manually back up your database using the Faspex Utility application:

1. Log in to Faspex Utility.
2. Go to **Manage database**.

In the **Backup or restore the database** area:

3. Select **Backup**.
4. Select the backup method:
   - **Store backup on the server**: The path for the backup is defined by the FASPEX_UTILITY_DB_BACKUP_MOUNT variable in the /opt/aspera/faspex/conf/docker/utility.env file. By default, the path is /opt/aspera/faspex/data/db_backups
   - **Download backup to your computer**
5. Click **Backup**.

Restoring the database

Restore a previous version of the Faspex database using the Faspex Utility application.

1. Log in to Faspex Utility.
2. Go to **Manage database**.

In the **Backup or restore the database** area:

3. Select **Restore**.
4. Select **Upload a database backup file** or select a previously created backup already stored on the faspex server

   If you selected **Upload a database backup file**, upload a database backup file.

   Note: You can restore a backup file generated by another Faspex instance by copying the backup file into the FASPEX_UTILITY_DB_BACKUP_MOUNT folder set in /opt/aspera/faspex/conf/docker/utility.env. The folder is, by default, /opt/aspera/faspex/data/db_backups.
5. Click **Restore**.

Troubleshooting

Use diagnostics tools to self-service in the best-case scenario and collect crucial information for IBM Support in the worst-case scenario.

Recommended steps

1. Check the container logs for insight into what went wrong.
2. Gather information to send to IBM Support.

Viewing application information

Find information about the application using UI and command-line tools.

Checking component versions

Run `faspexctl version` to see the versions of running containers and the version of MariaDB and Nginx running in containers.

Application information window

You can find useful information about the application in the application information window. When working with Support, Support may ask you to open this window to assist with troubleshooting.

You can access this window by holding down the **Option** key on your keyboard and clicking your profile icon and selecting **Application information** from the drop-down menu.

Checking container version

Run `faspexctl version` to check the version of the containers. If a container has been patched, the command will also report the patch number as:

```
Fix Pack: PATCH_VERSION
```

Viewing container logs

Viewing logs

Use **`docker logs`** *`service`* to access the logs for a container.

Setting log levels

You can configure log levels for Faspex by adding the corresponding variable to each `.env` file(s) located in `/opt/aspera/faspex/conf/docker/` For example: `FASPEX_SERVICE_LOG_LEVEL=debug`

Variable	.env file
FASPEX_UTILITY_LOG_LEVEL=debug	`utility.env`
FASPEX_CORE_LOG_LEVEL=debug	`core.env`
FASPEX_SERVICE_LOG_LEVEL=debug	`service.env`
FASPEX_ROUTER_LOG_LEVEL=debug	`router.env`
FASPEX_UI_LOG_LEVEL=debug	`ui.env`

Supported log levels

You can set the desired log level to filter out messages. They are listed in increasing order of severity.

`FASPEX_SERVICE` supported log levels:

- debug
- info
- warning
- error
- fatal
- panic

`FASPEX_CORE` supported log levels:

- debug
- info
- warning
- error
- fatal

Redirecting container logs to `syslog`

While you can use the **`docker logs`** *`service`* command to access the logs for a container, you can also configure Docker to redirect output to system logs.

1. Edit or create the `/etc/docker/daemon.json file.`

   ```
   {
     "log-driver": "syslog",
     "log-opts": {
       "syslog-address": "unixgram:///dev/log",
       "tag": "docker/{{.Name}}"
     }
   }
   ```

2. Restart the docker daemon:

   ```
   systemctl restart docker
   ```

3. Edit the `core.yml` and `service.yml` files located at `/opt/aspera/faspex/conf/docker_compose_templates`:

   a. Change the `driver` value to `syslog` in the `docker-compose` YAML file of any Faspex container.

   b. Remove the `max-size: "10m"`and `max-file: "10"options` under the `driver` parameter.

   For example, after editing `core.yml` and `service.yml` :

   Note: This template is only an example, you should not copy and paste it.

   ```
   # cat core.yml
     core:
       image: icr.io/ibmaspera/faspex-core:5.0.5
       container_name: faspex-core
       env_file: [ docker/core.env, docker/db.env ]
       environment: [ SECRET_KEY_BASE=$FASPEX_CORE_SECRET,
   OLD_SECRET_KEY_BASE=$FASPEX_CORE_OLD_SECRET ]
       ports: ["$FASPEX_CORE_PORT:$FASPEX_CORE_PORT"]
       logging:
         driver: "syslog"

       networks: [ docker_network ]
       depends_on: [ db ]

   ######################################################################

   # cat service.yml
     service:
       image: icr.io/ibmaspera/faspex-service:5.0.5
       container_name: faspex-service
       env_file: [ docker/service.env, docker/db.env, docker/core.env ]
       ports: ["$FASPEX_SERVICE_PORT:$FASPEX_SERVICE_PORT"]
       logging:
         driver: "syslog"

       networks: [ docker_network ]
       depends_on: [ core ]
   ```

4. Run **`faspexctl setup`** to implement the changes.

5. Read the `/var/log/messages` log file to see container logs.

   Faspex containers begin log entries with `docker/faspex_`*`container_name`*. For example, the Faspex core container begins log entries with `docker/faspex-core`.

You can use **grep** and the container name to filter for or filter out specific log entries.

Filter for the core container:

```
tail -f /var/log/messages | grep "docker/faspex-core"
```

Filter out the core container:

```
tail -f /var/log/messages | grep -i "docker/faspex-core"
```

Creating and updating user accounts without logging in

You can use the Utility app to create a new admin account, to change the password of an existing admin account, and to change the password of an existing user account.

1. Log in to Faspex Utility.
2. Go to **Manage users**.

In the **Update credentials** area:

3. Enter the email address of the user you want to create or update.
4. Enter and confirm the new password.
5. Click **Update**.

Uninstall Faspex

In some cases, you may have to uninstall and reinstall Faspex to fix a problem.

There are two types of Faspex uninstalls: simple uninstall and deep uninstall. Before performing either process, fully back up the system (see "Back up your Faspex instance" on page 147). With a backup, you can reinstall Faspex without losing data or migrate the Faspex instance to another server.

Simple uninstall

To perform a simple uninstall, run:

```
rpm -e ibm-aspera-faspex
```

Deep uninstall

When you uninstall Faspex, some settings are preserved to make re-installation easier. To remove Faspex completely:

1. Stop Faspex:

   ```
   faspexctl stop
   ```

2. Prune docker containers, networks, images, and build cache by running:

   ```
   faspexctl prune
   ```

3. Uninstall the Faspex package by running:

   ```
   rpm -e ibm-aspera-faspex
   ```

4. Delete the `/opt/aspera/faspex` folder.

Back up your Faspex instance

With a backup, you can reinstall Faspex without losing data or migrate the Faspex instance to another server.

1. Use Faspex Utility to back up the database.
2. Back up your Faspex database backups and your Faspex configuration files:

The database backups are stored at `/opt/aspera/faspex/data/db_backups`.

The configuration files are at:

- **Docker environment files**: `/opt/aspera/faspex/conf/docker`
- **Database configuration files**: `/opt/aspera/faspex/conf/db`
- **Nginx configuration files**: `/opt/aspera/faspex/conf/nginx`
- **Docker-compose files**: `/opt/aspera/faspex/conf/docker_compose_templates`

You can use this simple command to copy all these files to a backup folder:

```
mkdir -p faspex_backup/conf/docker; mkdir -p faspex_backup/data; cp /opt/aspera/faspex/conf/
docker/*.env faspex_backup/conf/docker; cp -r /opt/aspera/faspex/conf/db /opt/aspera/faspex/
conf/nginx /opt/aspera/faspex/conf/docker_compose_templates faspex_backup/conf/; cp -r /opt/
aspera/faspex/data/db_backups faspex_backup/data/
```

Restore or migrate a Faspex instance from backups

Restore or migrate Faspex to another server.

1. Perform a full back up of your Faspex instance.
2. Reinstall Faspex or install Faspex on a new server following the steps in "Installing Faspex for the first time" on page 7.
3. Restore configuration files from the previous instance.

   The configuration files are at:

   - **Docker environment files**: `/opt/aspera/faspex/conf/docker`
   - **Database configuration files**: `/opt/aspera/faspex/conf/db`
   - **Nginx configuration files**: `/opt/aspera/faspex/conf/nginx`
   - **Docker-compose files**: `/opt/aspera/faspex/conf/docker_compose_templates`

   Important: Faspex requires the values of `FASPEX_CORE_SECRET`, `FASPEX_CORE_OLD_SECRET`, `FASPEX_CORE_UI_CLIENT_ID`, and `FASPEX_SERVICE_CLIENT_ID` to match the values set in the database. Make sure you use the values matching those in your database backup.

   Tip: If you used the sample command from "Back up your Faspex instance" on page 147, you can run this command to copy all backed up files to their correct locations:

   ```
   cp faspex_backup/conf/docker/*.env /opt/aspera/faspex/conf/docker/; cp faspex_backup/
   conf/db/* /opt/aspera/faspex/conf/db; cp faspex_backup/conf/nginx/* /opt/aspera/faspex/
   conf/nginx/; cp faspex_backup/conf/docker_compose_templates/* /opt/aspera/faspex/conf/
   docker_compose_templates/
   ```

4. Run **`faspexctl setup`** to restart Faspex with the restored configuration.
5. Restore the Faspex database:

   a) Copy database backup files into `/opt/aspera/faspex/data/db_backups`.

   b) Restore the database using the Faspex Utility application.
6. In the Faspex Utility application, migrate the Core and Service databases to latest schemas.

Optimizing Faspex performance

Increasing Faspex Puma workers

You can increase the number of Puma workers running on the API server (each worker is a separate OS process) by configuring the `FASPEX_CORE_CONCURRENT_WORKERS` variable in the `/opt/aspera/faspex/conf/docker/core.env` configuration file. Base the number of workers on the cores available on your machine.

Propagate my.cnf changes during Faspex upgrade

With the changes made to the `Faspex-db` container regarding a new Transport Layer Security (TLS) configuration, the contents of the `/opt/aspera/faspex/conf/db/my.cnf` file will be updated with SSL path variables for the container, which will contain the necessary certificate and key files for the new encryption method.

If you make any changes to the `my.cnf` file before the Faspex upgrade, they will be carried over to the new `my.cnf` file. However, if there are issues with the `Faspex-db` container during the upgrade, you can verify that all variables are correct and properly placed in the new `my.cnf` file.

References

Reference: Faspex configuration files

File locations

- **Docker environment files**: `/opt/aspera/faspex/conf/docker`
- **Database configuration files**: `/opt/aspera/faspex/conf/db`
- **Nginx configuration files**: `/opt/aspera/faspex/conf/nginx`
- **Docker-compose files**: `/opt/aspera/faspex/conf/docker_compose_templates`

Faspex API server

`/opt/aspera/faspex/conf/docker/core.env`

Variable	Usage	Default
`FASPEX_CORE_SECRET`	Production secret	Automatically generated
`FASPEX_CORE_OLD_SECRET`	Production secret	Automatically generated
`FASPEX_CORE_URL`	Use as the primary hostname of the server	Provided to **`faspexctl`** at install
`FASPEX_CORE_UI_CLIENT_ID`	API client ID used by `faspex-ui` to authorize users to Faspex	Automatically generated
`FASPEX_CORE_CONCURRENT_WORKERS`	The number of Puma workers running on the API server (each worker is a separate OS process). Increase the number of workers to increase the level of parallel processing for this container. Base the number of workers on the cores available on your machine	2
`FASPEX_CORE_LOG_LEVEL`	Provides extra debug logging in the Core logs. Setting the value to debug enables it	

Faspex database

Faspex 5 uses three database users:

- A root-level user for creating the database and for provisioning the other two users. Faspex 5 only uses this user when configuring the `faspex-db` container.
- An admin user for updating, creating, and dropping tables. This user is restricted to the Faspex 5 Utility application for migrations.
- A restricted user for performing production read and write operations.

`/opt/aspera/faspex/conf/docker/db.env`

Environment variables used to connect to the local or remote database and username and password for the restricted user:

⚠ **CAUTION:** Do not exceed the 80 character limit when creating passwords. This may cause issues during setup or when trying to perform migrations.

Variable	Usage	
`FASPEX_DB_HOST`	Hostname of the local or remote database	db (local container)
`FASPEX_DB_PORT`	Port of the local or remote database	`4406`
`FASPEX_DB_USERNAME`	Username of the database restricted user	`faspex`
`FASPEX_DB_PASSWORD`	Password of the database restricted user. Do not include the dollar sign ($) in this password, it is not supported. The following characters are supported: single quotes ('), ampersand (&), and backslash (/).	Automatically generated
`FASPEX_DB_MOUNT`	The folder for database data	`../data/db`
`FASPEX_DB_CONF_MOUNT`	The folder for database configuration files	`../conf/db`
`FASPEX_DB_CORE_DATABASE`	Name of the Faspex database	`faspex`

`/opt/aspera/faspex/conf/docker/db_admin.env`

Usernames and passwords for the root-level user and admin user:

Variable	Usage	
`FASPEX_DB_ASPERA_ADMIN_USERNAME`	Username of the database admin user	`aspera`
`FASPEX_DB_ASPERA_ADMIN_PASSWORD`	Password of the database admin user. Passwords cannot start with special characters, you should use alphanumeric characters instead. Do not include the dollar sign ($) in this password, it is not supported. The following characters are supported: single quotes ('), ampersand (&), and backslash (/).	Automatically generated

Variable	Usage	
FASPEX_DB_ROOT_USERNAME	Username of the database root-level user	root
FASPEX_DB_ROOT_PASSWORD	Password of the database root-level user. Passwords cannot start with special characters, you should use alphanumeric characters instead. Do not include the dollar sign ($) in this password, it is not supported. The following characters are supported: single quotes ('), ampersand (&), and backslash (/).	Automatically generated

Faspex UI

/opt/aspera/faspex/conf/docker/ui.env

Variable	Usage	Default
FASPEX_UI_LOG_LEVEL	Provides extra debug logging in the UI logs. Setting the value to debug enables it	

Faspex Utility

/opt/aspera/faspex/conf/docker/utility.env

CAUTION: Do not exceed the 80 character limit when creating passwords. This may cause issues during setup or when trying to perform migrations.

Variable	Usage	Default
FASPEX_UTILITY_USERNAME	The username of the Utility admin user	admin
FASPEX_UTILITY_PASSWORD	The password of the Utility admin user	Automatically generated
FASPEX_UTILITY_DB_BACKUP_MOUNT	The folder where the Utility application writes and reads database backups	../data/db_backups
FASPEX_UTILITY_LOG_LEVEL	Provides extra debug logging in the Utility logs. Setting the value to debug enables it	

Faspex Router

/opt/aspera/faspex/conf/docker/router.env

Variable	Usage	Default
FASPEX_ROUTER_CORE_URL	The URL and port the Router redirects traffic meant for the core container	core:3000

Variable	Usage	Default
FASPEX_ROUTER_UI_URL	The URL and port the Router redirects traffic meant for the core container	ui:5000
FASPEX_ROUTER_SERVICE_URL	The URL and port the Router redirects traffic meant for the core container	service:6000
FASPEX_ROUTER_UTILITY_URL	The URL and port the Router redirects traffic meant for the core container	utility:4000
FASPEX_ROUTER_CONF_MOUNT	The folder where the Router reads Nginx configurations	../conf/nginx
FASPEX_ROUTER_HTTP_PORT	The HTTP port for Faspex application	80
FASPEX_ROUTER_HTTPS_PORT	The HTTPS port for Faspex application	443
FASPEX_ROUTER_LOG_LEVEL	Provides extra debug logging in the Router logs. Setting the value to debug enables it	

Faspex Service

/opt/aspera/faspex/conf/docker/service.env

Variable	Usage	Default
FASPEX_SERVICE_CLIENT_ID	API client ID used by faspex-ui to authorize background jobs to Faspex	Automatically generated
FASPEX_SERVICE_LOG_LEVEL	Provides extra debug logging in the service logs. Setting the value to debug enables it	

Reference: Email notification templates

Note: All datetime values are displayed in UTC time.

Packages

Template name	Email-triggering event	Email recipients
Package received	Faspex successfully sends a package to the package recipients.	Any recipient that has the **Email me when I receive a package** setting (enabled by default) enabled in the recipient's user preferences
Package received CC	Faspex successfully sends a package to the package recipient.	Users included in the **When a recipient receives the package** field in the Send package form
Package downloaded	Faspex sends this email to users when a sent package has been downloaded.	The sender, if the sender has the **Email me when I download a package** setting (disabled by default) enabled in the sender's user preferences

Template name	Email-triggering event	Email recipients
Package downloaded CC	A user downloads a package.	Users included in the **When a recipient downloads the package** field in the Send package form Users included in the **When a recipient receives the package** field in the Send package form Users included in the **Notify these people when I download a package** setting in the sender's user preferences
Package sent CC	A user sends a package.	Users included in the **When a recipient receives the package** field in the Send package form
Upload result	A package upload to an inbox completes successfully.	The package sender.
Upload result CC	A package is uploaded successfully.	Users included in the **When the package is available** field in the Send package form Users included in the **When a recipient receives the package** field in the Send package form Users included in the **Notify these people when I upload a package** setting in the sender's user preferences
Relay started CC	A relay starts.	Users included in the **Notify on start** setting of a workgroup or shared inbox
Relay finished CC	A relay finishes.	Users included in the **Notify on complete** setting of a workgroup or shared inbox
Relay error CC	A relay fails.	Users included in the **Notify on error** setting of a workgroup or shared inbox
Workgroup package	A package is sent to the workgroup.	Members of the workgroup

Shared inbox

Template name	Email-triggering event	Email recipients
Shared inbox invitation	A external user is invited to submit a package to a shared inbox.	The external user
Shared inbox submit	A external user sends a package to a shared inbox.	The external user

Personal submission

Template name	Email-triggering event	Email recipients
Personal invitation	A Faspex user invites a external user to submit a package.	The external user
Personal submit	A external user sends a package to the Faspex user.	The external user

Account

Template name	Email-triggering event	Email recipients
Account approved	An admin approves a self-registration application.	The self-registration applicant
Account denied	An admin denies a self-registration application.	The self-registration applicants
Welcome email	An administrator creates a new user, unless the **Send welcome email to all new users** (**Server > Configuration > Security)** option is disabled.	The new user
Forgot password	A user clicks the **Forgot my password** link on the local login page or when an admin manually resets a user account's password.	The user

Reference: Email notification variables

Welcome email

String	Description
`USER_NAME`	The full name of the email recipient
`USER_FIRST_NAME`	The first name of the email recipient
`USER_LAST_NAME`	The last name of the email recipient
`SERVER_ADDRESS`	The Faspex hostname or IP address.
`LOGIN`	The login name of the email recipient
`PRIMARY_ADDRESS`	The primary hostname or IP address
`ALTERNATE_ADDRESS_`*`number`*	The configured alternate hostname or IP address identified by *number*. See “Configuring alternate addresses” on page 94.

Forgot password

String	Description
`USER_NAME`	The full name of the email recipient
`USER_FIRST_NAME`	The first name of the email recipient
`USER_LAST_NAME`	The last name of the email recipient
`SERVER_ADDRESS`	The Faspex hostname or IP address.
`LOGIN`	The login name of the email recipient
`PRIMARY_ADDRESS`	The primary hostname or IP address
`USER_NAME`	The full name of the email recipient
`USER_FIRST_NAME`	The first name of the email recipient
`USER_LAST_NAME`	The last name of the email recipient
`SERVER_ADDRESS`	The Faspex hostname or IP address.
`LOGIN`	The login name of the email recipient

String	Description
PRIMARY_ADDRESS	The primary hostname or IP address
ALTERNATE_ADDRESS_*number*	The configured alternate hostname or IP address identified by *number*. See "Configuring alternate addresses" on page 94.

Package received

String	Description
SENDER_NAME	Full name of the sender of the package
SENDER_FIRST_NAME	First name of the sender of the package
SENDER_LAST_NAME	Last name of the sender of the package
SENDER_EMAIL	Email address of sender
SENDER_LOGIN	Login name of sender
USER_NAME	The full name of the email recipient
USER_FIRST_NAME	The first name of the email recipient
USER_LAST_NAME	The last name of the email recipient
PACKAGE_NAME	Name of the package sent to the e-mail recipient
PACKAGE_UUID	The UUID of the package
PACKAGE_URL	Package's download URL
PACKAGE_DATE	Package's sent date
PACKAGE_SIZE	Size of the data in the package
PACKAGE_FILES	Number of files in the package
PACKAGE_FILE_LIST_FIRST_10	The first 10 files or folders at the top level of the package
PACKAGE_NOTE	Message associated with the package
ALL_PUBLIC_RECIPIENTS	All recipients of the package
ALL_PUBLIC_RECIPIENTS_EMAIL	Email addresses of all recipients of the package
ALL_CC_RECIPIENTS	All contacts that were notified about the receipt of this package
ALL_CC_RECIPIENTS_EMAIL	Email addresses of all contacts that were notified about the receipt of this package
LINK_EXPIRATION_INFO	If the download link expires, a sentence describing when the link expires
PRIMARY_ADDRESS	The primary hostname or IP address
PRIMARY_ADDRESS	The primary hostname or IP address
ALTERNATE_ADDRESS_*number*	The configured alternate hostname or IP address identified by *number*. See "Configuring alternate addresses" on page 94.

Package Received CC

String	Description
CC_NAME	Full name of the user who received the CC
CC_EMAIL	Email of the user who received the CC
SENDER_NAME	Full name of the sender of the package
SENDER_FIRST_NAME	First name of the sender of the package
SENDER_LAST_NAME	Last name of the sender of the package
SENDER_EMAIL	Email address of sender
SENDER_LOGIN	Login name of sender
PACKAGE_NAME	Name of the package sent to the e-mail recipient
PACKAGE_UUID	The UUID of the package
PACKAGE_URL	Package's download URL
PACKAGE_DATE	Package's sent date
PACKAGE_SIZE	Size of the data in the package
PACKAGE_FILES	Number of files in the package
PACKAGE_FILE_LIST_FIRST_10	The first 10 files or folders at the top level of the package
PACKAGE_NOTE	Message associated with the package
ALL_PUBLIC_RECIPIENTS	All recipients of the package
ALL_PUBLIC_RECIPIENTS_EMAIL	Email addresses of all recipients of the package
ALL_CC_RECIPIENTS	All contacts that were notified about the receipt of this package
ALL_CC_RECIPIENTS_EMAIL	Email addresses of all contacts that were notified about the receipt of this package
PRIMARY_ADDRESS	The primary hostname or IP address
PRIMARY_ADDRESS	The primary hostname or IP address
ALTERNATE_ADDRESS_*number*	The configured alternate hostname or IP address identified by *number*. See "Configuring alternate addresses" on page 94.

Package Sent CC

String	Description
CC_NAME	Full name of the user who received the CC
CC_EMAIL	Email of the user who received the CC
SENDER_NAME	Full name of the sender of the package
SENDER_FIRST_NAME	First name of the sender of the package
SENDER_LAST_NAME	Last name of the sender of the package
SENDER_EMAIL	Email address of sender
SENDER_LOGIN	Login name of sender

String	Description
PACKAGE_NAME	Name of the package sent to the e-mail recipient
PACKAGE_UUID	The UUID of the package
PACKAGE_URL	Package's download URL
PACKAGE_DATE	Package's sent date
PACKAGE_SIZE	Size of the data in the package
PACKAGE_FILES	Number of files in the package
PACKAGE_FILE_LIST_FIRST_10	The first 10 files or folders at the top level of the package
PACKAGE_NOTE	Message associated with the package
ALL_PUBLIC_RECIPIENTS	All recipients of the package
ALL_PUBLIC_RECIPIENTS_EMAIL	Email addresses of all recipients of the package
ALL_CC_RECIPIENTS	All contacts that were notified about the receipt of this package
ALL_CC_RECIPIENTS_EMAIL	Email addresses of all contacts that were notified about the receipt of this package
PRIMARY_ADDRESS	The primary hostname or IP address
PRIMARY_ADDRESS	The primary hostname or IP address
ALTERNATE_ADDRESS_*number*	The configured alternate hostname or IP address identified by *number*. See "Configuring alternate addresses" on page 94.

Package Downloaded

String	Description
DOWNLOADER_NAME	Full name of the user who downloaded the package
DOWNLOADER_FIRST_NAME	First name of the user who downloaded the package
DOWNLOADER_LAST_NAME	Last name of the user who downloaded the package
DOWNLOADER_EMAIL	Email of the user who downloaded the package
DOWNLOADER_LOGIN	Login name of user who downloaded the package
SENDER_NAME	Full name of the sender of the package
SENDER_FIRST_NAME	First name of the sender of the package
SENDER_LAST_NAME	Last name of the sender of the package
SENDER_EMAIL	Email address of sender
SENDER_LOGIN	Login name of sender
USER_NAME	The full name of the email recipient
USER_FIRST_NAME	The first name of the email recipient
USER_LAST_NAME	The last name of the email recipient
PACKAGE_NAME	Name of the package sent to the e-mail recipient
PACKAGE_UUID	The UUID of the package

String	Description
PACKAGE_URL	Package's download URL
PACKAGE_DATE	Package's sent date
PACKAGE_SIZE	Size of the data in the package
PACKAGE_FILES	Number of files in the package
PACKAGE_FILE_LIST_FIRST_10	The first 10 files or folders at the top level of the package
PACKAGE_NOTE	Message associated with the package
ALL_PUBLIC_RECIPIENTS	All recipients of the package
ALL_PUBLIC_RECIPIENTS_EMAIL	Email addresses of all recipients of the package
ALL_CC_RECIPIENTS	All contacts that were notified about the receipt of this package
ALL_CC_RECIPIENTS_EMAIL	Email addresses of all contacts that were notified about the receipt of this package
PRIMARY_ADDRESS	The primary hostname or IP address
PRIMARY_ADDRESS	The primary hostname or IP address
ALTERNATE_ADDRESS_*number*	The configured alternate hostname or IP address identified by *number*. See "Configuring alternate addresses" on page 94.

Package Downloaded CC

String	Description
CC_NAME	Full name of the user who received the CC
CC_EMAIL	Email of the user who received the CC
DOWNLOADER_NAME	Full name of the user who downloaded the package
DOWNLOADER_FIRST_NAME	First name of the user who downloaded the package
DOWNLOADER_LAST_NAME	Last name of the user who downloaded the package
DOWNLOADER_EMAIL	Email of the user who downloaded the package
DOWNLOADER_LOGIN	Login name of user who downloaded the package
SENDER_NAME	Full name of the sender of the package
SENDER_FIRST_NAME	First name of the sender of the package
SENDER_LAST_NAME	Last name of the sender of the package
SENDER_EMAIL	Email address of sender
SENDER_LOGIN	Login name of sender
PACKAGE_NAME	Name of the package sent to the e-mail recipient
PACKAGE_UUID	The UUID of the package
PACKAGE_URL	Package's download URL
PACKAGE_DATE	Package's sent date
PACKAGE_SIZE	Size of the data in the package

String	Description
PACKAGE_FILES	Number of files in the package
PACKAGE_FILE_LIST_FIRST_10	The first 10 files or folders at the top level of the package
PACKAGE_NOTE	Message associated with the package
ALL_PUBLIC_RECIPIENTS	All recipients of the package
ALL_PUBLIC_RECIPIENTS_EMAIL	Email addresses of all recipients of the package
ALL_CC_RECIPIENTS	All contacts that were notified about the receipt of this package
ALL_CC_RECIPIENTS_EMAIL	Email addresses of all contacts that were notified about the receipt of this package
PRIMARY_ADDRESS	The primary hostname or IP address
PRIMARY_ADDRESS	The primary hostname or IP address
ALTERNATE_ADDRESS_*number*	The configured alternate hostname or IP address identified by *number*. See "Configuring alternate addresses" on page 94.

Workgroup Package

String	Description
USER_NAME	The full name of the email recipient
USER_FIRST_NAME	The first name of the email recipient
USER_LAST_NAME	The last name of the email recipient
WORKGROUP_NAME	Name of the workgroup the package was sent to
SENDER_NAME	Full name of the sender of the package
SENDER_FIRST_NAME	First name of the sender of the package
SENDER_LAST_NAME	Last name of the sender of the package
SENDER_EMAIL	Email address of sender
SENDER_LOGIN	Login name of sender
PACKAGE_NAME	Name of the package sent to the e-mail recipient
PACKAGE_UUID	The UUID of the package
PACKAGE_URL	Package's download URL
PACKAGE_DATE	Package's sent date
PACKAGE_SIZE	Size of the data in the package
PACKAGE_FILES	Number of files in the package
PACKAGE_FILE_LIST_FIRST_10	The first 10 files or folders at the top level of the package
PACKAGE_NOTE	Message associated with the package
ALL_PUBLIC_RECIPIENTS	All recipients of the package
ALL_PUBLIC_RECIPIENTS_EMAIL	Email addresses of all recipients of the package
ALL_CC_RECIPIENTS	All contacts that were notified about the receipt of this package

String	Description
ALL_CC_RECIPIENTS_EMAIL	Email addresses of all contacts that were notified about the receipt of this package
PRIMARY_ADDRESS	The primary hostname or IP address
PRIMARY_ADDRESS	The primary hostname or IP address
ALTERNATE_ADDRESS_*number*	The configured alternate hostname or IP address identified by *number*. See "Configuring alternate addresses" on page 94.

Upload Result

String	Description
SENDER_NAME	Full name of the sender of the package
SENDER_FIRST_NAME	First name of the sender of the package
SENDER_LAST_NAME	Last name of the sender of the package
SENDER_EMAIL	Email address of sender
SENDER_LOGIN	Login name of sender
PACKAGE_NAME	Name of the package sent to the e-mail recipient
PACKAGE_UUID	The UUID of the package
PACKAGE_URL	Package's download URL
PACKAGE_DATE	Package's sent date
PACKAGE_SIZE	Size of the data in the package
PACKAGE_FILES	Number of files in the package
PACKAGE_FILE_LIST_FIRST_10	The first 10 files or folders at the top level of the package
PACKAGE_NOTE	Message associated with the package
UPLOAD_RESULT	The result of the package upload
STATUS_URL	URL to check package upload status (does not work in subject)
STATUS_LINK	Link to check package upload status (does not work in subject)
ALL_PUBLIC_RECIPIENTS	All recipients of the package
ALL_PUBLIC_RECIPIENTS_EMAIL	Email addresses of all recipients of the package
ALL_CC_RECIPIENTS	All contacts that were notified about the receipt of this package
ALL_CC_RECIPIENTS_EMAIL	Email addresses of all contacts that were notified about the receipt of this package
PRIMARY_ADDRESS	The primary hostname or IP address
PRIMARY_ADDRESS	The primary hostname or IP address
ALTERNATE_ADDRESS_*number*	The configured alternate hostname or IP address identified by *number*. See "Configuring alternate addresses" on page 94.

Upload Result CC

String	Description
CC_NAME	Full name of the user who received the CC
CC_EMAIL	Email of the user who received the CC
SENDER_NAME	Full name of the sender of the package
SENDER_FIRST_NAME	First name of the sender of the package
SENDER_LAST_NAME	Last name of the sender of the package
SENDER_EMAIL	Email address of sender
SENDER_LOGIN	Login name of sender
PACKAGE_NAME	Name of the package sent to the e-mail recipient
PACKAGE_UUID	The UUID of the package
PACKAGE_URL	Package's download URL
PACKAGE_DATE	Package's sent date
PACKAGE_SIZE	Size of the data in the package
PACKAGE_FILES	Number of files in the package
PACKAGE_FILE_LIST_FIRST_10	The first 10 files or folders at the top level of the package
PACKAGE_NOTE	Message associated with the package
UPLOAD_RESULT	The result of the package upload
ALL_PUBLIC_RECIPIENTS	All recipients of the package
ALL_PUBLIC_RECIPIENTS_EMAIL	Email addresses of all recipients of the package
ALL_CC_RECIPIENTS	All contacts that were notified about the receipt of this package
ALL_CC_RECIPIENTS_EMAIL	Email addresses of all contacts that were notified about the receipt of this package
PRIMARY_ADDRESS	The primary hostname or IP address
PRIMARY_ADDRESS	The primary hostname or IP address
ALTERNATE_ADDRESS_*number*	The configured alternate hostname or IP address identified by *number*. See "Configuring alternate addresses" on page 94.

Relay Started CC

String	Description
CC_NAME	Full name of the user who received the CC
CC_EMAIL	Email of the user who received the CC
WORKGROUP_NAME	Name of the workgroup the package was sent to
DESTINATION_NODE	Storage node
DESTINATION_DIRECTORY	Docroot relative path to the destination directory on the storage node

String	Description
SENDER_NAME	Full name of the sender of the package
SENDER_FIRST_NAME	First name of the sender of the package
SENDER_LAST_NAME	Last name of the sender of the package
SENDER_EMAIL	Email address of sender
SENDER_LOGIN	Login name of sender
PACKAGE_NAME	Name of the package sent to the e-mail recipient
PACKAGE_UUID	The UUID of the package
PACKAGE_URL	Package's download URL
PACKAGE_DATE	Package's sent date
PACKAGE_SIZE	Size of the data in the package
PACKAGE_FILES	Number of files in the package
PACKAGE_FILE_LIST_FIRST_10	The first 10 files or folders at the top level of the package
PACKAGE_NOTE	Message associated with the package
ALL_PUBLIC_RECIPIENTS	All recipients of the package
ALL_PUBLIC_RECIPIENTS_EMAIL	Email addresses of all recipients of the package
ALL_CC_RECIPIENTS	All contacts that were notified about the receipt of this package
ALL_CC_RECIPIENTS_EMAIL	Email addresses of all contacts that were notified about the receipt of this package
PRIMARY_ADDRESS	The primary hostname or IP address
PRIMARY_ADDRESS	The primary hostname or IP address
ALTERNATE_ADDRESS_*number*	The configured alternate hostname or IP address identified by *number*. See "Configuring alternate addresses" on page 94.

Relay Finished CC

String	Description
CC_NAME	Full name of the user who received the CC
CC_EMAIL	Email of the user who received the CC
WORKGROUP_NAME	Name of the workgroup the package was sent to
DESTINATION_NODE	Storage node
DESTINATION_DIRECTORY	Docroot relative path to the destination directory on the storage node
SENDER_NAME	Full name of the sender of the package
SENDER_FIRST_NAME	First name of the sender of the package
SENDER_LAST_NAME	Last name of the sender of the package
SENDER_EMAIL	Email address of sender
SENDER_LOGIN	Login name of sender

String	Description
PACKAGE_NAME	Name of the package sent to the e-mail recipient
PACKAGE_UUID	The UUID of the package
PACKAGE_URL	Package's download URL
PACKAGE_DATE	Package's sent date
PACKAGE_SIZE	Size of the data in the package
PACKAGE_FILES	Number of files in the package
PACKAGE_FILE_LIST_FIRST_10	The first 10 files or folders at the top level of the package
PACKAGE_NOTE	Message associated with the package
ALL_PUBLIC_RECIPIENTS	All recipients of the package
ALL_PUBLIC_RECIPIENTS_EMAIL	Email addresses of all recipients of the package
ALL_CC_RECIPIENTS	All contacts that were notified about the receipt of this package
ALL_CC_RECIPIENTS_EMAIL	Email addresses of all contacts that were notified about the receipt of this package
PRIMARY_ADDRESS	The primary hostname or IP address
PRIMARY_ADDRESS	The primary hostname or IP address
ALTERNATE_ADDRESS_*number*	The configured alternate hostname or IP address identified by *number*. See "Configuring alternate addresses" on page 94.

Relay Error CC

String	Description
CC_NAME	Full name of the user who received the CC
CC_EMAIL	Email of the user who received the CC
WORKGROUP_NAME	Name of the workgroup the package was sent to
DESTINATION_NODE	Storage node
DESTINATION_DIRECTORY	Docroot relative path to the destination directory on the storage node
SENDER_NAME	Full name of the sender of the package
SENDER_FIRST_NAME	First name of the sender of the package
SENDER_LAST_NAME	Last name of the sender of the package
SENDER_EMAIL	Email address of sender
SENDER_LOGIN	Login name of sender
PACKAGE_NAME	Name of the package sent to the e-mail recipient
PACKAGE_UUID	The UUID of the package
PACKAGE_URL	Package's download URL
PACKAGE_DATE	Package's sent date
PACKAGE_SIZE	Size of the data in the package

String	Description
PACKAGE_FILES	Number of files in the package
PACKAGE_FILE_LIST_FIRST_10	The first 10 files or folders at the top level of the package
PACKAGE_NOTE	Message associated with the package
ALL_PUBLIC_RECIPIENTS	All recipients of the package
ALL_PUBLIC_RECIPIENTS_EMAIL	Email addresses of all recipients of the package
ALL_CC_RECIPIENTS	All contacts that were notified about the receipt of this package
ALL_CC_RECIPIENTS_EMAIL	Email addresses of all contacts that were notified about the receipt of this package
PRIMARY_ADDRESS	The primary hostname or IP address
PRIMARY_ADDRESS	The primary hostname or IP address
ALTERNATE_ADDRESS_*number*	The configured alternate hostname or IP address identified by *number*. See "Configuring alternate addresses" on page 94.

shared inbox Invitation

String	Description
EMAIL	Email address of the invited outside email user
shared inbox_NAME	shared inbox to which the outside email user was invited
shared inbox_URL	The URL that the outside email user can use to send packages to the shared inbox
shared inbox_LINK	HTML link that the outside email user can use to send packages to the shared inbox
LINK_EXPIRATION_INFO	If the download link expires, a sentence describing when the link expires
PRIMARY_ADDRESS	The primary hostname or IP address
PRIMARY_ADDRESS	The primary hostname or IP address
ALTERNATE_ADDRESS_*number*	The configured alternate hostname or IP address identified by *number*. See "Configuring alternate addresses" on page 94.

shared inbox Submit

String	Description
shared inbox_NAME	shared inbox to which the outside email user was invited
SENDER_EMAIL	Email address of sender
PACKAGE_NAME	Name of the package sent to the e-mail recipient
PACKAGE_UUID	The UUID of the package
PACKAGE_URL	Package's download URL
PACKAGE_DATE	Package's sent date
PACKAGE_SIZE	Size of the data in the package

String	Description
PACKAGE_FILES	Number of files in the package
PACKAGE_FILE_LIST_FIRST_10	The first 10 files or folders at the top level of the package
PACKAGE_NOTE	Message associated with the package
STATUS_URL	URL to check package upload status (does not work in subject)
STATUS_LINK	Link to check package upload status (does not work in subject)
PRIMARY_ADDRESS	The primary hostname or IP address
PRIMARY_ADDRESS	The primary hostname or IP address
ALTERNATE_ADDRESS_*number*	The configured alternate hostname or IP address identified by *number*. See "Configuring alternate addresses" on page 94.

Personal Invitation

String	Description
EMAIL	Email address of the invited outside email user
RECIPIENT_NAME	Full name of the recipient who invited the outside email
RECIPIENT_FIRST_NAME	First name of the recipient who invited the outside email
RECIPIENT_LAST_NAME	Last name of the recipient who invited the outside email
SUBMISSION_URL	The URL that the outside email user can use to send a package
SUBMISSION_LINK	HTML link that the outside email user can use to send a package
LINK_EXPIRATION_INFO	If the download link expires, a sentence describing when the link expires
PRIMARY_ADDRESS	The primary hostname or IP address
PRIMARY_ADDRESS	The primary hostname or IP address
ALTERNATE_ADDRESS_*number*	The configured alternate hostname or IP address identified by *number*. See "Configuring alternate addresses" on page 94.

Personal Submit

String	Description
RECIPIENT_NAME	Full name of the recipient who invited the outside email
RECIPIENT_FIRST_NAME	First name of the recipient who invited the outside email
RECIPIENT_LAST_NAME	Last name of the recipient who invited the outside email
SENDER_EMAIL	Email address of sender
PACKAGE_NAME	Name of the package sent to the e-mail recipient
PACKAGE_UUID	The UUID of the package
PACKAGE_URL	Package's download URL
PACKAGE_DATE	Package's sent date
PACKAGE_SIZE	Size of the data in the package

String	Description
PACKAGE_FILES	Number of files in the package
PACKAGE_FILE_LIST_FIRST_10	The first 10 files or folders at the top level of the package
PACKAGE_NOTE	Message associated with the package
STATUS_URL	URL to check package upload status (does not work in subject)
STATUS_LINK	Link to check package upload status (does not work in subject)
PRIMARY_ADDRESS	The primary hostname or IP address
PRIMARY_ADDRESS	The primary hostname or IP address
ALTERNATE_ADDRESS_*number*	The configured alternate hostname or IP address identified by *number*. See "Configuring alternate addresses" on page 94.

Account Approved

String	Description
USER_NAME	The full name of the email recipient
USER_FIRST_NAME	The first name of the email recipient
USER_LAST_NAME	The last name of the email recipient
SERVER_ADDRESS	The Faspex hostname or IP address.
LOGIN	The login name of the email recipient
PRIMARY_ADDRESS	The primary hostname or IP address

Account Denied

String	Description
USER_NAME	The full name of the email recipient
USER_FIRST_NAME	The first name of the email recipient
USER_LAST_NAME	The last name of the email recipient
SERVER_ADDRESS	The Faspex hostname or IP address.
LOGIN	The login name of the email recipient
PRIMARY_ADDRESS	The primary hostname or IP address

Reference: User account settings

Account details and security

Setting	Description
Account expires	Select to set an expiration date for the user. The user becomes inactive on the specified date. **Note:** Admin accounts do not expire.
Account is active	Select to activate this account so that the user can log into Faspex. Clear to disable the account.

Setting	Description
	Note: Admin accounts are always active.

Custom password policy

Setting	Description
Override default password policies	Enable to set password expiration and password reuse limits.
Expire passwords and **Days until expiration**	Expire passwords a specified number of days after they are set.
Prohibit password reuse and **Number of previous passwords to prohibit**	Prevent users from reusing a specified number of previously used passwords.
Require strong passwords	When enabled, existing passwords remain valid. By default, new passwords must be at least six characters long, with at least one letter, one number, and one symbol. You can override these default strong password requirements by configuring the regex expression password criteria.
Regex expression password criteria	A regular expression that can be used to customize strong password requirements. Changing this setting does not affect existing passwords, but any new password must match the regular expression you configure. Example: (?=.*[A-Z])(?=.*(\d|\W|_)).{7,} If you don't configure this setting, the default strong password settings apply.
Password requirements	Describe the password requirements for users. This appears to users as helper text under the New password field.

Package sending

Setting	Description
User can edit receipt addresses when sending a package	Allow user to modify addresses to receive notification emails regarding a package sent by this user.
Send a copy of receipt emails to these addresses	Faspex sends a copy of every package receipt notification sent to this account to the Faspex users and email addresses listed in this field. Recipients listed in this field receive notifications for every package sent and received by this account. The CC Receipt field on the New Package page is auto-populated with the addresses listed in this field. If the sender **Can edit receipt addresses when sending a package**, whatever the sender enters in the CC Receipt field on the New Package page

Setting	Description
	overrides this setting. If the sender removes any of the original email addresses from the field, Faspex does not send a notification to that user. If the sender does not have permission to **Allow editing of receipt addresses on package creation**, then this field is honored. **Note:** If you are adding multiple email addresses, separate them with commas (,), semicolons (;), or white-spaces.

Package deletion

Setting	Description
Downloads-based policy	Override the server default and set a downloads-based expiration policy specific to this user. For more information on each option, see "Download-based policy" on page 95.
User can set package-deletion policy when sending a package	Override the server default and choose whether this user can pick a package expiration policy when sending a new package.

Advanced transfer settings

Field	Description
Initial upload rate (Kbps)	Specify the target transfer rate for user-to-server transfers.
Initial download rate (Kbps)	Specify the target transfer rate for server-to-user transfers.
Lock minimum rate and policy	Prevent clients from adjusting the transfer policy or minimum transfer rate
Maximum upload target (Kbps)	Specify the maximum transfer rate for user-to-server transfers.
Maximum download target (Kbps)	Specify the maximum transfer rate for server-to-user transfers.

Reference: User permissions

User permissions

Setting	Description
Upload packages	The user can upload packages to a node. Disabling this does not prevent the user from sending a package from a shared folder, if **Create packages from a remote source** is enabled.
Download packages	The user can download received packages. When disabled, the user can still receive packages, but cannot download packages or the packages' files.

Setting	Description
Forward packages	The user can forward received packages to other users.
Send packages from a remote source	The user can include content from a shared inbox when sending a package. You must first add remote sources to Faspex to see the **Source** drop-down menu. This setting must be set on a per-user basis. There is no global option to enable this setting for all users.
Send normal packages	If disabled, the user can only send packages to shared inboxes. You can change the server default by doing to **Security > Users**.
Invite external senders	You must enable this option globally to override this feature. For more information, see "Allow Faspex users to send packages to external users" on page 109. Override the server default and select **Allow** to enable this user to invite anyone to send a package to this user.
Share submission links	You must enable this option globally to see this feature. For more information, see "Public submission links" on page 103. Override the server default and select **Allow** to enable users to share their public submission link. Anyone with a public submission link can submit packages to this user through this submission link. **Note:** Even if the public submission link feature is enabled for registered Faspex users, they can override the feature for their own account by going to their user preferences. See "Share your public submission link" on page 49.
Send packages to external emails	Override the server default and select **Allow** to enable the user to send packages to external email addresses. For more information, see "Allow Faspex users to send packages to external users" on page 109 Select **Allow** to enable this user to send packages to external.
Send to all Faspex users	Override the server default and select **Allow** to allow users to send packages to all Faspex users. If this feature is enabled, all existing Faspex users appear in the contact list. By default, users can send packages to: • Users in their contacts • Personal distribution lists • Shared inboxes • Workgroups they are a part of (if workgroup settings allow) • Members of workgroups they are a part of (if workgroup settings allow) • Global distribution lists (if allowed to see)
See global distribution lists	Override the server default and select **Allow** to enable the user to see and send to global distribution lists.
Keep user directory private	Override the server default and select **Yes** to prevent users from being able to see the entire user directory, even if they have permissions to send to all Faspex users.

IP permissions

Important: Restricting access by IP address requires the requesting client's IP address be preserved through the TCP/IP network until the request reaches Faspex. If you have a network configuration that passes the requesting client's IP address through the network with the `X-Forwarded-For` HTTP Header, you will not be able to restrict access to Faspex via IP addresses.

Tip: For all of these settings, you can use a wildcard (*) to allow a range of options. For example, specifying `192.0.2.*` allows a user to login from `192.0.2.1`, `192.0.2.2`, `192.0.2.3`, and so on. Separate multiple IP addresses with commas (,).

Setting	Description
User can log in from only these IP addresses	Specify the IP addresses that a Faspex user can login from.
User can download only when downloading from these IP addresses	Specify the IP addresses that the user must access Faspex from to download packages.
User can send packages only when sending from these IP addresses	Specify the IP addresses that the user must access Faspex to upload packages.

Reference: User accounts provisioned by Just-In-Time (JIT) provisioning

When a SAML user logs in to Faspex for the first time, Faspex automatically creates a new user account based on the information provided by the SAML response. If the SAML response also contains group information, and that group does not yet exist in Faspex, Faspex automatically creates a new SAML group for each group of which the user is a member. For more information about SAML groups, see "Creating SAML groups" on page 117.

Note: If an admin enables the **Restrict access to known groups** feature for the SAML configuration, only members of existing Faspex SAML groups can log in. This also means that new SAML groups are not automatically created when SAML users log in. For more information about SAML configuration options, see "Configuring authentication rules for a SAML configuration" on page 118.

SAML users and external users

When a SAML user logs in to Faspex for the first time, Faspex checks for existing external users matching the email address of the SAML user. If such a user exists, Faspex merges the two accounts.

Group permissions

A SAML user belonging to multiple groups is given the permissions and settings of all groups it belongs to with permissions overriding restrictions. For example, if Group A disallows sending to external users but Group B does not, users who belong to both groups are allowed to send to external users. Settings that require specific handling are as follows:

- Account expiration is only enabled if all groups to which a user belongs specify account expiration. If account expiration is enabled, the expiration date is set to the latest expiration date from among all groups.
- For any settings that use **Default**, **Yes** or **Allow**, and **No** or **Deny**, the setting is set to **Yes** if any group specifies **Yes**, and it is set to **No** if all groups are set to **No**. Otherwise, it is set to use the server default.
- For package deletion policy, override is enabled if all groups specify override, or if the least restrictive group setting is less restrictive than the server-wide setting. If override is enabled, the least restrictive group setting is used. **Do nothing** is less restrictive than **Delete files after all recipients download all files**, which in turn is less restrictive than **Delete files after any recipient downloads all files**.

- For advanced transfer settings, override is enabled if all groups specify override or if any group specifies any transfer rate that is higher than the server default. If override is enabled, each transfer rate is set to the higher of the highest value from among the groups and the server default. The minimum rate policy is locked only if all groups specify the setting.

Appendix

Faspex GDPR

GDPR stands for General Data Protection Regulation. The GDPR has been adopted by the European Union and European Economic Area (“EU”) starting May 25, 2018.

Why is GDPR important?

GDPR establishes a stronger data protection regulatory framework for processing of personal data of individuals, impacts IBM and IBM's client contracts, policies and procedures when handling personal data. GDPR brings:

- New and enhanced rights for individuals
- Widened definition of personal data
- New obligations for processors
- Significant financial penalties for non-compliance
- Compulsory data breach notification

Data collection and data life cycle

Understand what data is collected and the end-to-end process personal data goes through when using Faspex.

Data collection

Faspex handles these personal data types:

- Account details
  - Username
  - Password
  - First Name
  - Email
  - Roles & privileges
  - User profile fields (see *Configuring Custom User Fields*)
  - User IP addresses
- Server configuration
- Logs
- User-accessible content
  - Uploaded files and packages
  - Downloaded files and packages
  - Package metadata

Data life cycle

Account data

Event in which Faspex stores account data	What data is stored in the database	When is the data removed
An admin or manager creates a user account.	Account details are stored in the database. Only the password is encrypted on the database.	An admin or manager removes the user.
An admin imports a SAML user and Faspex automatically creates a new account for the user.	Account usernames and email addresses are stored in the database, but no passwords.	An admin removes the new Faspex account.
A user logs in through SAML and Faspex automatically creates a new account for the user.	Account usernames and email addresses are stored in the database, but no passwords.	An admin removes the new Faspex account.
An admin creates a workgroup and adds a user account to the workgroup.	Account permissions are stored in the database.	An admin removes the user from the workgroup or removes the user account.
An admin creates a shared inbox and adds a user account to the shared inbox.	Account permissions are stored in the database.	An admin removes the user from the workgroup or removes the user account.
An admin elevates a user to the admin role.	Account authorizations are stored in the database.	An admin demotes the user or removes the user account.
An admin elevates a user to the manager role.	Account authorizations are stored in the database.	An admin demotes the user or removes the user account.
An admin elevates a user to the workgroup admin role.	Account authorizations are stored in the database.	An admin demotes the user, removes the user from the workgroup, or removes the user account.
An admin elevates a user to the shared inbox admin role.	Account authorizations are stored in the database.	An admin demotes the user, removes the user from the shared inbox, or removes the user account.
An admin creates a global distribution list.	Email addresses in the distribution list are stored in the database.	An admin deletes the distribution list.
A user creates a distribution list.	Email addresses in the distribution list are stored in the database.	An admin deletes the distribution list.
A user registers an account.	Account details are stored in the database.	An admin removes the user account.
A user requests an account.	Account details are stored in the database.	An admin rejects the request.
A user modifies account settings.	Account details are stored in the database.	An admin removes the user account.
A user sends content to an external email address.	The recipient email address is stored in the database.	A user or admin removes the contact associated with the email address.
A user logs in	The user's IP address is stored in the logs.	A system admin removes the logs.

Server configuration

Event in which Faspex stores account data	What data is stored	When is the data removed
An admin adds a transfer node to Faspex	Node details are stored in the database.	An admin removes the node
An admin adds a file storage location from a source node	File storage details and permissions are stored in the database.	An admin removes the file storage or the source node
An admin changes server configuration settings	Server configuration settings are stored in the database.	A system admin uninstalls Faspex

Logs

Event in which Faspex stores account data	What data is stored	When is the data removed
All events	Event details are stored in logs written as standard output in the Docker container logs (see "Viewing container logs" on page 145).	A system admin removes the log files manually or automatically via log retention policies.

User-accessible content

Event in which Faspex stores account data	What data is stored	When is the data removed
A user uploads content to a file storage location.	Content is stored on the destination node.	A user removes the content through the Faspex UI or the system admin removes the content from the directory on the destination node.
Faspex relays a package	Content is stored on the relay destination node.	A user removes the content through the Faspex UI or the system admin removes the content from the directory on the destination node.
An end user downloads content from Faspex.	Content is stored on the end user's computer.	End user's discretion.
A user sends a package	Custom metadata defined by user profile fields is stored in the Faspex database.	An admin deletes the package.

Data storage

Control storage of personal data.

Database

Faspex account data and server configuration data are stored on the database. Faspex encrypts only stored passwords. The customer is expected to control and protect the database.

Nodes

User-accessible content is stored on destination nodes. The customer is expected to control and protect the nodes and to take necessary security measures, such as enabling file encryption on Faspex and on the nodes.

Backups

Admins can backup the database and configuration files using the Faspex Utility web application included in Faspex. The customer is expected to encrypt, control, and protect generated backups.

Logs

Faspex saves user activities in unencrypted logs. User activities include: user login, change of management, and access to client data. By default, logs are overwritten in a rotated manner. The customer is expected to encrypt, control, and protect logs.

Note: Log rotation is only supported on Docker-CE and Docker installations. Podman (as of version 4.2) has not implemented the `max-file` setting for containers, it truncates logs instead of rotating them. This means that once a log reaches the `max-size`, Podman deletes all logs and starts new ones.

Data access

Control access to personal data.

Roles and access rights

See "User roles" on page 3.

Separation of duties

Admins can appoint Faspex managers to create and manage users. Admins retain full control of the application and application data.

Authentication

Faspex supports two methods of authentication:

- Logging in with a username and password.
- "Logging in with SAML" on page 38.

Data processing

Control data processing.

Data protection in motion

Admins can configure Faspex to initiate transfers using AES-128 encryption mode.

Data protection at rest

Admins can configure encryption-at-rest (EAR) to encrypt uploaded content. When a user downloads content from the server, server-side EAR first decrypts the files and then transfers the files to the client in an unencrypted state. See "File encryption" on page 101.

The encryption key for EAR is saved in the aspera.conf configuration file on the node as cleartext. The customer is expected to take proper protection to avoid unauthorized access.

Data deletion

Understand who can delete user-accessible content and account data.

Deleting user-accessible content

- Admins can archive and delete packages.
- Shared inbox admins can archive and delete packages in the shared inbox.
- Regular users cannot archive or delete packages.

Deleting account data

Only admin accounts can remove user accounts.

Data subject rights

Understanding data subject rights.

Right to access

- End users can access their account data.
- Admins can grant or revoke a user's permission to access certain customer data.
- Admins admins can give external users permission to upload content to a public link.
- Admins admins can give external users permission to upload content to a shared inbox.
- Admins can give Faspex users permission to send content to external users.
- Admins can access all personal data except for passwords.
- Faspex does not have the off-the-shelf functionality to single out a specific user's data from logs or database backup.

Right to modify

- End users can modify their own account data.
- Admins can modify any user's management data.
- Admins can grant or revoke a user's permission to modify certain customer data.
- The activity log data cannot be modified by Faspex.

Right to restrict processing

- Faspex requires all data to provide adequate service. Admins control restricting data usage for other purposes.

Right to object

- Admins are in control.

Right to be forgotten

- Admins can remove any end user from the active system.
- Managers can remove any regular or workgroup user from the active system.
- Faspex does not provide off-the-shelf functionality to remove data from logs or database backups. It is up to system admin to design a way to remove data from logs and database backups.

Right to data portability

- Faspex does not provide off-the-shelf functionality to port data from logs or database backups. It is up to system admin to design a way to port user data.

Glossary

node

A node (also known as an Aspera transfer server) is a server running a transfer server product, such as IBM Aspera High-Speed Transfer Server (HSTS), configured for use with Faspex. Faspex requires that nodes run HSTS 4.3 and later.

external user

A external user is a user that is not associated with a Faspex user account. Faspex users can send *public packages* to external users with a *public submission link* and invite external users to send them a package through an email invitation or with a *public submission link*.

By default, external users do not need to create an account to download a public package, but an admin can configure Faspex to require external users to create an account before downloading the package (see "Allowing public packages" on page 105).

public package

A public package is a package that recipients can download using a link.

Faspex users can download public packages without logging in. By default, external users do not need to create an account to download a public package, but an admin can configure Faspex to require external users to create an account before downloading the package (see "Allowing public packages" on page 105).

public submission link

Anyone with a public submission link can send a package to the associated user or shared inbox.

shared inbox

A shared inbox provides a file submission system for users to share packages.

shared folder

Shared folders are also known as storage locations. Shared folders are directories on a node. You can use files and folders in shared folders as the source of content for a package.

storage location

Storage locations are also known as shared folders. Storage locations are directories on a node. You can use files and folders in storage locations as the source of content for a package.

workgroup

A workgroup defines a group of users that can be sent packages as a collective whole. A Faspex administrator determines who has permissions to send packages to a workgroup and where those packages are stored. The administrator also determines whether members can see and send packages to other workgroup members.

Integration and API guides

Introduction

Faspex 5 offers first-class citizen APIs by using current and modern industry standards to support customers in their integrations. There are multiple improvements (for example, in the authentication and authorization mechanisms) compared to the Faspex V3 and V4 APIs.

This guide provides basic examples (with sample code) for authorizing to Faspex 5 and for sending and downloading packages with either IBM Aspera Connect or IBM Aspera HTTP Gateway.

Migrating from Faspex 4 and before

If you are upgrading to Faspex 5 from Faspex 4 and before and need to migrate your existing integrations, see "Differences between Faspex 5 API and previous versions" on page 177.

Differences between Faspex 5 API and previous versions

Faspex 4 (and before) integration migration matrix with Faspex 5

Items	Faspex 4 and older	Faspex 5
Authentication	• Basic authentication • SAML	• Faspex 5 UI login page • SAML
Authorization	• OAuth flow • HMAC-signed bearer token	• OAuth 2 flow • JWT grant flow (with optional impersonation)
API	RESTful	RESTful with OAS 3.0 file (allows any industry-standard code generator to build client libraries)
API base path	`/aspera/faspex/api/`	`/aspera/faspex/api/v5/`
Data transfer	Use the API to create Faspex packages. Use an Aspera transfer tool or SDK to pick files and perform the transfer. Compatible transfer tools and SDKs are: • **ascp** • Connect SDK • HTTP Gateway Javascript SDK • FaspManagerSDK	Use the API to create Faspex packages. Use an Aspera transfer tool or SDK to pick files and perform the transfer. Compatible transfer tools and SDKs are: • **ascp** • Connect SDK • HTTP Gateway Javascript SDK • TransferSDK
Sending packages workflows	v3 1. Call POST `/send`. 2. Start the upload transfer by mapping JSON response fields directly to transfer tool or SDK fields.	1. Create a package by calling POST `/packages`. 2. Retrieve a valid transfer specification by calling POST `/packages/{id}/transfer_spec/upload` 3. Start the upload transfer by passing the retrieved

Items	Faspex 4 and older	Faspex 5
	Note: This procedure is prone to error. v4 1. Call POST `/api/users/me/packages`. 2. Call POST `/api/users/me/packages/{package_id}/transfer_specs` with `{"direction": "send"}` in the request. 3. Start the upload transfer by mapping JSON response fields directly to transfer tool or SDK fields. **Note:** This procedure is prone to error.	a. or passing directly the transfer_spec JSON response to Connect API AW.Connect#startTransfer : (safe) b. or passing directly the transfer_spec JSON response to Transfer SDK StartTransfer : (safe)
Receiving package	API calls followed by a FASP transfer for package sending or receiving use casesAPI-only calls for all the other use cases	API calls followed by a FASP transfer for package sending or receiving use casesAPI-only calls for all the other use cases
SDK	Java SDK with support for a limited number of API endpoints	Effort is done in making easy the code generation covering the entire API endpoints. Example provided.

Authentication and authorization

Faspex 5 adopts OAuth 2 as the authorization mechanism for its APIs. The Faspex V3 API used less-secure HTTP basic authorization and the Faspex V4 API did not decouple user authentication from authorization.

An administrator can register an API client to retrieve a bearer token to interact with the endpoints. The bearer token can be obtained through either a JWT grant flow, an authorization code flow or an authorization code with PKCE (Proof Key for Code Exchange) flow. See "Authentication and authorization" on page 191.

API differences

- Packages API
- Nodes API
- Metadata profiles
- Users API
- Shared inboxes (formerly dropboxes) API
- Dropbox and workgroup memberships API
- SAML configuration API
- Faspex services API

Packages API

In the Faspex 5 API, packages are handled by top-level endpoints and do not require you to identify packages by ownership (such as user or workgroup). Faspex 5 expects you to directly interact with packages using the package ID. Package availability is based on the role of the current logged-in user. This change makes it easier to work with packages, but may require those migrating from Faspex 4 to use several endpoints to achieve the same goal.

Enhancements to search and scoping are planned for a future release of Faspex 5.

Endpoint
API calls

View all packages (admin search)

Faspex 4

```
POST /api/packages/
```

Faspex 5

```
GET /api/v5/all/packages
```

Reference

Get all packages of a user

Faspex 4

```
GET /api/users/{user_id}/packages/
```

Faspex 5

- See all packages available to you as the current user:

  ```
  GET /api/v5/packages
  ```

  Reference
- As an admin, see all packages available to another user:

  ```
  GET /api/v5/packages?q=username
  ```

  The q query parameter searches sender and recipient user fields.

  Reference

View all packages (admin search)

Faspex 4

```
POST /api/packages/
```

Faspex 5

```
GET /api/v5/all/packages
```

Reference

Get all packages of a user

Faspex 4

```
GET /api/users/{user_id}/packages/
```

Faspex 5

See all packages available to the current user:

```
GET /api/v5/packages
```

Reference

As an admin, see all packages available to another user:

```
GET /api/v5/all/packages
```

Sort the response based on recipient user IDs.

Reference

Get a package of a user

Faspex 4

```
GET /api/users/{user_id}/packages/{package_delivery_id}
```

Faspex 5

Search for the package using the package ID:

```
GET /api/v5/packages/{package_id}
```

Reference

Create a package

Faspex 4

```
POST /api/users/{user_id}/packages
```

Faspex 5

```
POST /api/v5/packages
```

Reference

Delete a package of a user

Faspex 4

```
DELETE /api/users/{user_id}/packages/{package_delivery_id}
```

Faspex 5

```
DELETE /api/v5/packages/{package_id}
```

Reference

Delete a package from a dropbox

Faspex 4

```
DELETE /api/dropboxes/{dropbox_id}/packages/{package_delivery_id}
```

Faspex 5

```
DELETE /api/v5/packages/{package_id}
```

Reference

Delete all packages of a user

Faspex 4

```
DELETE /api/users/{user_id}/packages
```

Faspex 5

1. Get a list of packages:

   - See all packages available to you as the current user:

     ```
     GET /api/v5/packages
     ```

     Reference
   - As an admin, see all packages available to another user:

     ```
     GET /api/v5/packages?q=username
     ```

     The q query parameter searches sender and recipient user fields.
2. Delete multiple packages using their package IDs:

   ```
   DELETE /api/v5/packages?ids={package_id, package_id, package_id...}
   ```

   Reference

Delete all packages from a dropbox

Faspex 4

```
DELETE /api/dropboxes/{dropbox_id}/packages
```

Faspex 5

1. Get packages for a shared inbox:

   ```
   GET /api/v5/shared_inbox/{shared_inbox_id}/packages
   ```

   Reference
2. Delete multiple packages using their package IDs:

   ```
   DELETE /api/v5/packages?ids={package_id, package_id, package_id...}
   ```

   Reference

Update and edit package attributes, recipients, and metadata of a user

Faspex 4

```
PUT /api/users/{user_id}/packages/{package_delivery_id}
```

Faspex 5

```
PUT /api/v5/package/{package_id}
```

Reference

Update and edit package attributes, recipients, and metadata from a dropbox

Faspex 4

```
PUT /api/dropboxes/{dropbox_id}/packages/{package_delivery_id}
```

Faspex 5

```
PUT /api/v5/package/{package_id}
```

Reference

Search for a package of a user

Faspex 4

```
GET /api/users/{user_id}/packages
```

Faspex 5

Search for packages available to the current user with different filters:

```
GET /api/v5/{mailbox_type}/packages
```

Reference

Search for a package in a dropbox

Faspex 4

```
GET /api/dropboxes/{dropbox_id}/packages
```

Faspex 5

Search for packages in the shared inbox with different filters:

```
GET /api/v5/shared_inbox/{shared_inbox_id}/packages
```

Reference

Sort the response based on recipient types.

Forward a package of a user

Faspex 4

```
POST /api/users/{user_id}/packages/{package_delivery_id}
```

Faspex 5

```
POST /api/v5/packages/{id}/forward
```

Reference

Forward a package in a dropbox

Faspex 4

```
POST /api/dropboxes/{dropbox_id}/packages/{package_delivery_id}
```

Faspex 5

```
POST /api/v5/packagess/{id}/forward
```

Reference

Replicate package contents of a user

Faspex 4

```
POST /api/users/{user_id}/packages/{package_delivery_id}/replicate_contents
```

Faspex 5

Not currently available in Faspex 5.

Replicate package contents in a dropbox

Faspex 4

```
POST /api/dropboxes/{dropbox_id}/packages/{package_delivery_id}/replicate_contents
```

Faspex 5

Not currently available in Faspex 5.

Delete the files and folders from a package

Faspex 4

```
POST /api/users/{user_id}/packages/{package_id}/delete_contents
```

Faspex 5

Not currently available in Faspex 5.

Rename package contents

Faspex 4

```
POST /api/users/{user_id}/packages/{package_id}/rename_contents
```

Faspex 5

Not currently available in Faspex 5.

Get transfer specification needed for a package transfer

Faspex 4

```
POST api/users/{user_id}/packages/{package_id}/transfer_specs
```

Faspex 5

Uploads:

```
POST /api/v5/packages/{package_id}/transfer_spec/upload
```

Reference

Downloads:

```
POST /api/v5/packages/{package_id}/transfer_spec/download
```

Reference

Get package transfer history (for transfer sessions) for users

Faspex 4

```
GET /api/users/{user_id}/packages/{package_delivery_id}/transfers
```

Faspex 5

Uploads:

```
POST /api/v5/packages/{package_id}/upload_details
```

Reference

Downloads:

```
POST /api/v5/packages/{package_id}/download_details
```

Reference

Get package transfer history (for transfer sessions) for dropboxes

Faspex 4

```
GET /api/dropboxes)/{dropbox_id}/packages/{package_delivery_id}/transfers
```

Faspex 5

Uploads:

```
POST /api/v5/packages/{package_id}/upload_details
```

Reference

Downloads:

```
POST /api/v5/packages/{package_id}/download_details
```

Reference

Initiate a remote content upload for a user package

Faspex 4

```
POST /api/users/{user_id}/packages/{package_delivery_id}/transfers
```

Faspex 5

```
POST /api/v5/packages/{package_id}/remote_transfer
```

Reference

Nodes API

Shares are now called shared folders or storage locations in Faspex 5.

Endpoint

API calls

Get all nodes

Faspex 4

```
GET /api/nodes
```

Faspex 5

```
GET /api/v5/nodes
```

Reference

Get shares for a given node

Faspex 4

```
GET /api/nodes/{node_id}/shares
```

Faspex 5

```
GET /api/v5/nodes/{node_id}/shared_folders
```

Reference

Metadata profiles

Get current metadata profiles

Faspex 4

```
GET /api/metadata_profiles
```

Faspex 5

```
GET /api/v5/configuration/metadata_profiles
```

Reference

Get a metadata profile

Faspex 4

```
GET /api/metadata_profiles/{id}
```

Faspex 5

```
GET /api/v5/configuration/metadata_profiles/{id}
```

Reference

Get the default metadata profile

Faspex 4

```
GET /api/metadata_profiles/default_profile
```

Faspex 5

```
GET /api/v5/configuration/metadata_profiles/default_profile
```

Reference

Validate metadata

Faspex 4

```
POST /api/metadata_profiles/{id}/validate_metadata
```

Faspex 5

Not currently available in Faspex 5.

Users API

Endpoint

API calls

Get current user info

Faspex 4

```
GET /me
```

Faspex 5

```
GET /api/v5/account
```

Reference

Sign on (API v.3)

Faspex 4

```
POST /aspera/faspex/signon/
```

Faspex 5

Not currently available in Faspex 5.

Find available login methods (API v.3)

Faspex 4

```
GET /login/new
```

Faspex 5

```
GET /api/v5/saml_configs
```

Reference

Get user contacts (API v.3)

Faspex 4

```
GET /aspera/faspex/address-book
```

Faspex 5

```
GET /api/v5/contacts
```

Reference

Get users

Faspex 4

```
GET /api/users
```

Faspex 5

```
GET /api/v5/accounts
```

Reference

Create user

Faspex 4

```
POST /aspera/faspex/api/users
```

Faspex 5

```
POST /api/v5/accounts
```

Reference

Update user

Faspex 4

```
PUT /aspera/faspex/api/users/{id}
```

Faspex 5

```
PUT /api/v5/accounts/{id}
```

Reference

Delete user

Faspex 4

```
DELETE /aspera/faspex/api/users/{id}
```

Faspex 5

```
DELETE /api/v5/accounts/{id}
```

Reference

Get user configuration

Faspex 4

```
GET /api/users/{id}/configuration or GET /api/users/me/configuration
```

Faspex 5

```
GET /api/v5/accounts/{id}
```

Reference

Shared inboxes (formerly dropboxes) API

List all dropboxes to which an account has access (API v.3)

Faspex 4

```
GET /aspera/faspex/dropboxes
```

Faspex 5

```
GET /api/v5/shared_inboxes
```

Reference

Get information about a dropbox (API v.3)

Faspex 4

```
GET /aspera/faspex/dropboxes/{dropbox_id}
```

Faspex 5

```
GET /api/v5/shared_inboxes/{id}
```

Reference

Dropbox and workgroup memberships API

Endpoint

API calls

Get current dropbox memberships

Faspex 4

```
GET /api/dropbox_memberships
```

Faspex 5

- Members:

  ```
  GET /api/v5/shared_inboxes/{shared_inbox_id}/members
  ```

  Reference

- SAML groups:

  ```
  GET /api/v5/shared_inboxes/{shared_inbox_id}/saml_groups
  ```

  Reference

Get current workgroup memberships

Faspex 4

```
GET /api/workgroup_memberships
```

Faspex 5

- Members:

  ```
  GET /api/v5/workgroups/{shared_inbox_id}/members
  ```

  Reference
- SAML groups:

  ```
  GET /api/v5/workgroups/{shared_inbox_id}/saml_groups
  ```

  Reference

Get specific dropbox membership

Faspex 4

```
GET /api/dropbox_memberships/{id}
```

Faspex 5

Not currently supported in Faspex 5. Use the API calls to show all members and find the specific member in the response.

- Members:

  ```
  GET /api/v5/shared_inboxes/{shared_inbox_id}/members
  ```

  Reference
- SAML groups:

  ```
  GET /api/v5/shared_inboxes/{shared_inbox_id}/saml_groups
  ```

  Reference

Get specific workgroup memberships

Faspex 4

```
GET /api/workgroup_memberships/{id}
```

Faspex 5

Not currently supported in Faspex 5. Use the API calls to show all members and find the specific member in the response:

- Members:

  ```
  GET /api/v5/workgroups/{shared_inbox_id}/members
  ```

  Reference

- SAML groups:

  ```
  GET /api/v5/workgroups/{shared_inbox_id}/saml_groups
  ```

  Reference

Create a dropbox

Faspex 4

```
POST /api/dropbox_memberships
```

Faspex 5

```
POST /api/v5/shared_inboxes
```

Reference

Create a workgroup

Faspex 4

```
POST /api/workgroup_memberships
```

Faspex 5

```
POST /api/v5/workgroups
```

Reference

Delete a dropbox membership

Faspex 4

```
DELETE /api/dropbox_memberships/{id}
```

Faspex 5

Member:

```
DELETE /api/v5/shared_inboxes/{shared_inbox_id}/members/{member_id}
```

Reference

SAML group:

```
DELETE /api/v5/shared_inboxes/{shared_inbox_id}/saml_groups/{saml_group_id}
```

Reference

Delete a workgroup membership

Faspex 4

```
DELETE /api/workgroup_memberships/{id}
```

Faspex 5

Member:

```
DELETE /api/v5/workgroups/{shared_inbox_id}/members/{member_id}
```

Reference

SAML group:

```
DELETE /api/v5/workgroups/{shared_inbox_id}/saml_groups/{saml_group_id}
```

Reference

SAML configuration API

Endpoint

API calls

Get user profile

Faspex 4

```
GET /api/user_profile_fields/{saml_configuration_id}
```

Faspex 5

```
GET /api/v5/saml_configs/{saml_config_id}/user_profile_fields
```

Reference

Get current SAML configurations

Faspex 4

```
GET /api/saml_configurations
```

Faspex 5

```
GET /api/v5/saml_configs
```

Look for the configuration that has `default` as `true`.

Reference

Get a SAML configuration

Faspex 4

```
GET /api/saml_configurations/{id}
```

Faspex 5

```
GET /api/v5/saml_configs/{id}
```

Reference

Faspex services API

Endpoint

API calls

XRDS Discovery Service

Faspex 4

```
GET /
```

Authorization: Basic

Accept: application/xrds+xml

Faspex 5

Faspex services are now run in the `faspex_services` container and not discoverable through the API. You can monitor background jobs using the

```
GET /aspera/faspex/health
```

Authentication and authorization

Authorizing to Faspex

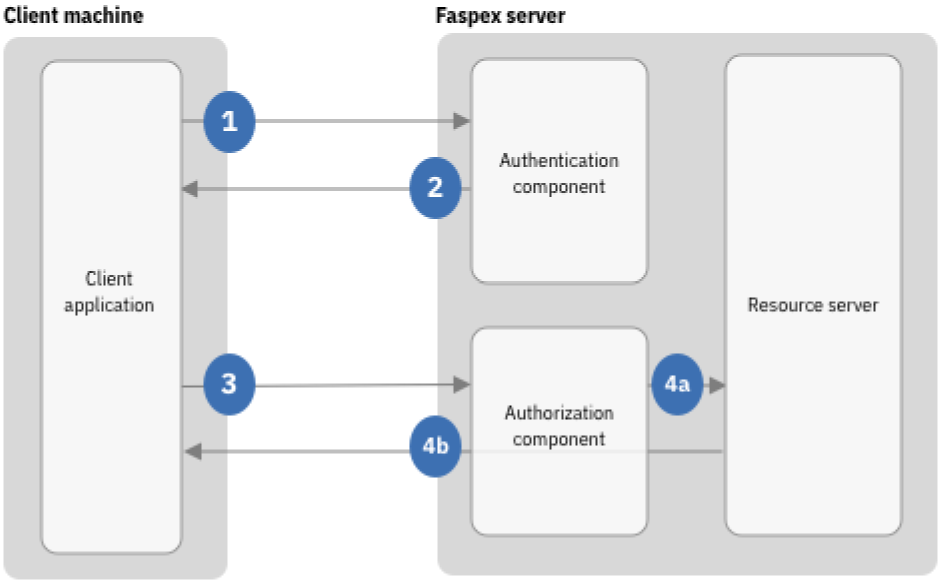
Faspex requires your application use OAuth 2 to authorize your application to access protected Faspex resources.

OAuth 2 overview

Faspex protected resources require you to provide a bearer token in the authorization header of a request. For example:

```
"Authorization: Bearer
eyJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoyLCJleHAiOjE1OTYwMzY3NzV9.068OqoSZoTLYYMpEjYf5poK0hxVEYpktaA
Yx7hBKP9I"
```

To retrieve a token, you must implement one of the supported OAuth 2 methods. Each method uses a workflow that at its core has these steps:



1. Your client application authorizes to a registered OAuth 2 client (configured based on the authorization method) through the Faspex authentication component.
2. Your user is redirected to the Faspex UI login page and securely submits the credentials to log in. On performing this step, the user authorizes your client application to access the Faspex API on your user's behalf.
3. Faspex provides your client application with a bearer token.
4. Your client application uses the bearer token to request access to protected endpoints on the Faspex resource server.
5. The Faspex authorization component confirms the validity of the bearer token and returns the requested resource.

The client application (your web client) does not submit the credentials to the Faspex authentication component. The faspex UI login page securely submits the credentials for authentication to Faspex.

OAuth 2 methods

Before your application can authorize to Faspex using OAuth 2, you must first register an API client for your application. Faspex supports these OAuth 2 methods:

OAuth 2 with SAML
Web applications requiring users to authenticated through a configured SAML identity provider (SAML IdP).

OAuth 2 PKCE (Proof Key for Code Exchange)
Web and mobile applications requiring users to enter credentials into a user login page, which then authenticates to the Faspex server.

Note: The Faspex UI acts as an OAuth 2 client to authenticate to the Faspex API server. The Faspex UI is a pre-registered OAuth 2 client.

OAuth 2 JWT (JSON Web Token Grant)
Non-web applications that do not require access to user-protected endpoints, such as an application that monitors background jobs.

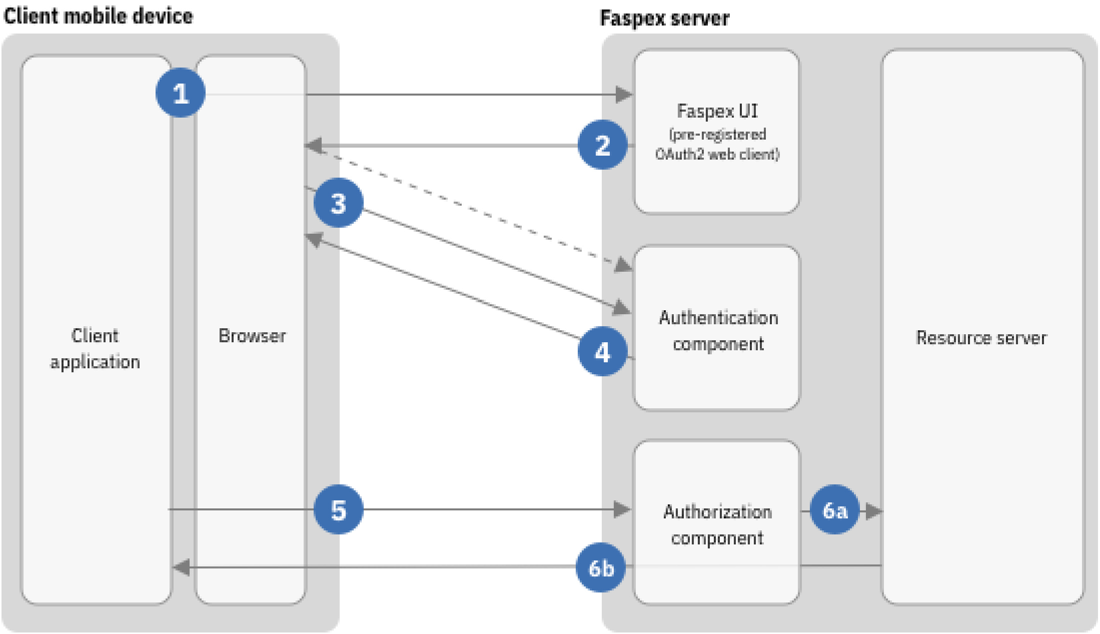
Configuring OAuth 2 for user-based workflows

Authorize your users using the PKCE method. Use PKCE authorization for securing integrations, especially with mobile applications.

PKCE authorization flow

Using PKCE protects against man-in-the-middle attacks that intercept the response from the OAuth 2 client and hijack the redirect URL to get the access token.

Authorization flow for a mobile device authorizing with Faspex UI



1. An end user connects to your application.
2. Your application redirects the user to the Faspex login page, served by the Faspex authentication component.
3. The user enters their credentials on the login page.
4. Your application submits those credentials along with the code verifier (matching the code challenge stored on the registered OAuth 2 client) to the Faspex authentication component to retrieve a bearer token.

5. As the now authenticated user interacts with your application, your application makes API calls to the Faspex resource server using the bearer token.
6. The Faspex authorization component server authorizes and serves the API call based on the identity associated with the bearer token.

To configure OAuth 2 for your application:

1. Register a new API client for your web application using the Faspex UI.
   a) Log in through the UI as an admin user.
   b) Go to the admin app.
   c) Go to **Configurations > API clients** and click **Create new**.
   d) Enter a **Name** and any **Redirect URIs** your web application uses.

      A redirect URI, or reply URL, is the location where the Faspex API sends your users once the Faspex login page has successfully authorized and granted an authorization code. Faspex sends the code to the redirect URI, so the redirect URI should be the endpoint your web application uses to retrieve and store an access token for your users.

      A standard endpoint is `https://`*`server`*`/token`.

      Note: Faspex requires redirect URIs to use the HTTPS format. If you do not have valid certificates to use for your application while in development, you can use self-signed certificates.
   e) Click **Save**.
2. Save the generated Client ID.
3. Implement the redirect URI endpoint in your web application.

   Your mobile application must accept two parameters, `code` and `state`, and use the values of those parameters to request a token from the API server using the `aspera/faspex/auth/token` endpoint.

   The sample Ruby code below returns a valid bearer token for use in subsequent API calls:

```
require "net/http"
require "json"

# Expect code and state from API call
authorization_code = params["code"]
state = params["state"]

# Set variables from environment
faspex_hostname = ENV["FASPEX_HOSTNAME"]
faspex_client_id = ENV["FASPEX_CLIENT_ID"]
app_redirect_uri = ENV["APP_REDIRECT_URI"]
code_verifier = ENV["CODE_VERIFIER"]

# Build token request header and payload
body = {
  "client_id": ENV["FASPEX_CLIENT_ID"],
  "grant_type": "authorization_code",
  "code": authorization_code,
  "code_verifier": Base64.encode64(params["state"]), # In this example, we assume the
verifier and the state are the same
  "redirect_uri": ENV["APP_REDIRECT_URI"],
  "state": state
}
# Make a GET request to /aspera/faspex/auth/token
uri = URI("https://#{FASPEX_HOSTNAME}/aspera/faspex/auth/token")
http = Net::HTTP.new(uri.host, uri.port)
http.use_ssl = true
http.verify_mode = OpenSSL::SSL::VERIFY_NONE # If using self-signed certs
request = Net::HTTP::Post.new(uri.request_uri)
request.body = body.to_json
request.content_type = "application/json"
response = http.request(request)
# Get bearer token from the response
JSON.parse(response.body)["access_token"]
```

4. Test your authorization flow.

a) Start your web application in HTTPS mode so that it can receive the authorization code and state at its `/token` endpoint:

```
FASPEX_HOSTNAME=faspex_hostname FASPEX_CLIENT_ID=faspex_hostname
APP_REDIRECT_URI=https://your_server/token puma -b 'ssl://localhost:8443?
key=ssl_certificate_key&cert=ssl_certificate_crt'
```

b) Request an authorization code by logging in to Faspex. Use the client ID and redirect ID from the registered API client to generate a login URL.

The URL syntax is:

```
GET https://server/aspera/faspex/auth/authorize?
client_id=faspex_client_id&redirect_uri=redirect_uri&response_type=code&state=state&code_c
hallenge=code_challenge&code_challenge_method=code_challenge_method"
```

Parameter	Definition	Example
`client_id`	The ID of the API client you registered.	`266c8e7b-8bcb-40c2-b605-078b46c39d2a`
`redirect_uri`	The location where the Faspex API sends your users once the Faspex login page has successfully authorized and granted an authorization code.	`https://faspex5.example.com/token`
`state`	Unique identifier used in the `/auth/token` request to prove to the API server the client requesting the bearer token is the same client that received the authorization code.	`96339c21-7d10-4d79-8043-93e7ab4cbe52`
`code_challenge`	The code challenge is a Base64-encoded SHA256 hash of the code verifier. The code challenge is used for PKCE requests.	`ZWYyYmJlMmMtODA4MC00NWQ3LThiN2QtYTY1YmZjOGY5Mjkz\n` (Code verifier: `ef2bbe2c-8080-45d7-8b7d-a65bfc8f9293`)
`code_challenge_method`	The code challenge method used to generate code challenge.	`s256`

c) After successful login, your application should provide you a valid bearer token. Use that token to retrieve your account information:

Example **`curl`** command:

```
curl "https://server/aspera/faspex/api/v5/account" \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer token"
```

Example Ruby code:

```
require "net/http"

access_token = access_token
uri = URI("https://#{ENV['FASPEX_HOST']}/aspera/faspex/api/v5/account")
http = Net::HTTP.new(uri.host, uri.port)
http.use_ssl = true
http.verify_mode = OpenSSL::SSL::VERIFY_NONE # If using self-signed certs
request = Net::HTTP::Get.new(uri.request_uri)
request.content_type = "application/json"
request["Authorization"] = "Bearer #{access_token}"
http.request(request)
```

Configuring OAuth 2 for non-user-based workflows (JWT)

Authorize using the JSON web token (JWT) grant method for application-to-application workflows.

JWT authorization flow

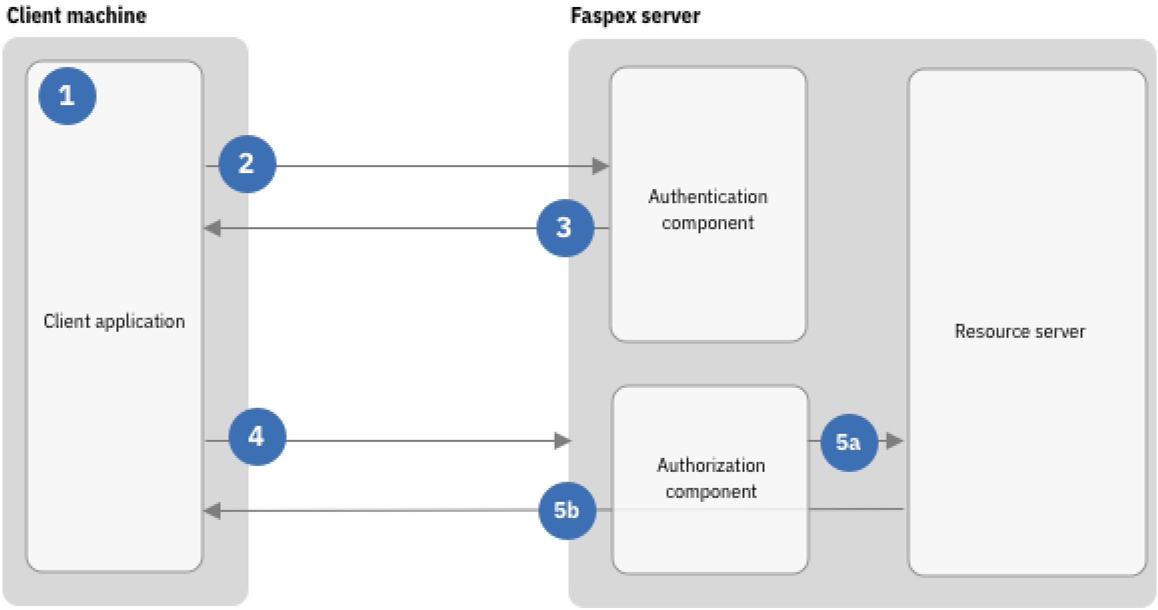
Aspera Faspex 5 supports the JSON Web Token-based OAuth 2.0 grant type (a grant type specifies how a client interacts with an identity server to authorize access to protected resources) to enable client applications to use the API without a user having to log in from a web browser.

A JSON web token includes, among other things, a user ID (the email of the user to authenticate), and is signed with a valid private key. In response to a valid JWT request submitted to the API, Faspex returns an access token for the user identified in the JWT token request. Further requests to the API must include this access token to be valid. Requests as a specific user are permitted according to that user's permissions and access in the Faspex application.

Note: JWT authentication for a given Faspex user requires either a user-specific public/private key pair, or a client-level 'global' (not user-specific) public/private key pair, depending on your configuration. Creating a public/private key pair is a simple, standardized process; search the web for a procedure appropriate to your platform.

When using JWT, you do not have to first authorize to an OAuth 2 client. You directly request a token from the OAuth 2 client. The Faspex Service application uses this authorization flow.

Authorization flow for a client authorizing with JWT



1. Your client application generates a JWT assertion.
2. Your client signs the assertion with the private key paired with the public key stored in the registered OAuth 2 client.
3. Your client sends the JWT assertion to the Faspex authentication component to retrieve a bearer token.
4. The now authenticated application makes API calls to the Faspex resource server using the bearer token.
5. The Faspex authorization component server (a) authorizes and (b) serves the API call based on the bearer token.

Instead of manually generating a JWT payload, you can use an existing JWT library. See https://jwt.io/libraries.

To set configure OAuth 2 for your application:

1. Generate a new SSH key-pair (or use an existing one). The SSH key-pair must be in PEM format.
   a) Generate the key-pair and save it in a location easily accessible by your application:

      ```
      ssh-keygen -t rsa -m PEM -f location_and_key_name
      ```

      For example, if you use `certs/test_rsa` as the location, the command generates the `test_rsa` private key and `test_rsa.pub` public key files.

   b) Convert your public key to PEM format:

      ```
      ssh-keygen -f public_key -e -m pem
      ```

   c) Use the output in registering your API client.
2. Register a new API client for your application using the Faspex UI.
   a) Log in through the UI as an admin user.

   b) Go to the admin app.

   c) Go to **Configurations > API clients** and click **Create new**.

   d) Enter a **Name**.

   e) Enable **Enable JWT grant type**.
3. Set an expiration for the access token.

   The access token expiration defines the maximum duration (in hours) of an active session unless the refresh token duration (see following parameter) is configured to extend the session. Unless refresh tokens are configured, users must re-authenticate when the login token expires.
4. Configure who can use your application to access endpoints that require user access.

   To allow all your Faspex users to access the API with your application, set the toggle labeled **All users** to **On**.

   ⚠ **CAUTION:** Enabling this feature allows anyone with an authorized token to impersonate *any* Faspex user using the user's email address.

   To allow only specified users to access the API, set this toggle to **Off** and enter the specific user names.
5. Configure which keys an authorized user can use to sign the JWT payload.

   To allow use of the global key, set the toggle labeled **Global key** to **On** and enter the global public key (in PEM format) in the field. The global key is tied to the client just as the user-specific key is, but the global key can be applied to all users.

   To require users to sign with their own user-specific key, set the toggle labeled **Global key** to **Off**. Be sure that your application users have their user-specific key entered in their account (**Account settings > Preferences > Public key**).
6. Click **Create**.
7. Save the generated Client ID.
8. Implement the authentication flow in your application.

   Your web application must send a JWT payload to request a token from the API server using the `aspera/faspex/auth/token` endpoint.

   The sample Ruby code below uses the `ruby-jwt` gem to and requests a valid bearer token for use in subsequent API calls:

   ```
   require "jwt"
   require "uri"
   require "net/http"
   ```

```
require "json"
require "openssl"
require "time"
require "securerandom"
require "base64"

private_key_file ='/root/faspex5'
FASPEX_CLIENT_ID = ''
FASPEX_HOSTNAME = 'test.faspex5.com'
APP_REDIRECT_URI = 'https://test.faspex5.com/aspera/faspex/'
email = 'user@aspera.com'

private_key = OpenSSL::PKey::RSA.new(File.read(private_key_file))

client_id = FASPEX_CLIENT_ID
time = Time.now.to_i

sub = "user:#{email}"

payload = {
  iss: client_id,
  sub: sub,
  aud: client_id,
  jti: SecureRandom.uuid,
  exp: time + 103600,
  iat: time
}

token = JWT.encode(payload, private_key, "RS256", { typ: "JWT" })

uri = URI("https://#{FASPEX_HOSTNAME}/aspera/faspex/auth/token")
body = {
  client_id: client_id,
  grant_type: "urn:ietf:params:oauth:grant-type:jwt-bearer",
  redirect_uri: ENV["APP_REDIRECT_URI"],
  state: SecureRandom.uuid,
  assertion: token
}
# Make a POST request to /aspera/faspex/auth/token
uri = URI("https://#{FASPEX_HOSTNAME}/aspera/faspex/auth/token")
http = Net::HTTP.new(uri.host, uri.port)
http.use_ssl = true
http.verify_mode = OpenSSL::SSL::VERIFY_NONE # If using self-signed certs
request = Net::HTTP::Post.new(uri.request_uri)
request.body = body.to_json
request.content_type = "application/json"
response = http.request(request)
# Get bearer token from the response
puts JSON.parse(response.body)["access_token"]
```

9. The application code above should provide you a valid bearer token. Use that token to check the health of your server:

   Example **curl** command:

   ```
   curl "https://server/aspera/faspex/health" \
     -H "Content-Type: application/json" \
     -H "Authorization: Bearer token"
   ```

   Example Ruby code:

   ```
   access_token = access_token
   uri = URI("https://#{ENV['FASPEX_HOST']}/aspera/faspex/health")
   http = Net::HTTP.new(uri.host, uri.port)
   request = Net::HTTP::Get.new(uri.request_uri)
   http.use_ssl = true
   http.verify_mode = OpenSSL::SSL::VERIFY_NONE # If using self-signed certs
   request.content_type = "application/json"
   request["Authorization"] = "Bearer #{access_token}"
   http.request(request)
   ```

Sending packages

Sending a package with Connect

Use the Faspex API and Connect SDK to send a package to a recipient.

Before following the steps in this guide:

- Install IBM Aspera Connect version 3.10 or later.
- Implement authorization in your application (see "Authorizing to Faspex" on page 191).

Note:

- The code examples for the Faspex API are written with `curl`.
- The code examples for the Connect SDK are written in JavaScript.

1. Retrieve a bearer token from the Faspex server.
2. Create a package in Faspex:

   Endpoint: POST `https://faspex5.example.com/aspera/faspex/api/v5/packages`

   Example:

   ```
   curl -X POST 'https://faspex5.example.com/aspera/faspex/api/v5/packages/' \
   -H 'Content-Type: application/json' \
   -H "Authorization: Bearer
   eyJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoyLCJleHAiOjE1OTYwMzY3NzV9.068OqoSZoTLYYMpEjYf5poK0hxVEYpk
   taAYx7hBKP9I"
   -d '{
     "title": "Example Package 1",
     "recipients": [
       "name":"admin@ibm.com" // This example uses the authenticated user as both sender and
   recipient
     ]
   }'
   ```

   Result:

   If successful, Faspex returns a response with the package information:

   ```
   {
     "id": "43",
     "title": "Example Package 1",
     "note": "",
     "recipients": [
       {
         "recipient_type": "user",
         "id": "256",
         "name": "admin@ibm.com",
         "first_name": "John",
         "last_name":
   "Doe",


                                                                          "email":
   "jhwan@us.ibm.com"
       }
     ],
     "release_policy": "now",
     "release_date": "2022-02-18T21:30:05.000+0000",
     "sender": "admin@ibm.com",
     "state": "held",
     "prevent_http_download": false,
     "archived": false,
     "obfuscation_enabled": false,
     "ear_enabled": null,
     "notified_on_upload": [],
     "notified_on_download": [],
     "notified_on_receipt": [],
     "active_downloads": 0,
     "active_downloaders": [],
     "download_count": 0,
     "downloaders": [],
     "total_bytes": 0,
   ```

```
  "total_files": 0,
  "recalculation_needed": false,
  "recalculation_in_progress": false,
  "creation_date": "2022-02-18T21:30:05.000+0000",
  "last_modified": "2022-02-18T21:30:05.000+0000",
  "package_uuid": "feedbdbc-0468-4c9d-bd49-0873171683eb",
  "expiration_policy": "none",
  "mailbox": "inbox"
}
```

3. Use the Connect SDK and the retrieved transfer specification to send files.

   a) Create a new JS file with the sample code for `upload.js` and change the constants using the results from the previous steps:

   - `FASPEX_HOSTNAME`: The hostname of your Faspex server.
   - `BEARER_TOKEN`: The bearer token generated when you authenticated to the Faspex server.
   - `PACKAGE_ID`: The ID of the package you created on the Faspex server.

   `upload.js`

```
//* Change these constants */
const FASPEX_HOSTNAME = "https://faspex_hostname";
const BEARER_TOKEN = "bearer_token";
const PACKAGE_ID = "package_id";

/*
 * fetchTransferSpec() - Fetch a transfer specification for HTTP Gateway to
 * use to send a package.
 *
 * Required params:
 *   filepaths - array of filepaths with format:
 *     { paths: [ { source: /path/to/file } ] }
 */
async function fetchTransferSpec(filepaths) {
  // Retrieve upload transfer specification for Connect
  const ts_url = `${FASPEX_HOSTNAME}/aspera/faspex/api/v5/packages/${PACKAGE_ID}/
transfer_spec/upload?transfer_type=connect`

  const ts_response= await fetch(ts_url, {
    method: "POST",
    headers: {
      "Content-Type": "application/json",
      "Authorization": `Bearer ${BEARER_TOKEN}`
    },
    body: JSON.stringify(filepaths)
  });

  let ts_data = await ts_response.json();

  console.log("Transfer specification", ts_data)
  ts_data["paths"] = filepaths["paths"];


  // If successful, Faspex returns a transfer specification
  return ts_data;
}

/*
 * uploadFile() is a callback that calls fetch_transfer_spec() with the chosen
 * file and transfers the package with the retrieved transfer specification
 */
function uploadFile(data) {
  const { files } = data.dataTransfer;
  if (files.length === 1) {
    let fileToUpload = files[0].name;
    const filepaths = {
      paths: [
        {
          source: `${fileToUpload}`
        }
      ]
    };

    fetchTransferSpec(filepaths).then(transferSpec => {
      this.client.startTransfer(transferSpec);
    });

  }
}

/*
 * upload() opens the Connect file picker and calls uploadFile() to transfer the
 * picked file
 *
 * This function is triggered by a button in `upload.html`.
 *
 */
function upload() {
  const options = {
    allowMultipleSelection: false
  };
  /* Display a file browser for the user to select a file. */
  this.client.showSelectFileDialogPromise(options)
    .then(uploadFile)
    .catch(() => {
      console.error('Unable to select files');
    });
}

/* Initializes the Connect client from the Connect SDK.
 *
 * Used in `upload.html` on page load.
 *
 */
function initAsperaConnect() {
  this.client = new AW4.Connect();
  this.client.initSession();
}
```

b) Create a new HTML file with the sample code in `upload.html` in the same directory.

`upload.html`

```
<!DOCType HTML>
<html>
  <head>
    <title>Testing Connect Upload</title>
    <style>
div {
  margin-left: auto;
  margin-right: auto;
  width: 8rem;
}

button {
  width: 8rem;
}
    </style>
    <script src="https://d3gcli72yxqn2z.cloudfront.net/connect/v4/asperaweb-4.min.js"></
script>
    <!-- the src below must point to the JS file you created -->
    <script type="text/javascript" src="upload.js"></script>
  </head>
  <body onload="initAsperaConnect();">
    <h1 style="text-align:center;">IBM Aspera Connect simple upload example</h1>
    <h2 style="text-align:center;">Connect must be installed.</h2>
    <div>
      <button onclick="upload()">Send a file</button>
    </div>
  </body>
</html>
```

c) Open the HTML file, pick a file, and send.

Note:

- You must have Connect running on your computer for the transfer to start.
- Connect requires users use the Connect file picker to select files as a security measure.

Connect sends the package to the recipient Faspex user.

Sending a package with HTTP Gateway

Leverage the Faspex API and the HTTP Gateway SDK to send a package to a recipient.

Before following the steps in this guide:

- Set up an HTTP Gateway server.
- Download the HTTP Gateway Javascript SDK.
- Implement authorization in your application (see “Authorizing to Faspex” on page 191).

Note:

- The code examples for the Faspex API are written with `curl`.
- The code examples for the HTTP Gateway SDK are written in JavaScript.

1. Retrieve a bearer token from the Faspex server.
2. Create a package in Faspex:

   Endpoint: POST `https://faspex5.example.com/aspera/faspex/api/v5/packages`

   Example:

```
curl -X POST 'https://faspex5.example.com/aspera/faspex/api/v5/packages/' \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer
eyJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoyLCJleHAiOjE1OTYwMzY3NzV9.068OqoSZoTLYYMpEjYf5poK0hxVEYpk
taAYx7hBKP9I"
-d '{
  "title": "Example Package 1",
  "recipients": [
    "name":"admin@ibm.com" // This example uses the authenticated user as both sender and
recipient
  ]
}'
```

   Result:

   If successful, Faspex returns a response with the package information:

```
{
  "id": "43",
  "title": "Example Package 1",
  "note": "",
  "recipients": [
    {
      "recipient_type": "user",
      "id": "256",
      "name": "admin@ibm.com",
      "first_name": "John",
      "last_name":
"Doe",


                                                                       "email":
"jhwan@us.ibm.com"
    }
  ],
  "release_policy": "now",
  "release_date": "2022-02-18T21:30:05.000+0000",
  "sender": "admin@ibm.com",
  "state": "held",
  "prevent_http_download": false,
  "archived": false,
  "obfuscation_enabled": false,
  "ear_enabled": null,
  "notified_on_upload": [],
  "notified_on_download": [],
  "notified_on_receipt": [],
  "active_downloads": 0,
  "active_downloaders": [],
  "download_count": 0,
  "downloaders": [],
  "total_bytes": 0,
  "total_files": 0,
  "recalculation_needed": false,
  "recalculation_in_progress": false,
  "creation_date": "2022-02-18T21:30:05.000+0000",
  "last_modified": "2022-02-18T21:30:05.000+0000",
  "package_uuid": "feedbdbc-0468-4c9d-bd49-0873171683eb",
  "expiration_policy": "none",
  "mailbox": "inbox"
}
```

3. Use the HTTP Gateway SDK to retrieve the transfer specification from Faspex and send the package. Copy the sample code below and change the constants using the results from the previous steps.

   a) In `upload.js`, change:

      - `FASPEX_HOSTNAME`: The hostname of your Faspex server.
      - `BEARER_TOKEN`: The bearer token generated when you authenticated to the Faspex server.
      - `PACKAGE_ID`: The ID of the package you created on the Faspex server.
      - `GATEWAY_HOSTNAME`: The hostname of your HTTP Gateway server.

      `upload.js`

```
/* Change these constants */
const FASPEX_HOSTNAME = "https://faspex_hostname";
const BEARER_TOKEN = "bearer_token";
const PACKAGE_ID = "package_id";
const GATEWAY_HOSTNAME = "https://http_gateway_hostname";

/* Constants used for picking files */
const formId = 'send-file';
const files = [];

/*
 * fetchTransferSpec() - Fetch a transfer specification for HTTP Gateway to use
 * to send a package.
 *
 * Required params:
 *   filepaths - array of filepaths with format:
 *     { paths: [ { source: /path/to/file } ] }
 */
async function fetchTransferSpec(filepaths) {
  // Retrieve upload transfer specification for HTTP Gateway
  const ts_url = `${FASPEX_HOSTNAME}/aspera/faspex/api/v5/packages/${PACKAGE_ID}/
transfer_spec/upload?transfer_type=http_gateway`
  const ts_response= await fetch(ts_url, {
    method: "POST",
    headers: {
"Content-Type": "application/json",
"Authorization": `Bearer ${BEARER_TOKEN}`
    },
    body: JSON.stringify(filepaths)
  });
  let ts_data = await ts_response.json()

  console.log("Transfer specification", ts_data)

  // If successful, Faspex returns a transfer specification
  return ts_data
}

/*
 * filePickCallback is a callback to save the user-selected file from the
 * browser file picker and to append the filename to the HTML document
 */
function filePickCallback(data) {
  const { files } = data.dataTransfer
  const files_for_spec = [];
  for (let file of files) {
    files_for_spec.push(file);
  }
  document.querySelector("#send-file").innerHTML = files_for_spec[0].name
  this.files = files_for_spec
  console.log('Files picked', files);
};

/*
 * pickFile() opens the browser file picker and saves the user selection.
 *
 * Used in `upload.html` to trigger the browser file picker.
 *
 */
function pickFile() {
  this.files = this.client.getFilesForUpload(filePickCallback, this.formId)
};
```

```
/* monitorTransfers is a callback to print transfer progress to Console */
const monitorTransfers = (result) => {
  result.transfers.forEach(transfer => {
    console.log(
      `New Transfer:
- Percent: ${transfer.percent * 100}%,
- Status: ${transfer.status},
- Data Sent: ${transfer.bytes_written},
- Data Total: ${transfer.bytes_expected}
`
    );
  });
  console.log("Transfer completed")
}

/*
 * upload() registers the callback and uploads the picked file
 *
 * This function triggered by a button in `upload.html`
 *
 */
async function upload() {
  this.client.registerActivityCallback(monitorTransfers);
  console.log("Registered callback to monitor transfers")
  if (this.files.length === 1) {
    let fileToUpload = this.files[0].name;
    const filepaths = {
      "paths": [
        fileToUpload
      ]
    }
    fetchTransferSpec(filepaths).then(transferSpec => {
      this.client.upload(transferSpec, this.formId).then(response => {
        console.log('Upload started', response);
      }).catch(error => {
        console.log('Upload could not start', error);
      });
    })
  }
}

/*
 * initHttpGateway() initializes the HTTP Gateway client from the HTTP Gateway SDK.
 *
 * Used in `upload.html` on page load.
 */
function initHttpGateway() {
  this.client = asperaHttpGateway
  const gateway_url = `${GATEWAY_HOSTNAME}/aspera/http-gwy/v1/`
  this.client.initHttpGateway(gateway_url).then(response => {
    console.log('HTTP Gateway SDK started', response);
  }).catch(error => {
    console.warn('HTTP Gateway SDK did not start', error);
  })
}
```

b) In `upload.html`, change the path to the HTTP Gateway Javascript SDK (find *`/path/to/http-gateway.js`*).

`upload.html`

```
<!DOCType HTML>
<html>
  <head>
    <title>Testing HTTP Gateway Upload</title>
    <style>
div {
  margin-left: auto;
  margin-right: auto;
  width: 8rem;
}

button {
  width: 8rem;
}
    </style>
    <script src="/path/to/http-gateway.js"></script>

    <script type="text/javascript" src="upload.js"></script>
  </head>
  <body onload="initHttpGateway();">
    <h1 style="text-align:center;">IBM Aspera HTTP Gateway simple upload example</h1>
    <div>
      <button onclick="pickFile()">Pick a file to send</button>
    </div>
    <div>
      <p>File picked:
      <p id="send-file"></p>
    </div>
    <div>
      <button onclick="upload()">Send file</button>
    </div>
  </body>
</html>
```

c) Open the HTML file, pick a file, and send.

HTTP Gateway sends the package to the recipient Faspex user.

Downloading packages

Downloading a package with Connect

Use the Faspex API and Connect SDK to download a package.

Before you begin, install IBM Aspera Connect version 3.10 or later.

Note:

- The code examples for the Faspex API are written with `curl`.
- The code examples for the Connect SDK are written in JavaScript.

1. Retrieve a bearer token from the Faspex server.
2. List the packages you've received and choose the package you want to download.

   Endpoint: GET `https://faspex5.example.com/aspera/faspex/api/v5/inbox/packages`

   Example:

```
curl 'https://faspex5.example.com/aspera/faspex/api/v5/inbox/packages/' \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer
eyJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoyLCJleHAiOjE1OTYwMzY3NzV9.068OqoSZoTLYYMpEjYf5poK0hxVEYpk
taAYx7hBKP9I"
```

   Result:

```
{
  "total_count":116,
  "offset":0,
  "limit":20,
  "packages": [
    {
      "id": 66,
      "title":" Example Package 1",
      "release_date":" 2020-09-23T17:14:26.000Z",
      "notified_on_download":"",
      "total_bytes": 0,
      "total_files": 0,
      "message":"",
      "upload_status":" Submitted",
      "created_time":" 2020-09-23T17:14:26.000Z",
      "upload_completion_time": null,
      "sender":" admin",
      "metadata":  {},
      "package_uuid":" c1568306-f92d-49bc-9bca-ab4904a96ef1",
      "recipients":  [
        {
          "id":" 2",
          "name":" admin"
        }
      ],
      "completed_download_count": 2,
      "active_download_count": 0,
      "completed_partial_download_count": 0
    },
    {
      "id": 67,
      "title": "Example Package 1",
      "release_date": "2020-09-23T17:14:02.000Z",
      "notified_on_download": "",
      "total_bytes": 0,
      "total_files": 0,
      "message": "",
      "upload_status": "Submitted",
      "created_time": "2020-09-23T17:14:02.000Z",
      "upload_completion_time": null,
      "sender": "admin",
      "metadata": {},
      "package_uuid": "8e85b3ea-7537-4832-a11c-767c1a1742c8",
        "recipients":
        [
          {
            "id":"2",
            "name":"admin"
          }
      ],
      "completed_download_count": 0,
      "active_download_count": 0,
      "completed_partial_download_count": 0
    },

    ...

  ]
};
```

3. Using the package ID from the response above, retrieve a transfer specification to enable Connect to download the package:

   Endpoint: `POST https://faspex5.example.com/aspera/faspex/api/v5/packages/{id}/transfer_spec/download`

```
curl -X POST 'https://faspex5.example.com/aspera/faspex/api/v5/packages/66/transfer_spec/
download?transfer_type=connect&type=received' \
-H 'Content-Type: application/json' \
-H "Authorization: Bearer
eyJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoyLCJleHAiOjE1OTYwMzY3NzV9.068OqoSZoTLYYMpEjYf5poK0hxVEYpk
taAYx7hBKP9I"
```

   Result:

```
{
  "cookie": "aspera.faspex20:d:26fa86a1-
d350-45e2-967a-95c75209a82e:amh3YW5AdXMuaWJtLmNvbQ:eyJfcGtnX3V1aWQiOiJjYmQ4ZGQyMS1kN2M1LTQ3ND
EtYjY4ZS1iOGRmMzMxMGQyNWMiLCJfcGtnX25hbWUiOiJGT3J3YXJkIiwiX2NyZWF0ZWRfdXRjIjoiMjAyMi0wMi0xNlQ
wMDo1ODozMC4wNDZaIn0:QQ",
  "target_rate_kbps": 10000,
  "lock_rate_policy": true,
  "lock_min_rate": true,
  "target_rate_cap_kbps": 20000,
  "tags": {
    "aspera": {
      "faspex": {
        "requestor": {
          "name": "jhwan@us.ibm.com",
          "first_name": "Admin",
          "last_name": "Admin",
          "email": "jhwan@us.ibm.com"
        },
        "metadata": {
          "_pkg_uuid": "cbd8dd21-d7c5-4741-b68e-b8df3310d25c",
          "_pkg_name": "A",
          "_created_utc": "2022-02-16T00:56:59.290Z"
        },
        "_pkg_id": 40,
        "downloader": {
          "name": "jhwan@us.ibm.com",
          "first_name": "Admin",
          "last_name": "Admin",
          "email": "jhwan@us.ibm.com"
        }
      },
      "xfer_id": "26fa86a1-d350-45e2-967a-95c75209a82e"
    }
  },
  "paths": [
    {
      "source": "/A - cbd8dd21-d7c5-4741-b68e-b8df3310d25c.aspera-package/PKG - A/"
    }
  ],
  "direction": "receive",
  "token": "ATM2_ACs1hDvAZ-qlHZ-psCGX1XtgoJbb8XN3Q_yK1_-
m8jVU6IAAGWnnwwb3QHvBpN2yYXvjMu_2MTA",
  "cipher": "aes-128",
  "rate_policy_allowed": "fixed",
  "rate_policy": "fair",
  "min_rate_kbps": 0,
  "remote_host": "ue-faspex-node1.fyre.ibm.com",
  "remote_user": "faspex",
  "ssh_port": 33001,
  "fasp_port": 33001,
  "authentication": "token",
  "multi_session": 0
}
```

4. Use the Connect SDK and the retrieved transfer specification to download a package.

```
// The connectDownload function takes a transferSpec and starts the download
function connectDownload(transferSpec) {
  // Initialize Connect
  let client = new AW4.Connect({
    minVersion: '3.10.0',
    dragDropEnabled: true
  });
  client.initSession();

  // Start Connect transfer
  client.startTransfer(transferSpec)
}
```

Connect downloads the package to your `Downloads` folder (default location).

Downloading a package with HTTP Gateway

Leverage the Faspex API and the HTTP Gateway SDK to download a package from your inbox.

To transfer with HTTP Gateway, you must have the URL of an HTTP Gateway server, and you must download the HTTP Gateway Javascript SDK.

Note:

- The code examples for the Faspex API are written with `curl`.
- The code examples for the HTTP Gateway SDK are written in JavaScript.

For a more detailed explanation of the endpoints used in this example, see the Faspex API reference by going to `https://`*`your_faspex_server/aspera/faspex/api`*. You do not need to log in to see the reference.

1. Retrieve a bearer token from the Faspex server.
2. List the packages you've received and choose the package you want to download.

   Endpoint: GET `https://faspex5.example.com/aspera/faspex/api/v5/inbox/packages`

   Example:

   ```
   curl 'https://faspex5.example.com/aspera/faspex/api/v5/inbox/packages/' \
   -H 'Content-Type: application/json' \
   -H "Authorization: Bearer
   eyJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoyLCJleHAiOjE1OTYwMzY3NzV9.068OqoSZoTLYYMpEjYf5poK0hxVEYpk
   taAYx7hBKP9I"
   ```

   Result:

   ```
   {
     "total_count":116,
   ```

```
  "offset":0,
  "limit":20,
  "packages": [
    {
      "id": 66,
      "title":" Example Package 1",
      "release_date":" 2020-09-23T17:14:26.000Z",
      "notified_on_download":"",
      "total_bytes": 0,
      "total_files": 0,
      "message":"",
      "upload_status":" Submitted",
      "created_time":" 2020-09-23T17:14:26.000Z",
      "upload_completion_time": null,
      "sender":" admin",
      "metadata":  {},
      "package_uuid":" c1568306-f92d-49bc-9bca-ab4904a96ef1",
      "recipients":  [
        {
          "id":" 2",
          "name":" admin"
        }
      ],
      "completed_download_count": 2,
      "active_download_count": 0,
      "completed_partial_download_count": 0
    },
    {
      "id": 67,
      "title": "Example Package 1",
      "release_date": "2020-09-23T17:14:02.000Z",
      "notified_on_download": "",
      "total_bytes": 0,
      "total_files": 0,
      "message": "",
      "upload_status": "Submitted",
      "created_time": "2020-09-23T17:14:02.000Z",
      "upload_completion_time": null,
      "sender": "admin",
      "metadata": {},
      "package_uuid": "8e85b3ea-7537-4832-a11c-767c1a1742c8",
        "recipients":
        [
          {
            "id":"2",
            "name":"admin"
          }
      ],
      "completed_download_count": 0,
      "active_download_count": 0,
      "completed_partial_download_count": 0
    },

    ...

  ]
};
```

3. Using the package ID from the response above, retrieve a transfer specification to enable HTTP Gateway to download the package:

   Endpoint: POST `https://faspex5.example.com/aspera/faspex/api/v5/packages/{id}/transfer_spec/download`

   ```
   curl -X POST 'https://faspex5.example.com/aspera/faspex/api/v5/packages/66/transfer_spec/
   download?transfer_type=http_gateway' \
   -H 'Content-Type: application/json' \
   -H "Authorization: Bearer
   eyJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoyLCJleHAiOjE1OTYwMzY3NzV9.068OqoSZoTLYYMpEjYf5poK0hxVEYpk
   taAYx7hBKP9I"
   ```

   Result:

   If successful, Faspex returns a response with the transfer specification:

   ```
   Response:
   201 - {
     "cookie":
   "aspera.faspex20:d:4b200abc-9eba-4117-9a55-1e621d10ae57:amh3YW5AdXMuaWJtLmNvbQ:eyJfcGtnX3V1aW
   ```

```
QiOiJjYmQ4ZGQyMS1kN2M1LTQ3NDEtYjY4ZS1iOGRmMzMxMGQyNWMiLCJfcGtnX25hbWUiOiJBIiwiX2NyZWF0ZWRfdXR
jIjoiMjAyMi0wMi0xNlQwMDo1Njo1OS4yOTBaIn0:QQ",
  "target_rate_kbps": 10000,
  "lock_rate_policy": true,
  "lock_min_rate": true,
  "target_rate_cap_kbps": 20000,
  "tags": {
    "aspera": {
      "faspex": {
        "requestor": {
          "name": "jhwan@us.ibm.com",
          "first_name": "Admin",
          "last_name": "Admin",
          "email": "jhwan@us.ibm.com"
        },
        "metadata": {
          "_pkg_uuid": "cbd8dd21-d7c5-4741-b68e-b8df3310d25c",
          "_pkg_name": "A",
          "_created_utc": "2022-02-16T00:56:59.290Z"
        },
        "_pkg_id": 39,
        "downloader": {
          "name": "jhwan@us.ibm.com",
          "first_name": "Admin",
          "last_name": "Admin",
          "email": "jhwan@us.ibm.com"
        }
      },
      "xfer_id": "4b200abc-9eba-4117-9a55-1e621d10ae57"
    }
  },
  "paths": [
    {
      "source": "/faspex_4_faspex_5_transfer_spec_workflows.pdf"
    }
  ],
  "direction": "receive",
  "source_root": "/A - cbd8dd21-d7c5-4741-b68e-b8df3310d25c.aspera-package/PKG - A",
  "token":
"ATM2_ACsdwXX4R03FLJcwBo9n1UgSIP2ipAhm1m4tnBZ6iHNp1MAAF57v5WnmMe5Jq3kkhYbl06_2MTA",
  "cipher": "aes-128",
  "rate_policy_allowed": "fixed",
  "rate_policy": "fair",
  "min_rate_kbps": 0,
  "remote_host": "ue-faspex-node1.fyre.ibm.com",
  "remote_user": "faspex",
  "ssh_port": 33001,
  "fasp_port": 33001,
  "download_name": "PKG - A",
  "zip_required": true,
  "authentication": "token",
  "multi_session": 0
}
```

4. Use the HTTP Gateway SDK and the retrieved transfer specification to download the package.

```
function download(gatewayHostname, transferSpec) {
  this.client = asperaHttpGateway
  const gateway_url = `https://${gatewayHostname}/aspera/http-gwy/v1/`
  this.client.initHttpGateway(gateway_url).then(response => {
    console.log('HTTP Gateway SDK started', response);
  }).catch(error => {
    console.warn('HTTP Gateway SDK did not start', error);
  })
  this.client.download(transferSpec).then(response => {
      console.log('Download started', response);
    }).catch(error => {
      console.log('Download could not start', error);
    });
}
```

Faspex 5.0 API reference

See the Faspex 5.0 API reference (OAS 3 format) at the IBM API Hub on the IBM Developer website.

Technical Support

Support Websites

For an overview of IBM Aspera Support services, visit https://www.ibm.com/products/aspera/support.

To view product announcements, webinars, and knowledge base articles, as well as access the Aspera Support Community Forum, sign into the IBM Aspera Support site at https://www.ibm.com/mysupport/ using your IBMid (not your company Aspera credentials), or set up a new account.

Technical Support

You may contact Aspera support using the IBM Aspera Support Guide: https://www.ibm.com/support/home/pages/support-guide/?product=3712142.

IBM®