# Contents

**Introduction**

**Standard Installation**

- Requirements.................................................................6
- Before Upgrading............................................................6
- Product Setup.................................................................7
- Configuring the Firewall......................................................9
- Securing your SSH Server..................................................10
- Testing a Locally Initiated Transfer.....................................16
- Updating the Product License.............................................17
- Uninstall...........................................................................18

**Transferring Files with the Application**

- Application Overview......................................................19
- Managing Connections.....................................................20
- Creating SSH Keys............................................................26
- Enabling a Transfer or HTTP Proxy.....................................29
- Transferring Files.............................................................33
- Advanced Transfer Mode...................................................36
- Configuring Transfer Notifications.....................................38
- Using Transfer Notifications...............................................46
- Reporting Checksums.........................................................48
- Configuring Multi-Session Transfers....................................51

**Managing Users**

- Setting Up Users.............................................................55
- Testing a User-Initiated Remote Transfer.................................57
- Setting Up Groups.............................................................58
- Setting Up Transfer Users (Terminal).....................................59
- Setting Up Transfer Groups (Terminal).....................................60
- Configuration Precedence...................................................62
- Setting Up a User's Public Key.............................................63

**General Configuration Reference**

- Document Root..............................................................64
- Advanced Symbolic Link Options (ascp)..................................65
- Server-Side Symbolic Link Handling.....................................66
- Authorization.....................................................................67
- Server-Side Encryption at Rest (EAR).....................................69
- Bandwidth.........................................................................72
- Network............................................................................76
- File Handling.....................................................................78
- aspera.conf - Authorization...............................................83
- aspera.conf - Transfer.......................................................85
- aspera.conf - File System...................................................93
Global Transfer Settings ........................................................................................................ 98
  Global Bandwidth Settings .................................................................................................. 98
  Setting Up Virtual Links ................................................................................................... 100
  Transfer Server Configuration .......................................................................................... 103

Configuring for Other Aspera Products .......................................................................... 106
  Configuring for Faspex .................................................................................................... 106
  Configuring for Shares .................................................................................................... 110
  Configuring for Aspera for SharePoint ........................................................................... 112

Managing the Node API ...................................................................................................... 118
  Overview: Aspera Node API ............................................................................................ 118
  Node API Setup ................................................................................................................ 118
  Setting up Node Users ..................................................................................................... 119
  Node Admin Tool .............................................................................................................. 119
  aspera.conf for Nodes ...................................................................................................... 121
  Redis DB Backup/Restore ................................................................................................. 124
  Setting up SSL for your Nodes ......................................................................................... 124

Pre- and Post-Processing (Prepost) .................................................................................. 128
  Setting Up Pre/Post ........................................................................................................... 128
  Pre/Post Variables ............................................................................................................ 129
  Pre/Post Examples ............................................................................................................ 131
  Setting Up Email Notification ......................................................................................... 132
  Email Notification Examples ........................................................................................... 134

Transferring from the Command Line ............................................................................ 137
  Ascp Command Reference ............................................................................................... 137
  Ascp General Examples .................................................................................................... 147
  Ascp File Manipulation Examples .................................................................................... 148
  Ascp Transfers to Cloud Storage ...................................................................................... 150
  Configuring the Aspera Watch Service ........................................................................... 151
  Transferring and Deleting Files with the Aspera Watch Service ..................................... 152
  Token Generation ............................................................................................................. 154
  Creating SSH Keys (Command Line) ............................................................................... 156
  Ascp FAQs ....................................................................................................................... 157

Configuring for the Cloud ................................................................................................. 159
  Configuring aspera.conf for S3 ....................................................................................... 159

Authentication and Authorization ...................................................................................... 161
  Access Key Authentication ............................................................................................... 161
  Adding a Node to Aspera Files ....................................................................................... 166
  Installing SSL Certificates .............................................................................................. 171
  Setting Up Token Authorization ....................................................................................... 174
  Configuring Token Authorization from the GUI ............................................................ 175
  Configuring Token Authorization With aspera.conf ......................................................... 176
Appendix................................................................................................................ 178
  Restarting Aspera Services...................................................................................... 178
  Optimizing Transfer Performance........................................................................... 178
  Log Files................................................................................................................... 179
  Accessing Shares from the GUI.............................................................................. 182
  Product Limitations................................................................................................. 183

Troubleshooting..................................................................................................... 184
  Using the Troubleshooter...................................................................................... 184
  Clients Can't Establish Connection...................................................................... 184

Technical Support.................................................................................................... 187

Legal Notice............................................................................................................ 188
IBM Aspera Enterprise Server is an universal file transfer server built upon Aspera's FASP transport. Enterprise Server offers the following features:

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FASP transport technology</td>
<td>File transfer protocol that dramatically speeds transfers over IP networks by eliminating the fundamental bottlenecks in conventional technologies. FASP features bandwidth control, resume, transfer encryption, content protection, and data integrity validation.</td>
</tr>
<tr>
<td>Transfer server</td>
<td>Allows an unlimited number of concurrent client transfers. Uses virtual links to manage aggregate bandwidth usage.</td>
</tr>
<tr>
<td>Enterprise Server application</td>
<td>A graphical file transfer application for initiating and managing transfers, and for configuring transfer users and server settings.</td>
</tr>
<tr>
<td>Pre- and Post-Processing (Prepost)</td>
<td>Executes customizable actions when transfer events - start and end of sessions and files - occur. <em>An email notification script is included.</em></td>
</tr>
<tr>
<td>ascp command</td>
<td>The command-line file transfer program.</td>
</tr>
</tbody>
</table>
Standard Installation

Install the IBM Aspera transfer product and set up your computer for FASP file transfers.

Requirements

Software and hardware requirements for optimal product performance

System requirements for IBM Aspera Enterprise Server:

- Product-specific Aspera license file.
- Linux kernel 2.4 or higher. Linux distributions/kernels released after the product release date may not be compatible.
- Libc version GLIB 2.5 or higher.
- SSH Server. Version 5.2 or higher is recommended.
- Screen resolution 1024 x 768 or higher for graphical user interface.

If you plan to set up and use the Node API, you must also meet the following requirements on each node machine:

- In order to use this application on a cloud platform and access the object-based cloud storage, you must obtain an on-demand license. Please contact Technical Support.
- Identify a directory you plan to use for sharing data. Later on (in Node API Setup), this directory will be used as the absolute path for the transfer user.
- Verify that the machine's hosts file has an entry for "127.0.0.1 localhost." For UNIX-based nodes, check /etc/hosts. For Windows nodes, check C:\Windows\system32\drivers\etc\hosts.
- For UNIX-based nodes, verify that SELINUX is disabled by running cat /etc/sysconfig/selinux. SELINUX can be "permissive" or "disabled," but not "enforced."

Before Upgrading

Steps to take before upgrading your IBM Aspera product.

The installer for IBM Aspera Enterprise Server automatically checks for a previous version of the product on your system. If a previous version is found, the installer automatically removes it and upgrades your computer to the newer version.

Although the installer performs your upgrade automatically, we highly recommend completing the tasks below before starting the installation/upgrade process. If you do not follow these steps, you risk installation errors or losing your former configuration settings. Skip any steps that do not apply to your specific product version.

Note: You cannot upgrade directly between different Aspera transfer products (such as from Point-to-Point to Desktop Client, or from Point-to-Point to Enterprise Server). To upgrade, you need to back up the configuration, uninstall the product, and perform a fresh install of the new version of the product. If you are upgrading your Enterprise Server to Connect Server, see the appendix in the Connect Server documentation located at http://asperasoft.com/en/documentation/4..

1. All Versions - Verify the version of your existing product

   Depending on your current product version, the upgrade preparation procedure may differ. In a terminal, execute this command:

   ```
   $ ascp -A
   ```

   This displays the product name and version number.

2. All versions - Stop all FASP transfer-related applications and connections.
Before upgrading the application, close the following applications and services:

- `ascp` connections
- SSH connections
- The Enterprise Server application

3. All versions - Back up the files

Depending on the version of your previous installation, back up the files in the specified locations:

<table>
<thead>
<tr>
<th>Version</th>
<th>Folder</th>
</tr>
</thead>
</table>
| 2.0.2 to 2.7+     | • /opt/aspera/etc/ (Server config, web config, user settings, license info)  
                  | • /opt/aspera/var/ (Pre- and Post-Processing scripts, Connect Server)     |
| 2.0.1 and earlier | • /var/opt/aspera/etc/ (Server config, web config, user settings, license info)  
                  | • /usr/local/aspera/var/ (Pre- and Post-Processing scripts, Connect Server) |

In particular, be aware that upgrading backs up and deletes the existing `sshd_config` file before installing the new, default `sshd_config`. Therefore, you may want to migrate any changes from the backup file (`sshd_config.old`) to the newly installed `sshd_config`.

### Product Setup

Install the IBM Aspera transfer product and set up your computer for FASP file transfers.

#### Important:
If this is a product upgrade, ensure that you have reviewed all prerequisites detailed under "Before Upgrading."

1. Download the IBM Aspera product installer.

   Download the installer from the link below. Use the credentials provided to your organization by Aspera to access:


   If you need help determining your firm's access credentials, contact Technical Support on page 187.

2. For product upgrades, ensure you have prepared your machine to upgrade to a newer version.

   Although the installer for Aspera Enterprise Server performs your upgrade automatically, Aspera highly recommends completing the tasks identified in the topic Before Upgrading. If you do not follow these steps, you risk installation errors or losing your former configuration settings.

3. Run the installer

   When downloaded, run the installer using the following commands and with the proper administrative permissions. Replace the file name accordingly.

<table>
<thead>
<tr>
<th>OS</th>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>RedHat</td>
<td><code>$ rpm -Uvh aspera-entsrv-version.rpm</code></td>
</tr>
<tr>
<td>Debian</td>
<td><code>$ dpkg -i aspera-entsrv-version.deb</code></td>
</tr>
</tbody>
</table>

4. Install the license.

   The following table describes two ways to install the license: through the GUI or via command line.
<table>
<thead>
<tr>
<th>Method</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>GUI</td>
<td>To install the license via the GUI, execute <code>asperascp</code> in a Terminal (as a root user) to launch the application. If this is a fresh install, an Enter License window appears. You may click the <strong>Import License File</strong> and select the license file, or <strong>Paste License Text</strong> to copy-and-paste the license file's content. When finished, the license information will appear in the window. Verify that it is correct and click <strong>Close</strong>.</td>
</tr>
</tbody>
</table>
| Terminal   | To install the license from the command line, create the following file and paste your license key string into it:  
  ```bash
  /opt/aspera/etc/aspera-license
  ```  
  When finished, save and close the file. Use this command to verify the license info:  
  ```bash
  $ ascp -A
  ``` |

If you are updating your product license after the installation, see *Updating the Product License* on page 17.

5. Review or update OpenSSH authentication methods.

Open your SSH Server configuration file with a text editor:

```bash
/etc/ssh/sshd_config
```

To allow public key authentication, set `PubkeyAuthentication yes`. To allow password authentication, set `PasswordAuthentication yes`, for example:

```bash
...  
PubkeyAuthentication yes  
PasswordAuthentication yes  
...```
When modified, run the following to reload SSH:

**RedHat**

$ sudo service sshd restart

**Debian**

$ sudo /etc/init.d/ssh restart

To further review your SSH Server's configuration to strengthen security, see *Securing your SSH Server* on page 10.

6. Convert the old `aspera.conf` file manually (necessary only when upgrading from product version 2.2 or earlier)

For product versions 2.5+, the docroot settings have been moved to the Aspera configuration file, `aspera.conf`. When upgrading from product version 2.2 or earlier, the installer converts your old configuration files to the new format, using a "strict" method. If the old `aspera.conf` file has errors or unrecognized directives, the conversion will fail. To review the errors, run a strict conversion manually. Change `aspera.conf`'s path if it is not in the default location.

```bash
$ cd /opt/aspera/etc
$ sudo asconfigurator -T -F convert_conf_V1_data ./aspera.conf
```

If an error occurs during the conversion, use the relaxed conversion method:

```bash
$ cd /opt/aspera/etc
$ sudo asconfigurator -F convert_conf_V1_data ./aspera.conf
```

7. (For upgrades) Check `aspera.conf` for errors

When upgrading your Aspera product to a newer version, it is recommended that you check the `aspera.conf` configuration file for errors. Run the following command in a terminal window to validate `aspera.conf`:

```bash
$ /opt/aspera/bin/asuserdata -v
```

## Configuring the Firewall

Required firewall settings.

Your Aspera transfer product requires access through the ports listed in the table below. If you cannot establish the connection, review your local corporate firewall settings and remove the port restrictions accordingly.

<table>
<thead>
<tr>
<th>Product</th>
<th>Firewall Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enterprise Server</td>
<td>An Aspera server runs one SSH server on a configurable TCP port (22 by default).</td>
</tr>
<tr>
<td></td>
<td>Important: Aspera strongly recommends running the SSH server on a non-default port to ensure that your server remains secure from SSH port scan attacks. For instructions on how to change your SSH port, see <em>Securing your SSH Server</em> on page 10.</td>
</tr>
<tr>
<td></td>
<td>Your firewall should be configured as follows:</td>
</tr>
<tr>
<td></td>
<td>• Allow inbound connections for SSH, which is on TCP/22 by default, or on another non-default, configurable TCP port. To ensure that your server is secure, Aspera recommends allowing inbound SSH connections on TCP/33001, and disallowing inbound connections on TCP/22. If you have a legacy customer base utilizing TCP/22, then you can allow inbound connections on both ports. Please refer to the topic <em>Securing your SSH Server</em> on page 10 for details.</td>
</tr>
</tbody>
</table>
### Standard Installation

<table>
<thead>
<tr>
<th>Product</th>
<th>Firewall Configuration</th>
</tr>
</thead>
</table>
| **Firewall Configuration** | • Allow inbound connections for FASP transfers, which use UDP/33001 by default, although the server may also choose to run FASP transfers on another port.  
  • If you have a local firewall on your server (like **iptables**), verify that it is not blocking your SSH and FASP transfer ports (such as TCP/UDP 33001).  
  The firewall on the server side must allow the open TCP port to reach the Aspera server. Note that no servers are listening on UDP ports. When a transfer is initiated by an Aspera client, the client opens an SSH session to the SSH server on the designated TCP port and negotiates the UDP port over which the data transfer will occur. |
| Client | Typically, consumer and business firewalls allow direct outbound connections from client computers on TCP and UDP. In this case, no configuration is required for Aspera transfers. In the special case of firewalls disallowing direct outbound connections, typically using proxy servers for Web browsing, the following configuration applies:  
  • Allow outbound connections from the Aspera client on the TCP port (TCP/33001 by default, when connecting to a Windows server, or on another non-default port for other server operating systems).  
  • Allow outbound connections from the Aspera client on the FASP UDP port (33001, by default).  
  • If you have a local firewall on your server (such as **iptables**), verify that it is not blocking your SSH and FASP transfer ports (such as TCP/UDP 33001). |

**Important:** If you have a local firewall on your server (Windows firewall, Linux iptables, or Mac ipfw), you will need to allow the Vlink UDP port (55001, by default) for multicast traffic. For additional information on setting up Vlinks, see [Setting Up Virtual Links](#) on page 100.

### Securing your SSH Server

Secure your SSH server to prevent security risks.

**Introduction**

Keeping your data secure is critically important. Aspera strongly recommends taking additional steps to set up and configure your SSH server so that it's protected against common attacks. Most automated robots try to log into your SSH server on Port 22 as root with various brute force and dictionary combinations in order to gain access to your data. Furthermore, automated robots can put enormous loads on your server as they perform thousands of retries to break into your system. This topic addresses steps to secure your SSH server against potential threats, including changing the default port for SSH connections from TCP/22 to TCP/33001.

**Why Change to TCP/33001?**

It is well known that SSH servers listen for incoming connections on TCP Port 22. As such, Port 22 is subject to countless, unauthorized login attempts by hackers who are attempting to access unsecured servers. A highly effective deterrent is to simply turn off Port 22 and run the service on a seemingly random port above 1024 (and up to 65535). To standardize the port for use in Aspera transfers, we recommend using TCP/33001.

**Note:** Remote Aspera application connections attempt to establish an SSH connection using the default port 33001. However, if the connection fails, the application attempts the connection using port 22.

The following explains how to change the SSH port to 33001 and take additional steps to secure your SSH server. The steps all require root access privileges.

1. Locate and open your system's SSH configuration file.
The SSH configuration file can be found in the following location:

/etc/ssh/sshd_config

2. Add a new SSH port.

Note: Before changing the default port for SSH connections, verify with your network administrators that TCP/33001 is open.

The OpenSSH suite included in the installer uses TCP/22 as the default port for SSH connections. Aspera recommends opening TCP/33001 and disabling TCP/22 to prevent security breaches of your SSH server.

To enable TCP/33001 while your organization is migrating from TCP/22, open port 33001 from your sshd_config file (where SSHD is listening on both ports). As demonstrated by this exercise, SSHD is capable of listening on multiple ports.

... 
Port 22
Port 33001
...

Once your client users have been notified of the port change (from TCP/22 to TCP/33001), you can disable port 22 in your sshd_config file. To disable TCP/22 and use only TCP/33001, comment out "Port 22" in your sshd_config file.

... 
#Port 22
Port 33001
...

Note: Aspera recognizes that disabling the default SSH connection port (TCP/22) may affect your client users. When you change the port, ensure that you advise your users on configuring the new port number. Basic instructions for specifying the SSH port for FASP file transfers are shown below. To change the SSH port for Desktop Client, click Connections on the main window, and select the entry for your computer. Under the Connection tab, click Show Advanced Settings and enter the SSH port number in the SSH Port (TCP) field.
To make an impromptu connection to TCP/33001 during an ascp session, specify the SSH port (33001) with the –P (capital P) flag. Note that this command does not alter ascp or your SSH server’s configuration.

```bash
$ ascp -P 33001 ...
```

3. Disable non-admin SSH tunneling

**Note:** The instructions below assume that OpenSSH 4.4 or newer is installed on your system. For OpenSSH 4.4 and newer versions, the **Match** directive allows some configuration options to be selectively overridden if specific criteria (based on user, group, hostname and/or address) are met. If you are running an OpenSSH version older than 4.4, the **Match** directive is not available; Aspera recommends updating to the latest version.

In OpenSSH versions 4.4 and newer, disable SSH tunneling to avoid potential attacks; thereby only allowing tunneling from root users. To disable non-admin SSH tunneling, open your SSH Server configuration file, `sshd_config`, with a text editor.

Add the following lines to the end of the file (or modify them if they already exist):

```plaintext
... AllowTcpForwarding no
Match Group root
AllowTcpForwarding yes
```

Depending on your `sshd_config` file, you may have additional instances of `AllowTCPForwarding` that are set to the default `Yes`. Review your `sshd_config` file for other instances and disable as appropriate.

Note that disabling TCP forwarding does not improve security unless users are also denied shell access, as they can always install their own forwarders. Review your user and file permissions, and see the instructions below on modifying shell access.

4. Update authentication methods

Public key authentication can prevent brute-force SSH attacks if all password-based authentication methods are disabled. For this reason, Aspera recommends disabling password authentication in the `sshd_config` file and enabling private/public key authentication. To do so, add or uncomment `PubkeyAuthentication yes` and comment out `PasswordAuthentication yes`.

```plaintext
... PubkeyAuthentication yes
#PasswordAuthentication yes
PasswordAuthentication no
... 
```

**Note:** If you choose leave password authentication enabled, be sure to advise account creators to use strong passwords. Be sure also to set `PermitEmptyPasswords` to “no”.

```plaintext
PermitEmptyPasswords no
```

5. Disable Root Login

OpenSSH defaults to allowing root logins; however disabling root access helps you to maintain a more secure server. Aspera recommends commenting out `PermitRootLogin yes` in the `sshd_config` file and adding `PermitRootLogin No`.

```plaintext
... #PermitRootLogin yes
PermitRootLogin no
... 
```

Administrators can then utilize the `su` command if root privileges are needed.
6. Restart the SSH server to apply new settings

When you have finished updating your SSH server configuration, you must restart or reload the SSH service to apply your new settings. Note that restarting or reloading SSH does not impact currently connected users.

To restart or reload your SSH server, run the following commands:

<table>
<thead>
<tr>
<th>OS Version</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>RedHat (restart)</td>
<td>$ sudo service sshd restart</td>
</tr>
<tr>
<td>RedHat (reload)</td>
<td>$ sudo service sshd reload</td>
</tr>
<tr>
<td>Debian (restart)</td>
<td>$ sudo /etc/init.d/ssh restart</td>
</tr>
<tr>
<td>Debian (reload)</td>
<td>$ sudo /etc/init.d/ssh reload</td>
</tr>
</tbody>
</table>

7. Restrict user access

Restricting user access is a critical component of securing your server. By default, all user accounts are allowed to browse and read all files on the server. To limit a user's access to a portion of the system, set the account's shell to the Aspera secured shell (aspershell) and create a document root (docroot) for that user. The aspershell permits only the following operations:

- Run Aspera uploads and downloads to or from this computer.
- Establish connections in the application and browse, create, delete, rename, or list contents.

The following instructions describe how to change a user account so that it uses the aspershell. Keep in mind that this is an example, and there may be other ways to do so for your system. Open the following file with a text editor:

```
/etc/passwd
```

Add or replace the user's shell with `/bin/aspershell`. For example, to apply aspershell to the user aspera_user_1, use the following settings in the passwd file:

```
...aspera_user_1:x:501:501:...:/home/aspera_user_1:/bin/aspershell
...
```

You can also restrict a user's file access by setting a document root (docroot). You can set a user's docroot by editing the aspera.conf file (`/opt/aspera/etc/aspera.conf`). The following template displays access options:

```
<file_system>
  <access>
    <paths>
      <path>
        <absolute>/sandbox/aspera_user_1</absolute>       <!-- Absolute Path -->
      </path>
      <read Allowed>true</read_allowed>                  <!-- Read Allowed -->
      <write Allowed>true</write_allowed>                <!-- Write Allowed -->
      <dir_allowed>true</dir_allowed>                    <!-- Browse Allowed -->
    </paths>
  </access>
...<file_system>
```
Once you’ve set the user’s shell and docroot, you can further restrict access by disabling read, write, and/or browse using `<path>` settings in `aspera.conf`, as shown in the example above.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Absolute Path</td>
<td>The area of the file system (path) that is accessible to the Aspera user. The default empty value gives a user access to the entire file system.</td>
<td>Path or blank</td>
</tr>
<tr>
<td>Read Allowed</td>
<td>Setting this to <code>true</code> allows users to transfer from the designated area of the file system as specified by the Absolute Path value.</td>
<td><code>true</code> <code>false</code></td>
</tr>
<tr>
<td>Write Allowed</td>
<td>Setting this to <code>true</code> allows users to transfer to the designated area of the file system as specified by the Absolute Path value.</td>
<td><code>true</code> <code>false</code></td>
</tr>
<tr>
<td>Browse Allowed</td>
<td>Setting this to <code>true</code> allows users to browse the directory.</td>
<td><code>true</code> <code>false</code></td>
</tr>
</tbody>
</table>

8. Run the `asp-check` tool to check for potential user-security issues

The `asp-check` tool performs the following secure checks:

- Searches for full-access users and reports how many exist on the system. Note that the existence of full-access users does not necessarily indicate that your system is vulnerable; however, it is being brought to the attention of the system administrator to ensure that the existence of full-access users is intentional.

- Searches for restricted users and potential misconfigurations, including incorrect login shell (one that is not restricted via `aspshell`); SSH tunnel access (which can be used to work around the restricted shell); and docroot settings that allow the users to access the home directory.

> Note: A docroot setting that allows access to the home directory does not necessarily indicate that your system is vulnerable; however, a user with this docroot can download or upload keys in `.ssh`, as well as upload `.login` scripts. These capabilities may be used to circumvent the intended, restricted-nature of the user. Aspera highly recommends setting the docroot under the user's home folder (such as `/home/jane/data`) or in an alternate location (for example, `/data`).

To run the `asp-check` tool, run the following on the command line:

```
$ sudo /opt/aspera/bin/asp-check.sh
```

Search results are displayed as in the following example. If potential issues are identified, review your users' settings before proceeding.

Users with full access: 22 (not considered insecure)
Restricted users: 0
Insecure users: 0
  - no restricted shell (aspshell): 0
  - docroot above home directory: 0
  - ssh tunneling enabled: 0

9. Review your logs periodically for attacks.

Aspera recommends reviewing your SSH log periodically for signs of a potential attack. Locate and open your syslog—for example, `/var/log/auth.log` or `/var/log/secure`. Depending on your system configuration, syslog's path and file name may vary.

Look for invalid users in the log, especially a series of login attempts with common user names from the same address, usually in alphabetical order. For example:

```
...
If you identify attacks, do the following:

- Double-check the SSH security settings in this topic.
- Report attackers to your ISP’s email address for abuse reports (often abuse@your.isp.com).

10. Set up transfer server authentication

For transfers mediated by a web application (such as Faspex, Shares, or Console), the client browser sets up the context for the transfer using an HTTPS connection to the server, and then delegates the transfer to the Aspera FASP engine. The FASP engine then connects to the transfer server. In so doing, it needs to ensure the server’s authenticity in order to protect the client against server impersonation and man-in-the-middle (MITM) attacks.

To verify the authenticity of the transfer server, the web app passes the client a trusted SSH host key fingerprint of the transfer server. When connecting to the transfer server, the client confirms the server’s authenticity by comparing the server’s fingerprint with the trusted fingerprint.

To configure transfer server authentication, open the transfer server’s aspera.conf file:

```
/opt/aspera/etc/aspera.conf
```

Locate the `<server>` section, and add the `<ssh_host_key_fingerprint>` or the `<ssh_host_key_path>` option.

- `<ssh_host_key_fingerprint>`
  
  Use this option to specify the fingerprint itself:

  ```
  <ssh_host_key_fingerprint>fingerprint</ssh_host_key_fingerprint>
  ```

  To retrieve the SSH fingerprint, locate the transfers server’s public or private key, and run the following command on a Linux, Mac, Isilon, or other UNIX computer:

  ```
  # cd /etc/ssh
  # cat ssh_host_rsa_key.pub | cut -d' ' -f2 | base64 -d | sha1sum | cut -d'sh-"' -f1
  ```

  The following is an example SSH fingerprint:


  By convention, Aspera uses a hex string without the colons (`:"`). For example:

  `435143a1b5fc8bb70a3a9b10f6673a8`

  The aspera.conf setting for this key would then be as follows:

  ```
  <ssh_host_key_fingerprint>435143a1b5fc8bb70a3a9b10f6673a8</ssh_host_key_fingerprint>
  ```

- `<ssh_host_key_path>`
Use the key-path option to specify the transfer server's public or private key file and its location. The fingerprint is extracted automatically.

```plaintext
<ssh_host_key_path>key_file</ssh_host_key_path>
```

On most Linux systems, the SSH keys are located in `/etc/ssh`. On OSX, the SSH keys are located in `/etc`. The following example uses a Linux server's public RSA key:

```plaintext
<ssh_host_key_path>/etc/ssh/ssh_host_rsa_key.pub</ssh_host_key_path>
```

After modifying `aspera.conf`, be sure to restart the node service by running `asperanoded`:

```plaintext
# /etc/init.d/asperanoded restart
```

## Testing a Locally Initiated Transfer

Test client functionality by transferring to and from the Aspera Demo Server.

To make sure the software is working properly, follow these steps to test download and upload transfers between your system and the Aspera Demo Server:

1. **Download test files from the Demo Server.**

   The first test is to download a test file from the Demo Server. The transfer command is based on the following settings:

<table>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Demo Server address</td>
<td>demo.asperasoft.com</td>
</tr>
<tr>
<td>Login account</td>
<td>aspera</td>
</tr>
<tr>
<td>password</td>
<td>demoaspera</td>
</tr>
<tr>
<td>Test file</td>
<td>/aspera-test-dir-large/100MB</td>
</tr>
<tr>
<td>Download location</td>
<td>/tmp/</td>
</tr>
<tr>
<td>Transfer settings</td>
<td>Fair transfer policy, target rate 10M, minimum</td>
</tr>
</tbody>
</table>
<pre><code>                   | rate 1M, encryption disabled.                  |
</code></pre>

Use the following command to download, press `y` to accept the server's key, and enter the password `demoaspera` when prompted:

```plaintext
$ ascp -QT -l 10M -m 1M aspera@demo.asperasoft.com:aspera-test-dir-large/100MB /tmp/
```

You should see the following session messages. The description from left to right is explained below:

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>100 MB</td>
<td>The name of the file that is being transferred.</td>
</tr>
<tr>
<td>23%</td>
<td>The percentage completed.</td>
</tr>
<tr>
<td>23 MB</td>
<td>The amount transferred.</td>
</tr>
</tbody>
</table>
### Standard Installation

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>509 Kb/s</td>
<td>The current transfer rate.</td>
</tr>
<tr>
<td>11:59 ETA</td>
<td>The estimated time the transfer will complete.</td>
</tr>
</tbody>
</table>

2. **Upload test files to the Demo Server.**

When the file is downloaded, try uploading the same file back to the Demo Server. Use the command to upload the file (100MB) to the Demo Server's /Upload directory. Enter the password demoaspera when prompted:

```
$ ascp -QT -l 10M -m 1M /tmp/100MB aspera@demo.asperasoft.com:Upload/
```

---

### Updating the Product License

**Update your product license.**

You can update your product license either from the application GUI or from the command line.

To update the license from the GUI, launch asperascp (as a root user) from the command line and click **Tools > License** to open the License window.

- You can either click the **Import License File** and select the license file, or click **Paste License Text** to paste the copied content of the license file. When finished, the license information will appear in the window. Verify that it is correct and click **Close**.
To update the license from the command line, open the following file with write permission, and replace the existing license key string with the new one:

```
/opt/aspera/etc/aspera-license
```

When finished, save and close the file. Enter the following command to verify the new license info:

```
$ ascp -A
```

If you are using the Node API, you must reload the `asperanoded` service.

```
$ /opt/aspera/bin/asnodeadmin --reload
```

## Uninstall

How to uninstall the Aspera product from your computer.

To uninstall the product, run the following commands. For RedHat and Debian, replace the package name with the name returned by the first command:

<table>
<thead>
<tr>
<th>Platform</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>RedHat</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$ rpm -qa</td>
</tr>
<tr>
<td></td>
<td>$ rpm -e package_name</td>
</tr>
<tr>
<td>Debian</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$ dpkg -l &quot;aspera*&quot;</td>
</tr>
<tr>
<td></td>
<td>$ dpkg -P package_name</td>
</tr>
</tbody>
</table>
Transferring Files with the Application

Using the desktop application to transfer files.

Application Overview

Desktop application overview.

To launch the desktop application, execute the following command in a terminal shell. To perform administrator tasks (such as server configuration, license updates, or configure email notification templates), run it with root permissions.

```
# asperascp
```

The application GUI appears as follows:

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>The transfer mode. Reveal the local/remote file browsers.</td>
</tr>
<tr>
<td>B</td>
<td>The transfer details mode. Show the selected transfer session's details and the transfer control options.</td>
</tr>
<tr>
<td>C</td>
<td>Bring up the Connection Manager window to manage the remote endpoints.</td>
</tr>
<tr>
<td>D</td>
<td>Bring up the Server Configuration window to configure the computer's FASP transfer settings.</td>
</tr>
<tr>
<td>E</td>
<td>Set the local computer's default transfer settings such as the FASP global bandwidth and the number of simultaneous transfers in the queue, and the SMTP server's information for transfer notifications.</td>
</tr>
<tr>
<td>F</td>
<td>Browse the local file system to find files to transfer.</td>
</tr>
<tr>
<td>G</td>
<td>When not connected, this panel shows connections that lists the saved connections. When connected, it becomes the remote file browser.</td>
</tr>
<tr>
<td>H</td>
<td>Display previous, ongoing, and queued transfers. Manage the priority.</td>
</tr>
</tbody>
</table>
Managing Connections

Add and manage remote FASP servers.

To connect to a remote computer or to a server in the cloud, you need to add it to the Connection Manager before establishing the connection. If you are planning to perform transfers with an S3 bucket, you must meet the following prerequisites:

- You (username) have permissions to access the S3 bucket.
- You know your username's S3 Access ID and Secret Key.
- To transfer files from or to an S3 storage device using an S3 direct connection, you cannot have a docroot. A local docroot will cause a transfer to fail. Be sure to confirm your docroot settings before attempting a transfer.

Start the application: asperascp. In the upper-right corner, click Connections to open the Connection Manager.
In the Connection Manager, click + to create a new connection. You can also use ‹ to duplicate a selected connection (i.e. copy all information into a new profile) and — to delete a connection profile.

To name or rename a connection, click the orange connection profile name that appears at the top of the screen. The Rename Connection dialog appears. You can also launch the Rename Connection dialog by clicking once on an already selected connection name in the left panel of the Connection Manager. When you have entered the new name, save it by clicking OK (once in the Rename Connection dialog and again in the Connection Manager).

The Connection Manager includes the following configuration tabs:
<table>
<thead>
<tr>
<th>Tab</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>The basic host information, such as the address, login credentials, and connection ports.</td>
</tr>
<tr>
<td>Transfer</td>
<td>The transfer session-related options, such as the transfer speed and retry rules.</td>
</tr>
<tr>
<td>Tracking</td>
<td>Options for tracking the transfer session, including the confirmation receipt and the email notifications.</td>
</tr>
<tr>
<td>Filters</td>
<td>Create filters to skip files that match certain patterns.</td>
</tr>
<tr>
<td>Security</td>
<td>Enable the transfer encryption and the content protection.</td>
</tr>
<tr>
<td>File Handling</td>
<td>Set up resume rule, preserve transferred file attributes, and remove source files.</td>
</tr>
</tbody>
</table>

The following tables detail all options in these tabs:

**Connection**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td>Required The server's address, such as 192.168.1.10 or companyname.com.</td>
</tr>
<tr>
<td>User</td>
<td>The login user for the server.</td>
</tr>
<tr>
<td>Authentication</td>
<td>Choose either password or public key for authentication. To use the key-based authentication, see Creating SSH Keys on page 26.</td>
</tr>
<tr>
<td>Storage Type</td>
<td>Use this drop-down menu to configure storage in the cloud. Note that the default option is local storage.</td>
</tr>
</tbody>
</table>

Storage types include the following:

- **Akamai NetStorage**
- **Amazon S3**: Once selected, you will be required to input your Access Id / Secret Access Key and identify a bucket. Note that the local machine must be reasonably time-synchronized in order to communicate with the Amazon servers. You can also select the Advanced button to modify the following settings:
  - **Host**: Amazon S3 hostname (default: s3.amazonaws.com).
  - **Port**: Default is port 443.
  - **HTTPS connection for file browsing**: Enable for secure browsing.
  - **Server-side file encryption**: Enable for AES256 encryption.
  - **Reduced redundancy storage class**: Assign objects to a to the "reduced redundancy" storage class (durability of 99.99%).
- **Google Storage**
- **Limelight**
- **OpenStack Swift**
- **Windows Azure**
- **Windows Azure SAS**
### Transferring Files with the Application

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Note:</strong></td>
<td>You can only choose special storage if you have <em>full access</em> to that storage on the cloud-based machine.</td>
</tr>
</tbody>
</table>

| Target Directory | The default directory when connecting to this computer. When leaving it blank, browsing the remote host brings up either the user account's document root (docroot), or the last-visited folder; when specifying a path, connecting to the host always brings up the exact directory. The default directory is shown in the **Connections** panel. |
| Share this connection ... | Check this box to share this connection with other users on your computer. When a connection is authenticated through Public Key, the SSH keys used by this connection should be shared as well. Refer to *Creating SSH Keys* on page 26. |
| Advanced Settings > SSH Port (TCP) | The TCP network port. Default: 33001. Note that if connecting on 33001 fails, the application attempts to establish a connection on port 22. If the connection on 22 succeeds, the setting is updated to 22. |
| Advanced Settings > fasp Port (UDP) | The UDP network port: Default: 33001. |
| Advanced Settings > Connection Timeout | Time out the connection attempt after the selected time. |
| Test Connection | Click this button to test the connection to the remote server with the settings you configured. An alert box opens and reports whether the connection is successful. |

### Transfer

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Transfer Name</strong></td>
<td>Choose between the following option: <em>Automatically generate</em> allows the user interface to generate the transfer name; <em>Automatically generate and add prefix</em> uses auto-generated name with prefix; <em>Specify</em> uses the user-specified name.</td>
</tr>
<tr>
<td><strong>Policy</strong></td>
<td>Select the FASP transfer policy.</td>
</tr>
<tr>
<td>• <strong>fixed</strong></td>
<td>Attempts to transfer at the specified target rate, regardless of the actual network capacity. This policy transfers at a constant rate and finishes in a guaranteed time. This policy typically occupies most of the network's bandwidth, and is not recommended in most file transfer scenarios. In fixed mode, a maximum (target) rate value is required.</td>
</tr>
<tr>
<td>• <strong>high</strong></td>
<td>Monitors the network and adjusts the transfer rate to fully utilize the available bandwidth up to the maximum rate. When congestion occurs, it transfers at a rate twice of a session with fair policy. In this mode, both the maximum (target) and the minimum transfer rates are required.</td>
</tr>
<tr>
<td>• <strong>fair</strong></td>
<td>Monitors the network and adjusts the transfer rate to fully utilize the available bandwidth up to the maximum rate. When other types of traffic build up and congestion occurs, it shares bandwidth fairly by transferring at an even rate. In this mode, both the maximum (target) and the minimum transfer rates are required.</td>
</tr>
<tr>
<td>• <strong>low</strong></td>
<td>Similar to fair mode, the low policy uses the available bandwidth up to the maximum rate, but is much less aggressive when sharing bandwidth with other network traffic. When congestion builds up, the transfer rate is reduced to the minimum rate until other traffic retreats.</td>
</tr>
<tr>
<td><strong>Speed</strong></td>
<td>Check this option to specify the transfer rate. The target rate is constrained by the global bandwidth in the <strong>Preferences</strong> window. Refer to <em>Global Bandwidth Settings</em> on page 98.</td>
</tr>
</tbody>
</table>
### Transferring Files with the Application

<table>
<thead>
<tr>
<th><strong>Option</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
</table>
| **Retry**  | Check this option to automatically retry the transfer after a recoverable failure. When checked, set the amount of time the transfer should be retried in seconds, minutes or hours. You may set the initial and maximum retry intervals by clicking the More Options... button.  

- **Initial interval**: The first retry waits for the initial interval. Input in seconds, minutes or hours.  
- **Maximum interval**: After the initial interval, the next interval doubles until the maximum interval is met, and then stops retrying after the retry time is reached. Input in seconds, minutes or hours.  

*Example 1:*  

<table>
<thead>
<tr>
<th>Retry at (seconds):</th>
<th>10s 30s 70s 130s 180s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interval progression (seconds):</td>
<td>10s 20s 40s 60s 60s 50s</td>
</tr>
</tbody>
</table>

*Example 2:*  

<table>
<thead>
<tr>
<th>Retry at (seconds):</th>
<th>30s 90s 210s 330s 450s 570s 600s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interval progression (seconds):</td>
<td>30s 60s 120s 120s 120s 120s 30s</td>
</tr>
</tbody>
</table>

| **Show Advanced Settings** | Click the Show Advanced Settings button to reveal the following options:  

- **Specify FASP datagram size (MTU)**: By default, the detected path MTU is used. Once you enable this checkbox, you can specify a value between 296 and 10000 bytes.  
- **Disable calculation of source files size before transferring**: By enabling this checkbox, you can turn off the job size calculation on the client-side (if allowed by the server). |

### Tracking

<table>
<thead>
<tr>
<th><strong>Option</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Generate delivery confirmation receipt</td>
<td>Check the option to create the delivery receipt file in the specified location.</td>
</tr>
<tr>
<td>Send email notifications</td>
<td>Send out email notifications based on specified events (start, complete, and error). Refer to Using Transfer Notifications on page 46 for more information.</td>
</tr>
</tbody>
</table>

### Filters

Click Add and enter the pattern to exclude files or directories with the specified pattern in the transfer. The exclude pattern is compared with the whole path, not just the file name or directory name. Two special symbols can be used in the setting of patterns:

<table>
<thead>
<tr>
<th><strong>Symbol</strong></th>
<th><strong>Name</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>Asterisk</td>
<td>Represents zero to many characters in a string, for example *.tmp matches .tmp and abcde.tmp.</td>
</tr>
<tr>
<td>?</td>
<td>Question mark</td>
<td>Represents one character, for example t?p matches tmp but not temp.</td>
</tr>
</tbody>
</table>

Examples:
### Filter Pattern

<table>
<thead>
<tr>
<th>Filter Pattern</th>
<th>Matched files</th>
</tr>
</thead>
<tbody>
<tr>
<td>*dirName</td>
<td>path/to/dirName, another/dirName</td>
</tr>
<tr>
<td>*1</td>
<td>a/b/file1, /anotherfile1</td>
</tr>
<tr>
<td>*filename</td>
<td>path/to/filename, /filename</td>
</tr>
<tr>
<td>path?/file?</td>
<td>path1/fileA, pathN/file5</td>
</tr>
</tbody>
</table>

### Security

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encryption</td>
<td>When checked, FASP encrypts files while transferring. Encryption may decrease performance, especially at higher transfer speeds and with slower computers.</td>
</tr>
<tr>
<td>Content Protection</td>
<td>Two options: Encrypt uploaded files with a password encrypts the uploaded files with the specified password. The protected file has the extension .aspera-env appended to the file name; Decrypt password-protected files downloaded prompts for the decryption password when downloading encrypted files.</td>
</tr>
</tbody>
</table>

### File Handling

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resume</td>
<td>Check Resume incomplete files to enable the resume feature. Under When checking files for differences, choose from the following options:</td>
</tr>
<tr>
<td></td>
<td>• Compare file attributes only checks whether the existing file is the same size.</td>
</tr>
<tr>
<td></td>
<td>• Compare sparse file checksums performs a sparse checksum on the existing file.</td>
</tr>
<tr>
<td></td>
<td>• Compare full file checksums perform a full checksum on the existing file.</td>
</tr>
<tr>
<td></td>
<td>Under When a complete file already exists at the destination, select an overwrite rule when the same file exists at the destination.</td>
</tr>
<tr>
<td>File Attributes</td>
<td>• Enable the Preserve Access Time checkbox to set the access time of the destination file to the same value as that of the source file.</td>
</tr>
<tr>
<td></td>
<td>• Enable the Preserve Modification Time checkbox to set the modification time of the destination file to the same value as that of the source file.</td>
</tr>
<tr>
<td></td>
<td>• Enable the Preserve Source Access Time checkbox to keep the access time of the source file the same as its value before the transfer.</td>
</tr>
<tr>
<td></td>
<td>Note: Access, modification, and source access times cannot be preserved for node and Shares connections that are using cloud storage.</td>
</tr>
<tr>
<td>Source Deletion</td>
<td>Check Automatically delete source files after transfer to delete successfully transferred files from the source. Check Delete empty source subdirectories to also remove empty folders (except a folder specified as the source to transfer).</td>
</tr>
<tr>
<td>Source Move</td>
<td>To move source files to a separate location after a successful transfer, check Automatically move source files to a directory after transfer and specify the location.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Only a path to an existing location on the client can be specified.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The GUI has no option to delete empty source subdirectories that may remain after source files are moved.</td>
</tr>
</tbody>
</table>
Important: When managing connections, changes are not saved until you click OK. Selecting Cancel will discard any unsaved changes made in the Connection Manager, including the addition and removal of connections.

To connect to this remote host, double-click the connection from the Connection panel, or select it and click Connect.

Importing and Exporting Connections

You may also import your connection list to and export your connection list from a text file. To export your connection list, right-click the remote server panel and select Export. To import your connection list, right-click the remote server panel and select Import. Both options are shown below (with "export" selected).

Note:

- If you are exporting a connection that uses keys, then you will need to back up those keys manually and import separately.
- A shared connection that is exported and imported by a non-administrator will import as a regular connection (not as shared).
- Email templates are not exported with the connection.

Creating SSH Keys

Create a key-pair for your computer.

Public key authentication (SSH Key) is a more secure alternative to password authentication that allows users to avoid entering or storing a password, or sending it over the network. Public key authentication uses the client computer to generate the key-pair (a public key and a private key). The public key is then provided to the remote computer's administrator to be installed on that machine. To use your Aspera product's transfer-client functionality with public key authentication, follow the steps below.

You can use the application GUI to generate key-pairs and to import existing key-pairs. You can also generate key-pairs using the command-line; for instructions, see Creating SSH Keys (Command Line) on page 156.
1. Create a key pair using the GUI

Start the application by launching `asperascp`. From the menu bar, select **Tools > Manage Keys**.

In the SSH Keys dialog, click `+` to bring up the New SSH Key Pair window.

The SSH Keys dialog is also available from the **Connection** tab in the Connection Manager. When you select **Public Key** for authentication, the **Manage Keys** button appears; clicking it opens the SSH Keys dialog.

In the New SSH Key Pair window, enter the requested information. Click **OK** when finished.
2. Distribute the public key

Then, you will need to provide the public key file (for example id_rsa.pub) to your server administrator, so that it can be set up for your server connection. To copy or export the public key, select the key in the Public Key Manager window, click **Copy Public Key to Clipboard**, and paste the string into an email and address it to the server administrator, or click **Export to File** and save the public key as a file. For information on how to install the public key on the server, see *Setting Up a User’s Public Key* on page 63; however, keep in mind that the server could be installed on an operating system that is different from the one where your client is installed.

You can find the public key in this path:
Transferring Files with the Application

3. Set up connections using public key authentication

When your public key has been installed on the remote host by its server administrator, click **Connections** to bring up the Connection Manager.

Click the **Connection** tab, then select the **Public Key** Authentication option and select the key that is installed on this host.

![Connection Manager](image)

**Note:** When you are sharing a connection with public key authentication (Share this connection with all users on this computer checkbox is checked), the SSH key should be shared as well.

To import keys created outside the GUI, go to **Tools > Manage Keys** to open the SSH Keys dialog. Clicking the **button** in the upper-left corner of the dialog opens a file browser. You can import the key pair by selecting either the private key or the public key; this will copy both keys into the user's .ssh directory. You cannot import a key pair if a key pair with the same identity already exists in the .ssh directory.

Imported key pairs can be shared with other users. In the SSH Keys dialog, select a key and click the **button** to open the Edit SSH Key Pair dialog. Select **Access** to allow shared connections to use this key. Shared keys are moved to the Enterprise Server etc directory.

### Enabling a Transfer or HTTP Proxy

Setting up your connection if you are behind a proxy server

If, for network-security reasons, you are behind a proxy server or an HTTP proxy server, you can enable these proxies for file transfer by configuring settings in the Preferences dialog. Preferences can be accessed either from the
Preferences button in the upper-right corner of the desktop client window menu, or from the Tools button in the main toolbar.

If you have admin privileges, you can enable transfer proxies for all users by setting global preferences. If you are a non-admin user, you can override global transfer-proxy settings for your own account, including enabling or disabling the feature.

By default, proxy settings are turned off.

**Global Proxy Settings**

To enable or adjust proxy settings globally, select Tools > Global Preferences. You must have admin privileges to set global preferences:

In the Proxy dialog, you can set the following:

**Enable transfer proxy**

- Check the Enable transfer proxy checkbox.
- Enter the proxy server's hostname or IP address and port number.
- Enable the Secure checkbox if your proxy server allows secure connections.
- Enter your username and password to authenticate with your proxy server.
Enable HTTP proxy

- Check the Enable HTTP proxy checkbox.
- Enter the HTTP proxy’s hostname or IP address and port number.
- If your HTTP proxy requires authentication, enable the Authenticated checkbox and enter the username and password for your HTTP proxy.

By default, all proxy settings are turned off. For global preferences, clicking Restore System Defaults clears all settings.

User Proxy Settings

To override the global settings, you can enter personal settings for your own account. Select Tools > Preferences or click the Preferences link in the upper-right corner of the desktop client window:
Under **Proxy**, the values inherited from the global proxy settings will be filled in initially. You can set the following:

**Enable transfer proxy**
- Check or uncheck **Enable transfer proxy** to enable or disable transfer proxy.
- Enter the proxy server's hostname or IP address and port number.
- Enable the **Secure** checkbox if your proxy server allows secure connections.
- Enter your username and password to authenticate with your proxy server.

You can also clear your personal settings by clicking **Restore Defaults**. Your settings will revert to the current global settings.

If you are an admin, you can access the global proxy dialog by clicking the **Global Preferences** button.

**Enable HTTP proxy**
- Check the **Enable HTTP proxy** checkbox.
- Enter the HTTP proxy's hostname or IP address and port number.
- If your HTTP proxy requires authentication, enable the **Authenticated** checkbox and enter the username and password for your HTTP proxy.
By default, all proxy settings are set to the global values. For personal preferences, clicking Restore Defaults changes all settings to the global values.

Transferring Files

Initiate and manage file transfers.

⚠️ Caution: Do not use the following characters in filenames:

```
/ \ " : ? > < & * |
```

1. Connect to the remote host

Start the application by launching `asperascp`, and double-click the connection within the Connection panel, or select it and click Connect.

In the connections panel, the Target Directory shows either a specific path when the target directory is set, or the last-visited folder when left blank. For how to set up the target directory, see Managing Connections on page 20.

2. Initiate the transfer

To transfer a file to or from the remote computer, select the file to transfer and then click the upload or download arrow.

You can transfer files or folders between the right and left browser panels using drag-and-drop or copy-and-paste. Within either the left or right browser panel, you can move files or folders using drag-and-drop or cut-and-paste, and you can copy them using copy-and-paste.

4. Transfer files without browsing the remote host
If you have entered the target directory for this connection (See Managing Connections on page 20), you can also transfer files without browsing the remote computer. To do so, select the files from the left panel (local), select the connection name from the right panel (remote) and click to push files to the remote computer's target directory (as shown in the screenshot), or to pull files from it.

Note: If you attempt to transfer too many files, regardless of the method, the transfer is disabled and the following warning message is displayed:
Too many files selected. Select fewer files, or transfer the folder containing your selection instead.

The file limit is OS dependent.

The limit does not apply to copy-and-paste operations within the same file browser panel.

5. Manage the transfer sessions in the Transfers panel
Once the transfer has been successfully initiated, a progress bar will appear in the Transfers panel. If you have multiple ongoing transfers, use the and to change the selected transfer's priority. The # field indicates the transfer's order in the queue. Also the , , and can be used to control the selected transfer session.

6. (Optional) Make adjustments to a transfer session's target rate, minimum rate and/or policy (if allowed)
The Details button provides additional visibility and control (if granted the proper permissions) over transfers. Select a transfer session from the Transfers panel and click Details to view details and/or adjust settings.

The following items are on the Details display:
### Transferring Files with the Application

<table>
<thead>
<tr>
<th>Item</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Details (tab)</td>
<td>Transfer details, including status (rate and ETA) and statistics (session size, files transferred vs. total files to be transferred, average speed, time elapsed, RTT delay and average loss in percent).</td>
</tr>
<tr>
<td>B</td>
<td>Files (tab)</td>
<td>All files being transferred in this session, along with each files' size and transfer progress.</td>
</tr>
<tr>
<td>C</td>
<td>Transfer controls</td>
<td>Set the FASP transfer policy and transfer rate, if allowed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- fixed – Attempts to transfer at the specified target rate, regardless of the actual network capacity. This policy transfers at a constant rate and finishes in a guaranteed time. This policy typically occupies most of the network's bandwidth, and is not recommended in most file transfer scenarios. In fixed mode, a maximum (target) rate value is required.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- high – Monitors the network and adjusts the transfer rate to fully utilize the available bandwidth up to the maximum rate. When congestion occurs, it transfers at a rate twice of a session with fair policy. In this mode, both the maximum (target) and the minimum transfer rates are required.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- fair – Monitors the network and adjusts the transfer rate to fully utilize the available bandwidth up to the maximum rate. When other types of traffic build up and congestion occurs, it shares bandwidth fairly by transferring at an even rate. In this mode, both the maximum (target) and the minimum transfer rates are required.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- low – Similar to fair mode, the low policy uses the available bandwidth up to the maximum rate, but is much less aggressive when sharing bandwidth with other network traffic. When congestion builds up, the transfer rate is reduced to the minimum rate until other traffic retreats.</td>
</tr>
<tr>
<td>D</td>
<td>Transfer Monitor</td>
<td>The transfer graph. Note that you may use the sliders to adjust the transfer rate up or down (if allowed).</td>
</tr>
</tbody>
</table>

---

7. **Update preferences for the transfer rate and maximum number of concurrent transfers**

If you have administrator privileges, you can set the target transfer rate for all users from the Global Preferences dialog. As an individual user, you can override the global settings from My Preferences. To update these settings, go to **Tools > Global Preferences** or **Tools > Preferences**. You can also open My Preferences from the...
Preferences button in the upper-right corner of the application's main window; from there you can also reach the Global Preferences dialog by clicking Global Preferences.

The following options are available under the Transfers tab:

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Bandwidth Limits</td>
<td>The aggregated bandwidth cap for all FASP transfers on this computer. For more advanced bandwidth settings, see Bandwidth on page 72. (Set by administrators only.)</td>
</tr>
<tr>
<td>Default Target Rate</td>
<td>The initial download and upload rates for all transfers.</td>
</tr>
<tr>
<td>Maximum Active Transfers</td>
<td>The maximum number of concurrent upload transfers and download transfers.</td>
</tr>
</tbody>
</table>

For information about settings under the Email tab, see Configuring Transfer Notifications on page 38.

Advanced Transfer Mode

More options for initiating transfers, such as filters, security, and scheduling.

You can start a transfer in advanced mode to set per-session transfer options that override the default transfer settings. To initiate a transfer in advanced mode, right-click a file or folder to open the context menu and select Upload (in the client panel) or Download (in the server panel).
The advanced transfer dialog includes the following configuration tabs:

<table>
<thead>
<tr>
<th>Tab</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transfer</td>
<td>The transfer session-related options, such as the transfer speed and retry rules.</td>
</tr>
<tr>
<td>Tracking</td>
<td>Options for tracking the transfer session, including the confirmation receipt and the email notifications.</td>
</tr>
<tr>
<td>Filters</td>
<td>Create filters to skip files that match certain patterns.</td>
</tr>
<tr>
<td>Security</td>
<td>Enable the transfer encryption and the content protection.</td>
</tr>
<tr>
<td>File Handling</td>
<td>Set up resume rule, preserve transferred file attributes, and remove or move source files.</td>
</tr>
<tr>
<td>Scheduling</td>
<td>Schedule this transfer.</td>
</tr>
</tbody>
</table>

**Note:** All configuration tabs, except **Scheduling**, are identical to those in the **Connection Manager** configuration screen. For information on these tabs, see **Managing Connections** on page 20. The **Scheduling** tab is described below.

**Scheduling**

To enable transfer scheduling, check the box for **Schedule this transfer**. When finished, click **Transfer**. The following scheduling options are available:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>Specify the transfer time.</td>
</tr>
<tr>
<td>Transfer repeats</td>
<td>Select a repeat mode.</td>
</tr>
<tr>
<td></td>
<td>For a single transfer, select <strong>Does not repeat</strong> and select a time and date.</td>
</tr>
<tr>
<td></td>
<td>For a daily transfer, select <strong>Daily</strong> and select a start time and an end date (either Never or a date and time).</td>
</tr>
<tr>
<td></td>
<td>For a daily transfer on weekdays only, select <strong>Monday - Friday</strong> and an end date (either Never or a date and time).</td>
</tr>
<tr>
<td></td>
<td>For a weekly transfer, select <strong>Weekly</strong>, select which day of the week, and specify an end date (either Never or a date and time). Note that with this option you can specify more than one day of the week to set specific days when the transfer should repeat.</td>
</tr>
</tbody>
</table>
For transfers that should repeat more frequently than daily, select Periodically and fill in the number of minutes between transfers.

When submitting a scheduled transfer, you will see it listed under the Transfers tab, along with an icon ( ) under the # column. To modify the transfer, right-click it and select Edit to reveal the transfer settings.

Note: When scheduling transfers, ensure that the application is running. Scheduled transfers do not run when the application is closed.

Configuring Transfer Notifications

Set up transfer notifications and modify the templates.

Transfer notification emails (which are based on default or customized mail templates) are triggered by three transfer session events: start, completion and error. Follow the instructions below to configure the SMTP server and/or to create/modify your email templates.

1. Launch Enterprise Server with root permissions

   Configuring transfer notifications requires root access. Run asperascp in a terminal shell as root to launch the application.

2. Configure global mail preferences

   Note: To configure global mail preferences, you must have root permissions.

   To set up global mail preferences, launch the application with root permissions, and select Tools > Global Preferences.
Click the **Mail** button to configure settings for email notifications. In the dialog that appears, select **Enable email notifications** to turn on email notifications for all users. If enabled, both a **From Address** and outgoing email server host name are required. To ensure that the mail server information is correct, click **Send test email**; a test message will be sent to the **From Address**.

**User Mail Preferences**

To override all global/default mail settings and enter personal settings for your own account, select **Tools > Global Preferences** or click the **Preferences** link in the upper-right corner of the main application window:
This opens the **My Preferences > Mail** dialog. When initially opened, this dialog is populated with the inherited global default values as set by an admin user. From here you can overwrite the inherited mail settings, including enabling or disabling notifications. To restore settings to the global values, click the **Restore Defaults** button.
3. Open the Mail Templates window

Templates are used to generate the content of notification emails. You can associate them with connections, hot folders, and individual transfers. We provide a default template. They can be changed to customize notification emails.

Click **Tools > Mail Templates** to bring up the Mail Templates window.

In the Mail Templates window, click **+** to create a template based on existing ones, or select an existing template and click **✓** to edit it.
The mail template supports MIME (Multipurpose Internet Mail Extensions) multipart messages that includes both the HTML and plain text versions of the mail body. In the Edit Template window, Enter the template in the specified field:

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The template name.</td>
</tr>
<tr>
<td>HTML</td>
<td>The HTML mail body. Click <strong>Insert Image</strong> to insert an image into the template. The selected image will be copied to the template directory. You may preview the template by clicking <strong>Preview</strong>.</td>
</tr>
<tr>
<td>Text</td>
<td>The plain text mail body. You may preview the template by clicking <strong>Preview</strong>.</td>
</tr>
<tr>
<td>Access</td>
<td>Check the option <strong>Share this template with all users on this computer</strong> to allow other system users to access this template.</td>
</tr>
</tbody>
</table>

4. Modify mail templates

Mail templates serve as models for the email that will be sent.
To modify mail templates, click **Tools > Mail Templates** to open the Mail Templates window.

![Mail Templates Window](image)

The templates are rendered using Apache Velocity (*Apache Velocity User Guide*). Content is generated for an email according to its template. A conditional statement only generates content if the condition matches. A `foreach` loop generates content for each iteration of the loop. Within a template, there are two predefined variables:

- `$formatter` - Contains some utility methods
- `$notifications` - Holds the transfer notifications

To iterate over notifications, use a `foreach` loop:

```
#foreach ($event in $notifications.getEvents())
...
#end
```

This declares a local `$event` variable that can be used within the for-each loop.

The following conditional statements can be used in the templates:

```
#if ...
#else ...
#end
```

All statements are categorized in four parts: conditional, session information, time, and statistics.

**Conditional**

Use these tests in an `if` statement. For example:

```
#if ($event.isFailed())
...
#end
```

<table>
<thead>
<tr>
<th>Statement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>$event.isStarted()</code></td>
<td>If the transfer session is started.</td>
</tr>
<tr>
<td><code>$event.isCompleted()</code></td>
<td>If the transfer session is completed.</td>
</tr>
<tr>
<td><code>$event.isEnded()</code></td>
<td>If the transfer session is ended.</td>
</tr>
</tbody>
</table>
### Session Information

<table>
<thead>
<tr>
<th>Statement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$event.isFailed()</td>
<td>If the transfer session is failed.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Statement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$event.getSourceHost()</td>
<td>The source hostname (or host address if the hostname is not discoverable).</td>
</tr>
<tr>
<td>$event.getSourceHostAddress()</td>
<td>The source host address.</td>
</tr>
<tr>
<td>$event.getSourcePaths()</td>
<td>The source file path.</td>
</tr>
<tr>
<td>$event.getDestinationHost()</td>
<td>The destination hostname (or host address if the hostname is not discoverable).</td>
</tr>
<tr>
<td>$event.getDestinationHostAddress()</td>
<td>The destination host address.</td>
</tr>
<tr>
<td>$event.getDestinationPath()</td>
<td>The destination file path.</td>
</tr>
<tr>
<td>$event.getInitiatingHost()</td>
<td>The session-initiating hostname (or host address if the hostname is not discoverable).</td>
</tr>
<tr>
<td>$event.getInitiatingHostAddress()</td>
<td>The session-initiating host address.</td>
</tr>
<tr>
<td>$event.getId()</td>
<td>The session ID.</td>
</tr>
<tr>
<td>$event.getName()</td>
<td>The session name.</td>
</tr>
<tr>
<td>$event.getType().getDescription()</td>
<td>The session state. Three outputs: &quot;STARTED&quot;, &quot;FAILED&quot;, and &quot;COMPLETED&quot;.</td>
</tr>
<tr>
<td>$event.getUser()</td>
<td>The transfer login.</td>
</tr>
<tr>
<td>$event.getFiles()</td>
<td>The files that are being transferred. Use this statement in a foreach loop: (Any text after ## is a comment)</td>
</tr>
</tbody>
</table>

```java
#foreach ($file in $event.getFiles())
## $file is a new variable visible in this foreach loop.
## $file holds the complete file path and file name.
## $formatter.decodePath() is used to ensure a correct string decoding.
$formatter.decodePath($file)
#end

And use the counter $velocityCount in an if statement to limit the output file count. For example, to list only the first ten files:

```java
#foreach ($file in $event.getFiles())
#if ($velocityCount > 10)
#break
#end
$file
#end
```
| $event.getMessage() | The message entered in the notification's "Message" field. |
| $event.getError() | The error message. |
Time

<table>
<thead>
<tr>
<th>Statement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$formatter.date(var, &quot;lang&quot;, &quot;format&quot;)</td>
<td>Formatting the date and time output. Enter three values in the parenthesis:</td>
</tr>
<tr>
<td>• Replace <code>var</code> with the following two statements; for example, $event.getStartTime()</td>
<td></td>
</tr>
<tr>
<td>• Replace <code>lang</code> with an abbreviate language name; for example, <code>en</code> for English.</td>
<td></td>
</tr>
<tr>
<td>• <code>format</code> is the display format. Use these symbols:</td>
<td></td>
</tr>
<tr>
<td>• <code>yyyy</code> The year; for example, 2010.</td>
<td></td>
</tr>
<tr>
<td>• <code>MM</code> Month of the year; for example, 03.</td>
<td></td>
</tr>
<tr>
<td>• <code>dd</code> Day of the month; for example, 26.</td>
<td></td>
</tr>
<tr>
<td>• <code>HH</code> Hour of the day; for example, 16.</td>
<td></td>
</tr>
<tr>
<td>• <code>mm</code> Minute.</td>
<td></td>
</tr>
<tr>
<td>• <code>ss</code> Second.</td>
<td></td>
</tr>
<tr>
<td>• <code>z</code> Time zone.</td>
<td></td>
</tr>
<tr>
<td>• <code>EEE</code> The abbreviated weekday name; for example, Fri.</td>
<td></td>
</tr>
<tr>
<td>For example, &quot;EEE, yyyy-MM-dd HH:mm:ss z&quot;</td>
<td>shows Fri, 2010-03-26 16:19:01 PST.</td>
</tr>
<tr>
<td>$event.getStartTime()</td>
<td>The session start time.</td>
</tr>
<tr>
<td>$event.getEndTime()</td>
<td>The session end time.</td>
</tr>
</tbody>
</table>

Statistics

<table>
<thead>
<tr>
<th>Statement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$event.getSourceFileCount()</td>
<td>The number of source files.</td>
</tr>
<tr>
<td>$event.getCompletedFileCount()</td>
<td>The number of files that successfully transferred.</td>
</tr>
<tr>
<td>$event.getFailedFileCount()</td>
<td>The number of files that failed to transferred.</td>
</tr>
<tr>
<td>$event.getAverageRatePercentage()</td>
<td>The average transfer rate in bps. Enclose this statement with $formatter.formatRate() to simplify the output.</td>
</tr>
<tr>
<td>$event.getAverageLossPercentage()</td>
<td>The average packet loss percentage.</td>
</tr>
<tr>
<td>$event.getSourceSizeB()</td>
<td>The source file size. Enclose this statement with $formatter.toBestUnit() to simplify the output.</td>
</tr>
<tr>
<td>$event.getTransferredB()</td>
<td>The transferred file size. Enclose this statement with $formatter.toBestUnit() to simplify the output.</td>
</tr>
<tr>
<td>$event.getWrittenB()</td>
<td>The destination file size. Enclose this statement with $formatter.toBestUnit() to simplify the output.</td>
</tr>
</tbody>
</table>

When configured, you can apply the notifications to a connection host, or a transfer session. For details, see Using Transfer Notifications on page 46.
Using Transfer Notifications

Use transfer notifications to send emails based on transfer events.

Transfer notifications can be sent for three transfer events: start, complete, and error. Follow these instructions to select and apply them to your transfer sessions:

1. Preview mail templates

You can preview existing templates to decide which one to use. In the application (asperascp), go to Tools > Mail Templates... to bring up the Mail Template window.

In the Mail Templates window, select an existing template and click to open the edit screen.

Mail templates supports MIME multipart messages, which include both HTML and plain text versions. In the Edit Template window, click Preview to view the template's output example.
2. Set up notifications for a connection

You can set up notifications for connections. When transferring with the host, emails will be sent to specified recipients on selected events.

To do so, click Connections, choose the connection, and select the Tracking tab. Check Send email notifications to enable this feature. Enter the following information, and then click OK:

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>When</td>
<td>Check the events to send notifications for.</td>
</tr>
<tr>
<td>To</td>
<td>Enter the recipients, comma separated.</td>
</tr>
<tr>
<td>Template</td>
<td>Select a mail template.</td>
</tr>
<tr>
<td>Item</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>Message</td>
<td>Optionally enter a message to include in the notifications.</td>
</tr>
</tbody>
</table>

3. Set up notifications for a transfer

Email notifications can also be applied to transfer sessions. Right click the file browser and select **Upload...** or **Download...** to open the advanced transfer window, select the **Tracking** tab, and check **Send email notifications** to enable this feature. Refer to the previous section for help on setting the options.

---

**Reporting Checksums**

Configure IBM Aspera Enterprise Server to report checksums for transferred files.

Internally, Enterprise Server determines the success of transfers by using checksums to verify that file contents at a destination match what was read at the source. Enterprise Server can also be configured to report these checksums to users.

**Note:** Checksum reporting requires that both the server and client nodes be running Enterprise Server, Connect Server, or Point-to-Point 3.4.2 or higher.

By default, checksum reporting is turned off. The feature can be enabled and configured on the server using any of the following methods:

- entering configuration options in `aspera.conf`
- setting configuration options in the desktop client GUI
- on a per-transfer basis, using a command-line option with `ascp`

If used, the command-line option overrides settings in `aspera.conf` and the GUI.

Each method allows you to enable checksum reporting by selecting or setting the following options:

- **md5** - Calculate and report an MD5 checksum.
- **sha1** - Calculate and report an SHA-1 checksum.
- **any** - Allow the checksum format to be whichever format the client requests.

Additional options in `aspera.conf` and the GUI allow you to configure where checksum reports should be saved.

**Enabling from aspera.conf**

Open the `aspera.conf` file on your server and add the `<file_checksum>` option to the `<file_system>` section, as in the example below.
**Note:** The none option is no longer supported as of 3.4.2. If your aspera.conf file has a `<file_checksum>` setting of none, transfers will fail with the error "Server aborted Session: Invalid configuration file".

To enable and configure the file manifest where checksum report data will be stored, add settings for `<file_manifest>` and `<file_manifest_path>`; for example:

```xml
<file_system>
  ...
  <file_checksum>md5</file_checksum> <!-- Enable checksum reporting (md5, sha1, any) -->
  <file_manifest>text</file_manifest> <!-- Enable file manifest (text, disable) -->
  <file_manifest_path>/tmp</file_manifest_path> <!-- Path to manifest file -->
  ...
</file_system>
```

The following table provides details on the configuration options for checksum reporting:

<table>
<thead>
<tr>
<th>Conf Option / GUI Config Setting</th>
<th>Description</th>
<th>Values</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;file_checksum&gt;</code></td>
<td>File checksum method</td>
<td>Enable checksum reporting, specifying the type of checksum to calculate for transferred files.</td>
<td>md5, sha1, or any</td>
</tr>
<tr>
<td><code>&lt;file_manifest&gt;</code></td>
<td>File Manifest</td>
<td>When set to text a text file &quot;receipt&quot; of all files within each transfer session is generated. If set to disable, no file manifest is created. The file manifest is a file containing a list of everything that was transferred in a given transfer session. The filename of the file manifest itself is automatically generated based on the transfer session's unique ID.</td>
<td>text, disable</td>
</tr>
<tr>
<td><code>&lt;file_manifest_path&gt;</code></td>
<td>File Manifest Path</td>
<td>The location where manifest files are to be written. The location can be an absolute path or a path relative to the transfer user's home. If no path is specified, the file will be generated under the destination path at the receiver, and under the first source path at the sender.</td>
<td>path name</td>
</tr>
</tbody>
</table>

Enabling from the GUI

Click **Configuration** to open the Server Configuration window. Select the **Global**, **Groups**, or **Users** tab, depending on whether you want to configure checksum reporting for all users, or for a particular group or user. Under the **File Handling** tab, locate the setting for **File checksum method**. Check the override box and for the effective value, select md5, sha1, or any.
To enable the file manifest from the GUI, locate the **File Manifest** setting. Check the override box and set the effective value to **text**.

Locate the **File Manifest Path** setting on the line just below. Check the override box and set the effective value to a folder where the manifest files are to be saved.

In the above examples, when files are transferred, the manifest is generated to a text file called `aspera-transfer-transfer_id-manifest.txt` in the directory `/tmp`. 
For details about the settings for File checksum method, File Manifest, and File Manifest Path, see the table of configuration options in the previous section.

**Enabling from the ascp Command Line**

To enable checksum reporting on a per-transfer-session basis, run `ascp` with the `--file-checksum=hash` option, where `hash` is sha1, md5, or any.

From the `ascp` command line, you can also enable the manifest with the option `--file-manifest=output` where `output` is either text or none. You can set the path to the manifest file with the option `--file-manifest-path=path`.

For example:

```
$ ascp --file-checksum=md5 --file-manifest=text --file-manifest-path=/tmp file aspera_user_1@189.0.202.39:/destination_path
```

**Setting up a Pre/Post-processing Script**

An alternative to enabling and configuring the file manifest to collect checksum reporting is to set up a pre/post-processing script to report the values.

The checksum of a successfully transferred file is stored in the pre/post environment variable `FILE_CSUM`. This environment variable can be used in pre/post scripts to capture file checksums. For example, the following script outputs the checksum to the file `/tmp/cksum.log`:

```
#!/bin/bash
if [ $TYPE == File ] && then
  if [ $STARTSTOP == Stop ]; then
    echo "The file is: $FILE" >> /tmp/cksum.log
    echo "The file checksum is: $FILE_CSUM" >> /tmp/cksum.log
    chmod 777 $FILE
  fi
fi
```

For information on how to set up pre- and post-processing scripts such as the above and how to use builtin pre/post environment variables, see *Pre- and Post-Processing (Prepost)* on page 128.

**Configuring Multi-Session Transfers**

The Aspera transfer server products Enterprise Server and Connect Server can achieve significant performance improvements by using multi-session transfers (also known as parallel transfers and multi-part transfers) on multi-node and multi-core systems.

To enable multi-session transfers, run `ascp` with the option `--C nid:ncount`, where `nid` is the node ID and `ncount` is the number of nodes or cores. Assign each session (or invocation) its own UDP port.

You can also specify that individual files may be split between multiple sessions by specifying the companion option `--multi-session-threshold=threshold`. The threshold value specifies, in bytes, the smallest-size file that can be split. That is, files greater than or equal to the threshold will be split, while those smaller than the threshold will not.

A default value for the threshold can be specified in the `aspera.conf` file by setting `<multi-session_threshold_default>` in the `<default>` section. Setting it to 0 (zero) indicates that files should not be split. The command-line setting overrides the `aspera.conf` setting. If the client's `aspera.conf` does not specify a default value for the threshold, then the default is taken from the server's `aspera.conf` (if specified).

If neither `--multi-session-threshold` nor `<multi-session_threshold_default>` is specified, then no files are split.
Using Multi-Session to Transfer Between Nodes

The following example shows a multi-session transfer on a dual-core system, which together can transfer at up to 200 Mbps. Each command uses a different UDP port, and each is run from a different terminal window. No multi-session threshold is specified either on the command line or in aspera.conf; therefore no file splitting occurs.

```
ascp -C 1:2 -O 33001 -l 100m /dir01 10.0.0.2:/remote_dir
ascp -C 2:2 -O 33002 -l 100m /dir01 10.0.0.2:/remote_dir
```

Assuming there are multiple files in dir01, ascp will distribute the files between each command to get the most efficient throughput. If there is only one file in dir01, only one of the commands will actually transfer the file.

In the following example, the multi-session threshold is added to the command line from the example above. This enables file splitting and specifies the threshold size, which is the minimum-size file that can be split.

```
ascp -C 1:2 -O 33001 -l 100m --multi-session-threshold=5242880 /dir01 10.0.0.2:/remote_dir
ascp -C 2:2 -O 33002 -l 100m --multi-session-threshold=5242880 /dir01 10.0.0.2:/remote_dir
```

In this case, if there are multiple files in dir01, all files less than 5 MB will be distributed between each command, while all files 5 MB or greater will be split to further achieve a level distribution. If there is only one file in dir01, it will be split if it's 5 MB or larger, otherwise the entire file will be transferred by only one of the commands.

Using Multi-Session to Transfer to an Aspera Transfer Cluster

**Note:** For cloud transfers, file-splitting is currently supported for AWS S3 only.

For transfers to cloud storage, the process of splitting files in multi-session transfers differs from regular (non-cloud) multi-session transfers. For cloud transfers, files are sent in chunks, and the chunk size is specified by `<chunk_size>` in aspera.conf:

```
<default>
  . . .
  <transfer>
    <protocol_options>
      <chunk_size>size_in_bytes</chunk_size>
    </protocol_options>
  </transfer>
</default>
```

For cloud storage, file-splitting needs to respect a minimum split size, which for cloud storage is a *part*. Part size must be set to the same value as the `ascp` chunk size. That is, each `ascp` call needs to deliver full parts. However, a file that would normally be split (due to being larger than the multi-session threshold) will not be split if it is smaller than the chunk/part size. Set chunk size and part size as follows:

1. **In** aspera.conf **set the chunk size to some value greater than 5 MB; for example:**

   `<chunk_size>67108864</chunk_size>  <!-- 64 MB -->`

2. **In** /opt/aspera/etc/trapd/s3.properties:
   - **Set the upload part size (default 64 MB) to the same value as the chunk size.**
   - **Use a ONE_TO_ONE gathering policy:**

   ```
aspera.transfer.upload.part-size=64MB
aspera.transfer.gathering-policy=ONE_TO_ONE
```
The following example uploads an 80 GB file into AWS S3 by means of a multi-session transfer to an Aspera Transfer Cluster. The setup for this example consists of one 10 Gbps system and a 20-node Aspera Transfer Cluster (ATC).

### Configuring the Aspera Transfer Cluster

1. Log into the Aspera Transfer Cluster Manager (ATCM) and check that the following exists in the cluster's Transfer Configuration:

   ```
   <default>
   <transfer>
     <protocol_options>
       <chunk_size>67108864</chunk_size>
     </protocol_options>
   </transfer>
   </default>
   ``

   To show the cluster's transfer configuration, select the cluster (in this case `jmoore-se-demo-cluster`) and click the **Action** button and select **Edit Transfer Configuration**:

2. From the **Action** drop-down menu, select **Edit Auto Scaling Policy**. Configure the cluster for at least 20 static nodes by setting **Max Nodes** and **Min Available Nodes** to 20 as shown below. Also ensure that **Max Start Frequency Count** is greater-than or equal-to the values for **Max Nodes** and **Min Available Nodes**.

### Configuring the Aspera Client Transfer System

Configure Aspera Enterprise Server or Connect Server as in the following example `aspera.conf` file:

1. Configure Aspera Enterprise Server or Connect Server as in the following example `aspera.conf` file:

   ```xml
   <?xml version='1.0' encoding='UTF-8'?><CONF version="2"
   <default>
     <file_system>
       <access>
         <paths>
           <path>
             <restrictions>
               <restriction>*</restriction>
             </restrictions>
           </path>
         </paths>
       </access>
     </file_system>
     <transfer>
       <multi_session_threshold_default>100000000</multi_session_threshold_default>
     </transfer>
   </default>
   <aaa/>
   <server>
     <server_name>66.211.105.165</server_name>
   ```
2. Create a JSON transfer request file, `ms-send-80g.json`, containing the following:

```json
{
   "transfer": {
      "remote_host": "jmoore-se-demo-cluster.dev.asperacloud.net",
      "remote_user": "xfer",
      "token": "Basic QVVrZ3VobUNsdjBsdjNoYXAxWnk6dXI0VGZXNW5rNldBVW1zSm5FRzFV2WFvUXFTRUtLd3JmanhvNE2IQQZKZnFt",
      "target_rate_kbps": 700000,
      "destination_root": "/",
      "multipart": 75,
      "paths": [
         {
            "source": "/80g-file"
         }
      ],
      "ssh_port": 33001,
      "fasp_port": 33001,
      "direction": "send",
      "overwrite": "always",
      "cookie": "multi-session upload"
   }
}
```

3. Initiate the transfer through the Node API with an HTTP POST of the JSON transfer request using a `curl` command as follows:

```bash
curl -k -v -X POST -d @ms-send-80g.json https://ak_data:aspera@localhost:9092/transfers
```

4. Monitor transfer progress, bandwidth utilization, and the distribution of transfers for each cluster node.

On UNIX/Linux systems, you can view bandwidth utilization from a terminal by running `nload` on the client system with the following command:

```bash
nload -u g -o 10000000 ens1f0
```

The `nload` report below shows bandwidth utilization at 9+ Gbps:

In the ATCM UI, selecting **Monitor Nodes** from the **Action** drop-down menu shows the transfer distribution and utilization for each of the 20 nodes in the cluster:
Managing Users

Add users for the FASP connection authentication, and set up user transfer settings.

Setting Up Users

Add users for the FASP connection authentication, and set up user transfer settings.

Note: This topic describes setting up transfer user accounts with the GUI. If you are setting up users in a terminal, see Setting Up Transfer Users (Terminal) on page 59.

Your Aspera server uses your system accounts to authenticate connections. These system accounts must be added and configured before attempting an Aspera transfer. When creating transfer accounts, you may also specify user-based settings, including those for bandwidth, document root (docroot) and file handling.

You must create systems accounts for transfer users before they can be configured on your Aspera server. After these system accounts have been created and initialized on your local host, follow the steps below to configure their transfer accounts.

1. Restrict user permissions with aspshell.

   By default, all system users can establish a FASP connection and are only restricted by file permissions. You can restrict the user's file manipulation operations through the aspshell, which permits only the following operations:
   • Run Aspera uploads and downloads to or from this computer.
   • Establish connections in the application and browse, create, delete, rename or list contents.

   The following instructions demonstrate how to change a user account so that it uses the aspshell. Keep in mind that this is an example, and there may be other ways to do so for your system.

   (1) Open the following file with a text editor:

   /etc/passwd

   (2) Locate the entry for aspera_user_1. For example:

   ...
   aspera_user_1:x:501:501::/home/aspera_user_1:/bin/bash
   ...

   Replace the user's shell, in this case /bin/bash, with /bin/aspshell (or add the shell setting if it does not already exist):

   ...
   aspera_user_1:x:501:501::/home/aspera_user_1:/bin/aspshell
   ...

   You can also restrict a user's file access with Document Root (docroot) settings. The instructions are explained in the following steps.

2. Add a system user to your Aspera server.

   Run asperascp in a terminal shell as root to launch the application. In the UI, click Configuration.
In Server Configuration, select the **Users** tab and click the **+** button.

3. **Set up user's docroot.**

You can limit a user's access to a given directory using the document root (docroot). To set it up, click **Configuration>Users>username>Docroot**. Check the **Override** box for **Absolute Path** and enter or select an existing path as the user's docroot -- for example, `/sandbox/aspera_user_1`. Make sure that at least the **Read Allowed** and **Browse Allowed** are set to **true**. When finished, click **OK** or **Apply**.

If there is a pattern in the docroot of each user, for example, `/sandbox/username`, you can take advantage of a substitutional string. This allows you to assign an independent docroot to each user without setting it individually for each user.

<table>
<thead>
<tr>
<th>Substitutional String</th>
<th>Definition</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(name)</td>
<td>system user's name</td>
<td>/sandbox/$(name)</td>
</tr>
<tr>
<td>$(home)</td>
<td>system user's home directory</td>
<td>$(home)/Documents</td>
</tr>
</tbody>
</table>

Set up a docroot with a substitutional string as follows: in the Server Configuration dialog, select the **Global** tab and the **Docroot** tab, and enter the docroot into the **Absolute Path** field. This value will be duplicated in all user settings.
Testing a User-Initiated Remote Transfer

Test FASP transfers initiated from a client computer.

**Important:** These instructions require you to take steps on both the Enterprise Server and a client computer. Ensure that you are performing the task on the indicated machine. As a prerequisite, Enterprise Server must have at least one transfer user. For instructions on adding a transfer user, see *Setting up Users*.

1. **On your client machine, verify your connection to Enterprise Server.**
   
   On the client machine, use the ping command in a Terminal window to verify connectivity to the host. In this example, the address of Enterprise Server is 10.0.0.2.

   ```
   $ ping 10.0.0.2
   PING 10.0.0.2 (10.0.0.2): 56 data bytes
   64 bytes from 10.0.0.2: icmp_seq=0 ttl=64 time=8.432 ms
   64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=7.121 ms
   64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=5.116 ms
   64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=4.421 ms
   64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=3.050 ms
   ...  
   ```

2. **On your client machine, initiate a transfer to Enterprise Server.**
   
   Attempt to establish a connection from your client machine to Enterprise Server. To do so, run the following command on your client machine (where `aspera_user_1` is the example transfer user):

   ```
   $ ascp -P 33001 -T --policy=fair -l 10000 -m 1000 /client-dir/files aspera_user_1@10.0.0.2:/dir
   ```

<table>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host Address</td>
<td>10.0.0.2</td>
</tr>
<tr>
<td>Transfer User</td>
<td>aspera_user_1</td>
</tr>
<tr>
<td>Item</td>
<td>Value</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------------------------------</td>
</tr>
<tr>
<td>Files to upload</td>
<td>/client-dir/files</td>
</tr>
<tr>
<td>Destination Folder</td>
<td>{user's docroot}/dir</td>
</tr>
</tbody>
</table>
| Transfer Options  | • Maximum transfer rate = 10 Mbps (-l 10000)  
|                   | • Minimum transfer rate = 1 Mbps (-m 1000)   
|                   | • Change default TCP port used for FASP session initiation = 33001 (-P 33001).  
|                   | • Disable encryption (-T)               
|                   | • Fair transfer policy (--policy=fair)   |

If you cannot establish a connection to Enterprise Server, see *Clients Cannot Establish Connection*.

### Setting Up Groups

Create system groups on your computer, and set up transfer settings for the group and its members.

**Note:** This topic demonstrates how to set up transfer/user groups within the GUI. If you are configuring transfer/user groups within a Terminal, please refer to *Setting Up Transfer Groups (Terminal)* on page 60.

You can set up transfer settings based on your system's user groups. If users within a group do not have individual transfer settings, then the group's transfer settings will be applied. Please note that Enterprise Server doesn't create user groups on the operating system for you, so you must ensure that the groups currently exist before adding them to your Aspera product. Follow the steps below to add user groups to Enterprise Server.

1. Determine the user group(s) that you would like to add to your Aspera transfer product
   
   Ensure that you have an existing user group on your operating system, or create a new user group. For information on creating user groups, see your operating system documentation.

2. Add the user group to your Aspera transfer product
   
   Run `asperascp` in a Terminal window (as a root user) to launch the application, and click **Configuration**.

   Within the Server Configuration window, select the **Groups** tab, click **+** and input the user group's name.

3. Configure the group's transfer settings
Go to **Configuration** and select the **Groups** tab. Choose your group, and utilize the **Docroot**, **Authorization**, **Bandwidth**, **Network**, **File Handling** and **Precedence** tabs to configure the transfer settings. Refer to the hyperlinked topics below for additional information.

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Document Root</strong> on page 64</td>
<td>The document root settings.</td>
</tr>
<tr>
<td><strong>Authorization</strong> on page 67</td>
<td>Connection permissions, token key, and encryption requirements.</td>
</tr>
<tr>
<td><strong>Bandwidth</strong> on page 72</td>
<td>Incoming and outgoing transfer bandwidth and policy settings.</td>
</tr>
<tr>
<td><strong>Network</strong> on page 76</td>
<td>Network IP, port, and socket buffer settings.</td>
</tr>
<tr>
<td><strong>File Handling</strong> on page 78</td>
<td>File handling settings, such as file block size, overwrite rules, and exclude pattern.</td>
</tr>
<tr>
<td><strong>Configuration Precedence</strong> on page 62</td>
<td>When a user is a member of multiple groups, the precedence setting can be used to determine priority.</td>
</tr>
</tbody>
</table>

## Setting Up Transfer Users (Terminal)

Add system users on your computer, and configure the account for the *fasp* transfer.

Aspera transfer products use system accounts for connection authentication, and these accounts require additional configuration for Aspera transfers. You can specify user-based settings, such as bandwidth, document root (docroot), and file handling rules.

Follow these steps to set up transfer accounts in a command terminal:

1. **Open `aspera.conf` with a text editor.**

   To set up system users for FASP files transfers, locate the Aspera transfer product's configuration file, `aspera.conf`, and open it with a text editor:

   ```
   /opt/aspera/etc/aspera.conf
   ```

   You can find an example of `aspera.conf` in the following location:

   ```
   /opt/aspera/etc/samples/aspera-everything.conf
   ```

   The following steps explain how to update this file.

2. **Restrict user permissions with `aspshell`.**

   By default, all system users can establish a FASP connection and are only restricted by file permissions. You can restrict the user's file operations through the `aspshell`, which permits only the following operations:

   - Running Aspera uploads and downloads to or from this computer.
   - Establishing connections in the application, and browsing, creating, deleting, renaming, or listing contents.

   The following steps explain how to change a user account so that it uses the `aspshell`. Keep in mind that this is an example, and there may be other ways to do so on your system. Open the following file with a text editor:

   ```
   /etc/passwd
   ```

   Add or replace the user's shell with `aspshell`. For example, to apply `aspshell` to the user `aspera_user_1`, use the following settings in this file:

   ```
   ...
   aspera_user_1:x:501:501:::/home/aspera_user_1:/bin/aspshell
   ...
   ```
You can also restrict a user's file access with docroot (document root) settings in the `<file_system/>` section of `aspera.conf`, using the following tags: `<absolute/>, `<read_allowed/>, `<write_allowed/>, and `<dir_allowed/>`. For details, see `aspera.conf - File System` on page 93.

3. Configure a user's transfer settings.

Besides the default (global) transfer settings, you can also create user-specific and group-specific transfer settings. The user-specific settings have the highest priority, overriding both group and global settings.

Add the following section to `aspera.conf`:

```xml
<?xml version='1.0' encoding='UTF-8'?>
<CONF version="2">
  <aaa>
    <realms>
      <realm>
        <users>
          <user>
            <name> <!-- user name -->
            <authorization>... <!-- authorization settings -->
            <transfer>... <!-- transfer settings -->
            <file_system>... <!-- file system settings -->
          </user>
          <!-- another user's profile -->
          ... <!-- settings -->
        </users>
      </realm>
    </realms>
  </aaa>
</CONF>
```

4. Verify the configuration.

When you have finished updating the user's settings in the `aspera.conf`, use the following command to verify it. (In this example, verify the settings for user asp_1):

```
$ /opt/aspera/bin/asuserdata -b -u asp_1
```

---

**Setting Up Transfer Groups (Terminal)**

Create system groups on your computer, and set up transfer settings for the group and its members.

You can set up transfer settings based on your system's user groups. If users within a group do not have individual transfer settings, then the group's transfer settings will be applied. Please note that APOD/SODEnterprise Server doesn't create user groups on the operating system for you, so you must ensure that the groups currently exist before adding them to your Aspera product. Follow the steps below to add user groups to APOD/SODEnterprise Server in a Terminal.

1. Determine the user groups you would like to add to your Aspera transfer product.

   Ensure that you have an existing user group on your operating system, or create a new user group. Please refer to your operating system's documentation for information on creating user groups. APOD/SODEnterprise Server reads group information from the following file:

   `/etc/group`

2. Add the user group to your Aspera transfer product
When a transfer group is specified, it overwrites global settings and applies group configuration to corresponding users. To add group-specific transfer settings, open your `aspera.conf` file with a text editor.

```
/opt/aspera/etc/aspera.conf
```

You can access an example `aspera.conf` file at the following location:

```
/opt/aspera/etc/samples/aspera-everything.conf
```

Add the following section to `aspera.conf`:

```
<?xml version='1.0' encoding='UTF-8'?>
<CONF version="2">
  <aaa>
    <realms>
      <realm>
        <users>
          ... <!-- user-specific settings -->
        </users>
        <groups>
          <group> <!-- Each group tag contains a group's profile. -->
            <name>aspgroup</name> <!-- The group name. -->
            <precedence>0</precedence> <!-- Group precedence. -->
            <authorization>...</authorization> <!-- Authorization settings. -->
            <transfer>...</transfer> <!-- Transfer settings. -->
            <file_system>...</file_system> <!-- File System settings. -->
          </group>
          <!-- Another group's settings-->
        </groups>
      </realm>
    </realms>
  </aaa>
</CONF>
```

See the following sections for precedence, authorization, transfer, and file system configuration options:

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Configuration Precedence</strong> on page 62</td>
<td>When a user is a member of multiple groups, the precedence setting can be used to determine priority.</td>
</tr>
<tr>
<td><code>aspera.conf</code> - Authorization on page 83</td>
<td>Connection permissions, token key, and encryption requirements.</td>
</tr>
<tr>
<td><code>aspera.conf</code> - Transfer on page 85</td>
<td>Incoming and outgoing transfer bandwidth and policy settings.</td>
</tr>
<tr>
<td><code>aspera.conf</code> - File System on page 93</td>
<td>Docroot, file and directory creation, access permissions, block sizes, and so on.</td>
</tr>
</tbody>
</table>

3. Verify your configuration.

When you have finished updating the group's settings in `aspera.conf`, use the following command to verify it (in this example, verify the group `asp-group's settings):

```
$ /opt/aspera/bin/asuserdata -g asp-group
```
Configuration Precedence

The priority of user, group, global, and default settings.

Enterprise Server APD/SOD gives precedence to settings as follows, where user settings have the highest priority and default settings have the lowest.

1. User
2. Group(s) (If a user belongs to more than one group, a precedence can be set for each group.)
3. Global
4. Default

If a user is a member of multiple groups, a precedence setting can be assigned to each group. The following table shows the setting values that a user aspera_user_1 is assigned in bold. In this example, aspera_user_1 is a member of both the admin and xfer groups. The admin group's precedence setting is 0, which supersedes the xfer group's setting of 1:

<table>
<thead>
<tr>
<th>Options</th>
<th>User aspera_user_1's Settings</th>
<th>Group admin's Settings</th>
<th>Group xfer's Settings</th>
<th>Global Settings</th>
<th>Default Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target rate</td>
<td>5M</td>
<td>10M</td>
<td>15M</td>
<td>40M</td>
<td>45M</td>
</tr>
<tr>
<td>Min rate</td>
<td>n/a</td>
<td>2M</td>
<td>8M</td>
<td>3M</td>
<td>0</td>
</tr>
<tr>
<td>Policy</td>
<td>n/a</td>
<td>n/a</td>
<td>Low</td>
<td>Fair</td>
<td>Fair</td>
</tr>
<tr>
<td>Docroot</td>
<td>n/a</td>
<td>n/a</td>
<td>/pod/$(name)</td>
<td>n/a</td>
<td></td>
</tr>
<tr>
<td>Encryption</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>any</td>
<td></td>
</tr>
</tbody>
</table>

You can configure a group's precedence from the GUI or by editing aspera.conf. To configure it from the GUI, launch the application and click Configuration.

In the Server Configuration dialog, select the Groups tab, choose a group, and select the Precedence tab. (The Precedence tab does not appear if there are no groups.) Click the Override checkbox to override the inherited value (default), and enter a precedence number for the group.

Note: A group's precedence setting must be greater than or equal to 0, where 0 is the highest precedence level.

You can configure a group's precedence by editing aspera.conf. Before assigning group precedence by editing aspera.conf, first ensure that the groups have already been added in the application, so that they will appear as entries in aspera.conf.
Locate the `aspera.conf` file as follows:

```
/opt/aspera/etc/aspera.conf
```

In the file, locate the entry for each group, add the `<precedence>` option, and assign a precedence value as shown in the example below.

```
<groups>
  <group>
    <name>admin</name>
    <precedence>0</precedence>
    ...
  </group>
  <group>
    <name>xfer</name>
    <precedence>1</precedence>
    ...
  </group>
</groups>
```

### Setting Up a User's Public Key

Install the public key provided by the clients to their user account.

Public key authentication is an alternative to password authentication, providing a more secure authentication method that allows users to avoid entering or storing a password, or sending it over the network. It is done by using the client computer to generate the key-pair (a public key and a private key), provide the public key to the server or the point-to-point, and have the public key installed on that machine.

1. Obtain the client's public key.

   The client should send you an e-mail with the public key, either a text string attached in the secure e-mail, or saved as a text file. In this example, the client's login user account is `aspera_user_1`.

   For instructions of creating public keys, see *Creating SSH Keys* on page 26, or *Creating SSH Keys (Command Line)* on page 156.

2. Install the client's public key to its login user account.

   To install the account's public key, create a folder called `.ssh` in the user's home directory. This example sets up the public key for the following user:

<table>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>User name</td>
<td>aspera_user_1</td>
</tr>
<tr>
<td>Key file</td>
<td>/tmp/id_rsa.pub</td>
</tr>
<tr>
<td>Public key install location</td>
<td>/home/aspera_user_1/.ssh/authorized_keys</td>
</tr>
</tbody>
</table>

Run the following commands to install the client's public key:

```
$ mkdir /home/aspera_user_1/.ssh
$ cat /tmp/id_rsa.pub >> /home/aspera_user_1/.ssh/authorized_keys
$ chown -R aspera_user_1:aspera_user_1 /home/aspera_user_1/.ssh
```
The general transfer configuration options.

**Document Root**

The document root settings.

The document root (Docroot) configuration options can be found in the application's Configuration, under the Global, Groups, and Users tabs.

The following table lists all configuration options:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Values</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Absolute Path</td>
<td>The Absolute Path is a path to the docroot, the area of the file system that is accessible to Aspera users. The default empty value gives users access to the entire file system. In aspera.conf, you can set multiple docroots and make them conditional based on the IP address from which the connection is made. To do so, set the absolute path as follows:</td>
<td>file path or Amazon S3 URI</td>
<td>blank</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;absolute peer_ip=&quot;ip_address&quot;&gt;path&lt;/absolute&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>You may also specify an Amazon S3 docroot in the following URI format: s3://MY_ACCESS_ID:<a href="mailto:MY_SECRET_KEY@s3.amazonaws.com">MY_SECRET_KEY@s3.amazonaws.com</a>/my_bucket/my_path</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
<td>Values</td>
<td>Default</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
<td>--------</td>
<td>---------</td>
</tr>
<tr>
<td></td>
<td>(where each of the MY_ACCESS_ID, MY_SECRET_KEY and my_bucket/my_path parts must be url_encoded). S3 server side options are specified through an additional query part in the URI, as shown below.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>s3://MY_ACCESS_ID:<a href="mailto:MY_SECRET_KEY@s3.amazonaws.com">MY_SECRET_KEY@s3.amazonaws.com</a>/my_bucket/my_path?storage-class=REDUCED_REDUNDANCY&amp;server-side-encryption=AES256</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Valid values are as follows:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• For storage-class: STANDARD (default if not specified) or REDUCED_REDUNDANCY.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• For server-side-encryption: AES256 is the only valid value.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Read Allowed</td>
<td>Setting this to true allows users to transfer from the designated area of the file system as specified by the Absolute Path value.</td>
<td>• true • false</td>
<td>blank</td>
</tr>
<tr>
<td>Write Allowed</td>
<td>Setting this to true allows users to transfer to the designated area of the file system as specified by the Absolute Path value.</td>
<td>• true • false</td>
<td>blank</td>
</tr>
<tr>
<td>Browse Allowed</td>
<td>Setting this to true allows users to browse the directory.</td>
<td>• true • false</td>
<td>blank</td>
</tr>
</tbody>
</table>

**Advanced Symbolic Link Options (ascp)**

Client-side handling of symbolic links is configured from the following `ascp` command line:

```
$ ascp --symbolic-links=option
```

The following section describes the possible configuration options:

### Configuration Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>copy</td>
<td>Copy only the alias file. If a file with the same name exists at the destination, the symbolic link will not be copied.</td>
</tr>
<tr>
<td>copy+force</td>
<td>Copy only the alias file. If a file with the same name exists at the destination, the symbolic link will replace the file. If the file of the same name at the destination is a symbolic link to a directory, it will not be replaced.</td>
</tr>
<tr>
<td>follow</td>
<td>Follow symbolic links and transfer the linked files. This is the default option.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>skip</td>
<td>Ignore the symbolic link.</td>
</tr>
</tbody>
</table>

Server-Side Symbolic Link Handling

The following section describes how Aspera handle symbolic links in\textit{ascp} based on settings configured in the \textit{aspera.conf} file. The \textit{aspera.conf} file can be found in the following location:

\texttt{/opt/aspera/etc/aspera.conf}

Configuration Options

The following configuration options are set in the \texttt{<file_system>} section of the \textit{aspera.conf} file:

\begin{verbatim}
<file_system>
  <symbolic_links>list_of_comma-separated_options</symbolic_links>
</file_system>
\end{verbatim}

\textbf{Note:} If no option is specified, the configuration defaults to create, follow.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Client Behavior</th>
<th>Server Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>create</td>
<td>Create symbolic links with arbitrary targets. This is option set by default.</td>
<td>Skip if not configured.</td>
<td>Symbolic links are always copied to the server if the client requests.</td>
</tr>
<tr>
<td>follow</td>
<td>Follow symbolic links with targets inside docroot. If at any point the path goes outside the docroot, \textit{ascp} will not complete the transfer. This is option set by default.</td>
<td>Symbolic links are always copied to the server if the client requests.</td>
<td>Skip if not configured. Follow symbolic links with targets inside the docroot.</td>
</tr>
<tr>
<td></td>
<td>\textbf{Note:} If the docroot is a symbolic link and is specified as the source or destination: As the receiver, follow the target widely (no docroot constraint) and unconditionally (regardless of symbolic link action(s) configured/ requested).</td>
<td></td>
<td>\textbf{Note:} If the docroot is a symbolic link and is specified as the source or destination: As the sender, follow the target widely (no docroot constraint) and unconditionally (regardless of symbolic link action(s) configured/ requested).</td>
</tr>
<tr>
<td>follow_wide</td>
<td>Follow symbolic links with arbitrary targets, even if the targets are outside the docroot.</td>
<td>Symbolic links are always copied to the server if the client requests.</td>
<td></td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
<td>Client Behavior</td>
<td>Server Behavior</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>----------------</td>
<td>----------------</td>
</tr>
<tr>
<td>none</td>
<td>Take no action with the symbolic link.</td>
<td>or destination: As the receiver, follow the target widely (no docroot constraint) and unconditionally (regardless of symbolic link action(s) configured/requested).</td>
<td></td>
</tr>
</tbody>
</table>

**Authorization**

Connection permissions, token key, and encryption requirements.

The **Authorization** configuration options can be found in the application's **Configuration**, under the **Global**, **Groups**, and **Users** tabs.

The following table lists all configuration options:
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Values</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Incoming Transfers</strong></td>
<td>The default setting of <strong>allow</strong> enables users to transfer to this computer. Setting this to <strong>deny</strong> will prevent transfers to this computer. When set to <strong>token</strong>, only transfers initiated with valid tokens will be allowed to transfer to this computer. Token-based transfers are typically employed by web applications such as Faspex and require a Token Encryption Key.</td>
<td>• allow</td>
<td>allow</td>
</tr>
<tr>
<td><strong>Incoming External Provider URL</strong></td>
<td>The value entered should be the URL of the external authorization provider for incoming transfers. The default empty setting disables external authorization. Aspera servers can be configured to check with an external authorization provider. This SOAP authorization mechanism can be useful to organizations requiring custom authorization rules.</td>
<td>HTTP URL</td>
<td>blank</td>
</tr>
<tr>
<td><strong>Incoming External Provider SOAP Action</strong></td>
<td>The SOAP action required by the external authorization provider for incoming transfers. Required if External Authorization is enabled.</td>
<td>text string</td>
<td>blank</td>
</tr>
<tr>
<td><strong>Outgoing Transfers</strong></td>
<td>The default setting of <strong>allow</strong> enables users to transfer from this computer. Setting this to <strong>deny</strong> will prevent transfers from this computer. When set to <strong>token</strong>, only transfers initiated with valid tokens will be allowed to transfer from this computer. Token-based transfers are typically employed by web applications such as Faspex and require a Token Encryption Key.</td>
<td>• allow</td>
<td>allow</td>
</tr>
<tr>
<td><strong>Outgoing External Provider URL</strong></td>
<td>The value entered should be the URL of the external authorization provider for outgoing transfers. The default empty setting disables external authorization. Aspera servers can be configured to check with an external authorization provider. This SOAP authorization mechanism can be useful to organizations requiring custom authorization rules.</td>
<td>HTTP URL</td>
<td>blank</td>
</tr>
<tr>
<td><strong>Outgoing External Provider Soap Action</strong></td>
<td>The SOAP action required by the external authorization provider for outgoing transfers. Required if External Authorization is enabled.</td>
<td>text string</td>
<td>blank</td>
</tr>
<tr>
<td><strong>Token Encryption Cipher</strong></td>
<td>The cipher used to generate encrypted authorization tokens.</td>
<td>• aes-128</td>
<td>aes-128</td>
</tr>
<tr>
<td><strong>Token Encryption Key</strong></td>
<td>This is the secret text phrase that will be used to authorize those transfers configured to require token. Token generation is part of the Aspera SDK. See the Aspera Developer's Network (Token-based Authorization Topic) for more information.</td>
<td>text string</td>
<td>blank</td>
</tr>
<tr>
<td><strong>Token Life (seconds)</strong></td>
<td>Sets token expiration for users of web-based transfer applications.</td>
<td>positive integer</td>
<td>86400 (24 hrs)</td>
</tr>
<tr>
<td><strong>Token Filename Hash</strong></td>
<td>Which algorithm should filenames inside transfer tokens be hashed with. Use MD5 for backward compatibility.</td>
<td>• sha1</td>
<td>sha1</td>
</tr>
</tbody>
</table>
### Server-Side Encryption at Rest (EAR)

**Capabilities**

When files are uploaded from an Aspera client to the server, server-side encryption-at-rest (EAR) saves files on disk in an encrypted state. When downloaded from the server, server-side EAR first decrypts files automatically, and then the transferred files are written to the client's disk in an unencrypted state. Server-side EAR provides the following advantages:

- It protects files against attackers who might gain access to server-side storage. This is important primarily when using NAS storage or cloud storage, where the storage can be accessed directly (and not just through the computer running Aspera Enterprise Server or Point-to-Point).
- It's especially suited for cases where the server is used as a temporary location--for example, when a client uploads a file and another one downloads it.
- Server-side EAR can be used together with client-side EAR. When using both together, content is doubly encrypted.
- Server-side EAR doesn't create an "envelope" as client-side EAR does. The transferred file stays the same size as the original file. The server stores the encryption and various metadata necessary for server-side EAR separately. (By contrast, client-side EAR creates a file envelope containing both the encrypted contents of the file and the encryption metadata, and it also changes the name of the file by adding the file extension .aspera-env.)
- It works with both regular transfers (FASP) and HTTP fallback transfers.

**Limitations and Considerations**

---

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Values</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strong Password Required for Content Encryption</td>
<td>When set to true, require the password for content encryption to contain at least 6 characters, of which at least 1 is non-alphanumeric, at least 1 is a letter, and at least 1 is a digit.</td>
<td>• true • false</td>
<td>false</td>
</tr>
<tr>
<td>Content Protection Secret</td>
<td>Enable server-side encryption-at-rest (EAR) using the specified passphrase. Files uploaded to this server will be encrypted. Files downloaded will be decrypted.</td>
<td>passphrase</td>
<td>(none)</td>
</tr>
</tbody>
</table>
| Content Protection Required | Setting to true requires that content be left encrypted at the destination.  
• Users will be required on upload to enter a password to encrypt the files on the server.  
• Users will be given the option when downloading to decrypt during transfer. | • true • false | false |
| Do encrypted transfers in FIPS-140-2-certified encryption mode | When set to true, ascp will use a FIPS 140-2-certified encryption module. Note: When this feature is enabled, transfer start is delayed while the FIPS module is verified. | • true • false | false |
| Encryption Allowed | Describes the type of transfer encryption accepted by this computer. When set to any the computer allows both encrypted and non-encrypted transfers. When set to none the computer restricts transfers to non-encrypted transfers only. When set to aes-128 the computer restricts transfers to encrypted transfers only. | • any • none • aes-128 | any |
Server-side EAR is not designed for cases where files need to move in an encrypted state between multiple computers. For that purpose, client-side EAR is more suitable: files are encrypted when they first leave the client, then stay encrypted as they move between other computers, and are decrypted when they reach the final destination and the passphrase is available.

Do not mix server-side EAR and non-EAR transfers. Doing so can cause problems for clients by overwriting files when downloading or uploading.

Server-side EAR does not work with multi-session transfers (using `ascp -C` or node API `multi_session` set to greater than 1).

**Configuring Server-side EAR**

1. **Set the docroot in URI format.**

   Server-side EAR requires the storage to have a docroot in URI format. That is, the docroot path must be prefixed with `file://`. Note that the third slash (`/`) does not serve as the root slash for an absolute path. In other words, a docroot of `/home/xfer` would be set as `file:///home/xfer` and a docroot of `C:\Users\xfer` would be set as `file:///C:\Users\xfer`. Set the docroot by modifying `aspera.conf`, found in the following location:

   ```
   /opt/aspera/etc/aspera.conf
   ```

   For each transfer user, add a docroot (or convert an existing docroot to URI format). All transfer users are configured in the `<users>` section. The following is an example of an entry for one user, `asp1`:

   ```
   <user>
   <name>asp1</name>
   ...
   <file_system>
   <access>
   <paths>
   <path>
   <absolute>file:///Users/testing/Public</absolute>
   </path>
   </paths>
   </access>
   </file_system>
   ...
   </user>
   ```

   The docroot can also be set for all users (globally, in the `<default>` section) or for groups.

   🚨 **Important:** The docroot in URI format *cannot* be set from the GUI or with `asconfigurator`.

2. **Set the password.**

   The server-side EAR password can be set for all users (global), per group, or per user in any of the following three ways:

   - by editing `aspera.conf` directly
   - by running `asconfigurator` (which modifies `aspera.conf`)
   - by setting it from the GUI (also modifies `aspera.conf`)

   • **Modifying `aspera.conf` directly or with `asconfigurator`**

   In the server's `aspera.conf` file, enter the following for the default (global) encryption settings:

   ```
   <default>
   <transfer>
   <encryption>
   <content_protection_secret>passphrase</content_protection_secret>
   </encryption>
   </transfer>
   ```
Encryption settings can be configured similarly per group and per user. The following example shows the settings for user aspi:

```xml
<user>
  <name>aspi</name>
  <transfer>
    <encryption>
      <content_protection_secret>passphrase</content_protection_secret>
    </encryption>
  </transfer>
</user>
```

You can also add or modify the above sections in your `aspera.conf` by running `asconfigurator` as follows.

For all users:

```
asconfigurator -x "set_node_data;transfer_encryption_content_protection_secret,passphrase"
```

For user `aspi`:

```
asconfigurator -x "set_user_data;user_name,aspi;transfer_encryption_content_protection_secret,passphrase"
```

You can also add `<content_protection_required>` and/or `<content_protection_strong_pass_required>` to the above. Both are optional and set to false by default. The `<content_protection_required>` option causes server-side EAR to fail if the passphrase is not present. The `<content_protection_strong_pass_required>` option causes server-side EAR to fail if the passphrase is not sufficiently strong (at least six characters, with at least one letter, number, and special character). The following `asconfigurator` command adds both these options for all users (global):

```
asconfigurator -x "set_node_data;transfer_encryption_content_protection_required,true; transfer_encryption_content_protection_strong_pass_required,true"
```

- **Modifying `aspera.conf` from the GUI**

Server-side EAR can also be enabled from the Enterprise Server UI. From the Server Configuration dialog, open the Users tab (and select a user) or the Global tab. Then open the Authorization tab and locate the setting for Content Protection Secret. Check the override box and fill in the passphrase.
If desired, you can also set **Content Protection Required** and/or **Strong Password Required for Content Encryption** to true.

**Bandwidth**

Incoming and outgoing transfer bandwidth and policy settings.

The **Bandwidth** configuration options can be found in the application's **Configuration**, under the **Global**, **Groups**, and **Users** tabs.
The following table lists all configuration options:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Values</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incoming Vlink ID</td>
<td>The value sets Vlink ID for incoming transfers. Vlinks are a mechanism to define aggregate transfer policies. The default setting of 0 disables Vlinks. One Vlink—the virtual equivalent of a network trunk—represents a bandwidth allowance that may be allocated to a node, a group, or a user. Vlink ID is defined in each Vlink created in Aspera Console. Vlink ID is a unique numeric identifier.</td>
<td>pre-defined value</td>
<td>0</td>
</tr>
<tr>
<td>Incoming Target Rate Cap (Kbps)</td>
<td>The value sets the Target Rate Cap for incoming transfers. The Target Rate Cap is the maximum target rate that a transfer can request, in kilobits per second. No transfer may be adjusted above this setting, at any time. The default setting of Unlimited signifies no Target Rate Cap. Clients requesting transfers with initial rates above the Target Rate Cap will be denied.</td>
<td>positive integer</td>
<td>unlimited</td>
</tr>
<tr>
<td>Incoming Target Rate Default (Kbps)</td>
<td>This value represents the initial rate for incoming transfers, in kilobits per second. Users may be able to modify this rate in real time as allowed by the software in use. This setting is not relevant to transfers with a Policy of Fixed.</td>
<td>positive integer</td>
<td>10000</td>
</tr>
</tbody>
</table>
| Incoming Target Rate Lock          | After an incoming transfer is started, its target rate may be modified in real time. The default setting of false gives users the ability to adjust the transfer rate. A setting of true allows users to lock the transfer rate. | • true  
• false         | false    |
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Values</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>prevents real-time modification of the transfer rate.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Incoming Minimum Rate Cap (Kbps)</td>
<td>The value sets the Minimum Rate Cap for incoming transfers. The Minimum Rate Cap is a level specified in kilobits per second, below which an incoming transfer will not slow, despite network congestion or physical network availability. The default value of Unlimited effectively turns off the Minimum Rate Cap.</td>
<td>positive integer</td>
<td>unlimited</td>
</tr>
<tr>
<td></td>
<td>This value represents the initial minimum rate for incoming transfers, in kilobits per second. Users may be able to modify this rate in real time as allowed by the software in use. This setting is not relevant to transfers with a Policy of Fixed.</td>
<td>positive integer</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>After an incoming transfer is started, its minimum rate may be modified in real time. The default setting of false gives users the ability to adjust the transfer's minimum rate. A setting of true prevents real-time modification of the transfer rate. This setting is not relevant to transfers with a Policy of Fixed.</td>
<td>• true</td>
<td>false</td>
</tr>
<tr>
<td></td>
<td>The value chosen sets the default Bandwidth Policy for incoming transfers. The default policy value may be overridden by client applications initiating transfers.</td>
<td>• fixed</td>
<td>fair</td>
</tr>
<tr>
<td></td>
<td>The value chosen sets the allowed Bandwidth Policy for incoming transfers. Aspera transfers use fixed, high, fair and low policies to accommodate network-sharing requirements. When set to any, the server will not deny any transfer based on policy setting. When set to high, transfers with a Policy of high and less aggressive transfer policies (e.g. fair or low) will be permitted. When set to fair, transfers of fair and low will be permitted, while fixed transfers will be denied. When set to low, only transfers with a Bandwidth Policy of low will be allowed.</td>
<td>• fixed</td>
<td>any</td>
</tr>
<tr>
<td></td>
<td>After an incoming transfer is started, its Policy may be modified in real time. The default setting of false gives users the ability to adjust the transfer's Policy. A setting of true prevents real-time modification of the Policy.</td>
<td>• true</td>
<td>false</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
<td>Values</td>
<td>Default</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Outgoing Vlink ID</td>
<td>The value sets Vlink ID for outgoing transfers. Vlinks are a mechanism to define aggregate transfer policies. The default setting of 0 disables Vlinks. One Vlink—the virtual equivalent of a network trunk—represents a bandwidth allowance that may be allocated to a node, a group, or a user. Vlink ID is defined in each Vlink created in Aspera Console. The Vlink ID is a unique numeric identifier.</td>
<td>pre-defined value</td>
<td>0</td>
</tr>
<tr>
<td>Outgoing Target Rate Cap (Kbps)</td>
<td>The value sets the Target Rate Cap for outgoing transfers. The Target Rate Cap is the maximum target rate that a transfer can request, in kilobits per second. No transfer may be adjusted above this setting, at any time. The default setting of Unlimited signifies no Target Rate Cap. Clients requesting transfers with initial rates above the Target Rate Cap will be denied.</td>
<td>positive integer</td>
<td>unlimited</td>
</tr>
<tr>
<td>Outgoing Target Rate Default (Kbps)</td>
<td>This value represents the initial rate for outgoing transfers, in kilobits per second. Users may be able to modify this rate in real time as allowed by the software in use. This setting is not relevant to transfers with a Policy of Fixed.</td>
<td>positive integer</td>
<td>10000</td>
</tr>
<tr>
<td>Outgoing Target Rate Lock</td>
<td>After an outgoing transfer is started, its target rate may be modified in real time. The default setting of false gives users the ability to adjust the transfer rate. A setting of true prevents real-time modification of the transfer rate.</td>
<td>• true</td>
<td>false</td>
</tr>
<tr>
<td>Outgoing Minimum Rate Cap (Kbps)</td>
<td>The value sets the Minimum Rate Cap for outgoing transfers. The Minimum Rate Cap is a level specified in kilobits per second, below which an outgoing transfer will not slow, despite network congestion or physical network availability. The default value of Unlimited effectively turns off the Minimum Rate Cap.</td>
<td>positive integer</td>
<td>unlimited</td>
</tr>
<tr>
<td>Outgoing Minimum Rate Default</td>
<td>This value represents the initial minimum rate for outgoing transfers, in kilobits per second. Users may be able to modify this rate in real time as allowed by the software in use. This setting is not relevant to transfers with a Policy of Fixed.</td>
<td>positive integer</td>
<td>0</td>
</tr>
<tr>
<td>Outgoing Minimum Rate Lock</td>
<td>After an outgoing transfer is started, its minimum rate may be modified in real time. The default setting of false gives users the ability to adjust the transfer's minimum rate. A setting of true prevents real-time modification of the transfer rate. This setting</td>
<td>• true</td>
<td>false</td>
</tr>
</tbody>
</table>
### Field | Description | Values | Default
--- | --- | --- | ---
Outgoing Bandwidth Policy Default | is not relevant to transfers with a Policy of Fixed. | | |
Outgoing Bandwidth Policy Allowed | The value chosen sets the default Bandwidth Policy for outgoing transfers. The default policy value may be overridden by client applications initiating transfers. | • fixed • high • fair (regular) • low | fair
Outgoing Bandwidth Policy Allowed | The value chosen sets the allowed Bandwidth Policy for outgoing transfers. Aspera transfers use fixed, high, fair and low policies to accommodate network-sharing requirements. When set to any, the server will not deny any transfer based on policy setting. When set to high, transfers with a Policy of high and less aggressive transfer policies (e.g. fair or low) will be permitted. When set to fair, transfers of fair and low will be permitted, while fixed transfers will be denied. When set to low, only transfers with a Bandwidth Policy of low will be allowed. | • fixed • high • fair (regular) • low | any
Outgoing Bandwidth Policy Lock | After an outgoing transfer is started, its Policy may be modified in real time. The default setting of false gives users the ability to adjust the transfer’s Policy. A setting of true prevents real-time modification of the Policy. | • true • false | false

### Network

Network IP, port, and socket buffer settings.

The **Network** configuration options can be found in the application's **Configuration**, under the **Global**, **Groups**, and **Users** tabs.
The following table explains all configuration options:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Values</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bind IP Address</td>
<td>Specify an IP address for server-side ascp to bind its UDP connection. If a valid IP address is given, ascp sends and receives UDP packets only on the interface corresponding to that IP address. <strong>Important:</strong> The bind address should only be modified (changed to an address other than 127.0.0.1) if you, as the System Administrator, understand the security ramifications of doing so, and have undertaken precautions to secure the SOAP service.</td>
<td>valid IPv4 address</td>
<td>blank</td>
</tr>
<tr>
<td>Bind UDP Port</td>
<td>Prevent the client-side ascp process from using the specified UDP port.</td>
<td>integer</td>
<td>33001</td>
</tr>
<tr>
<td>Server UDP Port reuse</td>
<td>Allow or disallow different processes to reuse the same UDP port at the server. By default, reuse is allowed (true).</td>
<td>• true</td>
<td>true</td>
</tr>
<tr>
<td>Disable Packet Batching</td>
<td>When set to <strong>true</strong>, send data packets back to back (no sending a batch of packets). This results in smoother data traffic at a cost of higher CPU usage.</td>
<td>• true</td>
<td>false</td>
</tr>
<tr>
<td>Maximum Socket Buffer (bytes)</td>
<td>Upper bound the UDP socket buffer of an ascp session below the input value. The default of 0 will cause the Aspera sender to use its default internal buffer size, which may be different for different operating systems.</td>
<td>positive integer</td>
<td>0</td>
</tr>
<tr>
<td>Minimum Socket Buffer (bytes)</td>
<td>Set the minimum UDP socket buffer size for an ascp session.</td>
<td>positive integer</td>
<td>0</td>
</tr>
<tr>
<td>RTT auto correction</td>
<td>Enable auto correction of base (minimum) RTT measurement. This feature is helpful for maintaining accurate transfer rates in hypervisor-based virtual environments.</td>
<td>• true</td>
<td>false</td>
</tr>
</tbody>
</table>
### Field

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Values</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reverse path congestion inference</td>
<td>Enable reverse path congestion inference, where the default setting of &quot;true&quot; prevents the transfer speed of a session from being adversely affected by congestion in the reverse (non data-sending) transfer direction. This feature is useful for boosting speed in bi-directional transfers.</td>
<td>• true</td>
<td>true</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• false</td>
<td></td>
</tr>
</tbody>
</table>

**File Handling**

File handling settings, such as file block size, overwrite rules, and exclude pattern.

The **File Handling** configuration options can be found in the application's **Configuration**, under the **Global**, **Groups**, and **Users** tabs.
The following table lists all configuration options:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Values</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Create Mode</td>
<td>Specify file creation mode (permissions). If specified, create files with these permissions (for example, 0755), irrespective of File Create Grant Mask and permissions of the file on the source computer. Only takes effect when the server is a non-Windows receiver.</td>
<td>positive integer (octal)</td>
<td>undefined</td>
</tr>
<tr>
<td>File Create Grant Mask</td>
<td>Used to determine mode for newly created files if File Create Mode is not specified. If specified, file modes will be set to their original modes plus the Grant Mask values. Only takes effect when the server is a non-Windows receiver and when File Create Mode is not specified.</td>
<td>positive integer (octal)</td>
<td>644</td>
</tr>
<tr>
<td>Directory Create Mode</td>
<td>Specify directory creation mode (permissions). If specified, create directories with these permissions irrespective of Directory Create Grant Mask and permissions of the directory on the source computer.</td>
<td>positive integer (octal)</td>
<td>undefined</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
<td>Values</td>
<td>Default</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Only takes effect when the server is a non-Windows receiver.</td>
<td>Directory Create Grant Mask Used to determine mode for newly created directories if Directory Create Mode is not specified. If specified, directory modes will be set to their original modes plus the Grant Mask values. Only takes effect when the server is a non-Windows receiver and when Directory Create Mode is not specified.</td>
<td>positive integer (octal)</td>
<td>755</td>
</tr>
<tr>
<td>This is a performance-tuning parameter for an Aspera sender (which only takes effect if the sender is a server). It represents the maximum number of bytes that can be stored within a block as the block is being transferred from the source disk drive to the receiver. The default of zero will cause the Aspera sender to use its default internal buffer size, which may be different for different operating systems.</td>
<td>Read Block Size (bytes)</td>
<td>positive integer, where 500MB or 524,288,000 bytes is the maximum block size.</td>
<td>0</td>
</tr>
<tr>
<td>This is a performance-tuning parameter for an Aspera receiver (which only takes effect if the receiver is a server). It represents the maximum bytes within a block that an ascp receiver can write to disk. The default of zero will cause the Aspera receiver to use its default internal buffer size, which may be different for different operating systems.</td>
<td>Write Block Size (bytes)</td>
<td>positive integer, where 500MB or 524,288,000 bytes is the maximum block size.</td>
<td>0</td>
</tr>
<tr>
<td>This is a performance-tuning parameter for an Aspera sender. It represents the number of threads the Aspera sender will use to read file contents from the source disk drive. It takes effect on both client and server, when acting as a sender. The default of zero will cause the Aspera sender to use its internal default, which may be different for different operating systems.</td>
<td>Number of I/O read threads</td>
<td></td>
<td></td>
</tr>
<tr>
<td>This is a performance-tuning parameter for an Aspera receiver. It represents the number of threads the Aspera receiver will use to write the file contents to the destination disk drive. It takes effect on both client and server, when acting as a receiver. The default of zero causes the Aspera receiver to use its internal default, which may be different for different operating systems.</td>
<td>Number of I/O write threads</td>
<td></td>
<td></td>
</tr>
<tr>
<td>This is a performance tuning parameter for an Aspera receiver. Enable or disable per-file memory caching at the data receiver. File level memory caching improves data write speed on Windows platforms in particular, but will use more memory. We suggest using a file cache on systems that are transferring data at speeds close to the performance of their storage device, and disable it for system with very high concurrency (because memory utilization will grow with the number of concurrent transfers).</td>
<td>Use File Cache</td>
<td>• true • false</td>
<td>true</td>
</tr>
<tr>
<td>This is a performance tuning parameter for an Aspera receiver. This value corresponds to the maximal size</td>
<td>Max File Cache Buffer (bytes)</td>
<td>positive integer</td>
<td>0</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
<td>Values</td>
<td>Default</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>allocated for per-file memory cache (see Use File Cache). Unit is bytes. The default of zero will cause the Aspera receiver to use its internal buffer size, which may be different for different operating systems.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resume Suffix</td>
<td>File name extension for temporary metadata files used for resuming incomplete transfers. Each data file in progress will have a corresponding metadata file with the same name plus the resume suffix specified by the receiver. Metadata files in the source of a directory transfer are skipped if they end with the sender's resume suffix.</td>
<td>text string</td>
<td>.aspx</td>
</tr>
<tr>
<td>Symbolic Link Actions</td>
<td>Actions to be taken upon encountering a symbolic link on the server side. The action to take depends on both the platform and the particular application context. Combinations of choices are logically OR'ed before use. For example, use none alone to mean skip, and shut out other choices; when both follow and follow_wide are present, the latter is recognized.</td>
<td>• none</td>
<td>follow,create</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• create</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• follow</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• follow_wide</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• any</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>combination</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>delimited</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>by commas</td>
<td></td>
</tr>
<tr>
<td>Preserve Attributes</td>
<td>Configure file creation policy. When set to none, do not preserve the timestamp of source files. When set to times, preserve the timestamp of the source files at destination. Note: For Limelight storage, only the preservation of modification time is supported.</td>
<td>• none</td>
<td>blank</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• times</td>
<td></td>
</tr>
<tr>
<td>Overwrite</td>
<td>Overwrite is an Aspera server setting that determines whether Aspera clients are allowed to overwrite files on the server. By default it is set to allow, meaning that clients uploading files to the servers will be allowed to overwrite existing files as long as file permissions allow that action. If set to deny, clients uploading files to the server will not be able to overwrite existing files, regardless of file permissions.</td>
<td>• allow</td>
<td>allow</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• deny</td>
<td></td>
</tr>
<tr>
<td>File Manifest</td>
<td>When set to text a text file &quot;receipt&quot; of all files within each transfer session is generated. If set to disable, no File Manifest is created. The file manifest is a file containing a list of everything that was transferred in a given transfer session. The filename of the File Manifest itself is automatically generated based on the transfer session's unique ID. The location where each manifest is written is specified by the File Manifest Path value. If no File Manifest Path is specified, the file will be generated under the destination path at the receiver, and under the first source path at the sender.</td>
<td>• text</td>
<td>none</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• disable</td>
<td></td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
<td>Values</td>
<td>Default</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------</td>
<td>-----------</td>
</tr>
<tr>
<td>File Manifest Path</td>
<td>Specify the location to store manifest files. Can be an absolute path or a</td>
<td>text string</td>
<td>blank</td>
</tr>
<tr>
<td></td>
<td>path relative to the transfer user's home.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> File manifests can only be stored locally. Thus, if you are using</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>S3, or other non-local storage, you must specify a <em>local</em> manifest path.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>File Manifest Suffix</td>
<td>Specify the suffix of the manifest file during file transfer.</td>
<td>text string</td>
<td>.aspera-</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> File manifests can only be stored locally. Thus, if you are using</td>
<td></td>
<td>inprogress</td>
</tr>
<tr>
<td></td>
<td>S3, or other non-local storage, you must specify a <em>local</em> manifest path.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pre-Calculate Job Size</td>
<td>Configure the policy of calculating total job size before data transfer. If</td>
<td>• any</td>
<td>any</td>
</tr>
<tr>
<td></td>
<td>set to <em>any</em>, follow client configurations (-o PreCalculateJobSize={yes</td>
<td>no}).</td>
<td>• yes</td>
</tr>
<tr>
<td></td>
<td>If set to <em>no</em>, disable calculating job size before transferring. If set to</td>
<td>• no</td>
<td></td>
</tr>
<tr>
<td></td>
<td><em>yes</em>, enable calculating job size before transferring.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>File Exclude Pattern List</td>
<td>Exclude files or directories with the specified pattern in the transfer. Add</td>
<td>text entries</td>
<td>blank</td>
</tr>
<tr>
<td></td>
<td>multiple entries for more exclusion patterns. Two symbols can be used in the</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>setting of patterns:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• &quot;*&quot; (Asterisk) Represents zero to many characters in a string, for example,</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>*.tmp matches .tmp and abcd.tmp.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• &quot;?&quot; (Question Mark) Represents one character, for example, t?p matches tmp</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>but not temp.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> This option applies only when the server is acting as a client.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Servers cannot exclude files or directories uploaded or downloaded by remote</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>clients.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Partial File Name Suffix</td>
<td>Filename extension on the destination computer while the file is being</td>
<td>text string</td>
<td>blank</td>
</tr>
<tr>
<td></td>
<td>transferred. Once the file has been completely transferred, this filename</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>extension is removed.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>If hot folders will be used as the upload destination, the partial filename</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>suffix should be set even if it means setting it to the default value .partial.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Setting it prevents partial files from being downloaded from a hot folder.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> This option only takes effect when it is set on the receiver side.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>File checksum method</td>
<td>The type of checksum to calculate for transferred files. The content of</td>
<td>text string</td>
<td>blank</td>
</tr>
<tr>
<td></td>
<td>transfers can be verified by comparing the checksum value at the destination</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>with the value read at the source. Check the override box and for the</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>effective value, select md5, sha1, or any. For details on configuring and</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>using the checksum feature, see <a href="#">Reporting Checksums</a> on page 48.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
<td>Values</td>
<td>Default</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
<td>---------</td>
</tr>
<tr>
<td>Async Log Directory</td>
<td>An alternative location for the Sync server's log files. If empty, log files go to the default location, or the location specified by the client with -R.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Async Log Level</td>
<td>The amount of detail in the Sync server activity log. Choices are disable, dbg1, and dbg2.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Async Snapdb Directory</td>
<td>An alternative location for the Sync server's snapshot DB files.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**aspera.conf - Authorization**

The configuration options in the `<authorization/>` section of aspera.conf.

This topic shows you how to modify the `<authorization/>` section of aspera.conf.

1. Open aspera.conf.

   ```bash
   /opt/aspera/etc/aspera.conf
   ```

   You can also find the configuration example in this path:

   ```bash
   /opt/aspera/etc/samples/aspera-everything.conf
   ```

2. Add or locate the `<authorization/>` section using a template.

   The following template includes all options:

   ```xml
   <authorization>
   <transfer>
   <in>
   <value>allow</value>       <!-- Incoming Transfer -->
   <external_provider>
   <url>...</url>   <!-- Incoming External Provider URL -->
   <soap>...</soap> <!-- Incoming External Provider SOAP Action -->
   </external_provider>
   </in>
   <out>
   <value>allow</value>       <!-- Outgoing Transfer -->
   <external_provider>
   <url>...</url>   <!-- Outgoing External Provider URL -->
   <soap>...</soap> <!-- Outgoing External Provider SOAP Action -->
   </external_provider>
   </out>
   </transfer>
   <token>
   <encryption_type>aes-128</encryption_type> <!-- Token Encryption Cipher -->
   <encryption_key> </encryption_key>      <!-- Token Encryption Key -->
   <filename_hash> </filename_hash>       <!-- Token Filename Hash -->
   <life_seconds>86400</life_seconds>     <!-- Token Life (seconds) -->
   </token>
   </authorization>
   ```

The following table lists all configuration options:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Values</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incoming Transfers</td>
<td>The default setting of <strong>allow</strong> enables users to transfer to this computer. Setting this to <strong>deny</strong> will prevent transfers to this computer. When set to <strong>token</strong>, only transfers initiated with valid tokens will be allowed to transfer to this computer. Token-based transfers are typically employed by web applications such as Faspex and require a Token Encryption Key.</td>
<td>• allow</td>
<td>allow</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• deny</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• token</td>
<td></td>
</tr>
<tr>
<td>Incoming External Provider URL</td>
<td>The value entered should be the URL of the external authorization provider for incoming transfers. The default empty setting disables external authorization. Aspera servers can be configured to check with an external authorization provider. This SOAP authorization mechanism can be useful to organizations requiring custom authorization rules.</td>
<td>HTTP URL</td>
<td>blank</td>
</tr>
<tr>
<td>Incoming External Provider SOAP Action</td>
<td>The SOAP action required by the external authorization provider for incoming transfers. Required if External Authorization is enabled.</td>
<td>text string</td>
<td>blank</td>
</tr>
<tr>
<td>Outgoing Transfers</td>
<td>The default setting of <strong>allow</strong> enables users to transfer from this computer. Setting this to <strong>deny</strong> will prevent transfers from this computer. When set to <strong>token</strong>, only transfers initiated with valid tokens will be allowed to transfer from this computer. Token-based transfers are typically employed by web applications such as Faspex and require a Token Encryption Key.</td>
<td>• allow</td>
<td>allow</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• deny</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• token</td>
<td></td>
</tr>
<tr>
<td>Outgoing External Provider URL</td>
<td>The value entered should be the URL of the external authorization provider for outgoing transfers. The default empty setting disables external authorization. Aspera servers can be configured to check with an external authorization provider. This SOAP authorization mechanism can be useful to organizations requiring custom authorization rules.</td>
<td>HTTP URL</td>
<td>blank</td>
</tr>
<tr>
<td>Outgoing External Provider SOAP Action</td>
<td>The SOAP action required by the external authorization provider for outgoing transfers. Required if External Authorization is enabled.</td>
<td>text string</td>
<td>blank</td>
</tr>
<tr>
<td>Token Encryption Cipher</td>
<td>The cipher used to generate encrypted authorization tokens.</td>
<td>• aes-128</td>
<td>aes-128</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• aes-192</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• aes-256</td>
<td></td>
</tr>
<tr>
<td>Token Encryption Key</td>
<td>This is the secret text phrase that will be used to authorize those transfers configured to require token. Token generation is part of the Aspera SDK. See the <em>Aspera Developer's Network (Token-based Authorization Topic)</em> for more information.</td>
<td>text string</td>
<td>blank</td>
</tr>
</tbody>
</table>
### General Configuration Reference

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Values</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Token Filename Hash</td>
<td>Which algorithm should filenames inside transfer tokens be hashed with. Use MD5 for backward compatibility.</td>
<td>• sha1</td>
<td>sha1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• MD5</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• sha256</td>
<td></td>
</tr>
<tr>
<td>Token Life (seconds)</td>
<td>Sets token expiration for users of web-based transfer applications.</td>
<td>positive integer</td>
<td>86400 (24 hrs)</td>
</tr>
</tbody>
</table>

4. **Validate aspera.conf.**

When you have finished updating `aspera.conf`, use this command to validate it:

```
$ /opt/aspera/bin/asuserdata -b -v -a
```

---

## aspera.conf - Transfer

The configuration options in `<transfer/>` section of `aspera.conf`

This topic shows you how to modify the `aspera.conf <transfer/>` section.

1. **Open aspera.conf.**

   ```
   /opt/aspera/etc/aspera.conf
   ```

   You can also find the example `aspera.conf` in the `samples` directory:

   ```
   /opt/aspera/etc/samples/aspera-everything.conf
   ```

2. **Add or locate the `<transfer/>` section using a template.**

   The following template includes all options:

   ```xml
   <transfer>
     <in>
       <bandwidth>
         <aggregate>  <!-- Incoming VLink ID -->
       </aggregate>
       <flow>
         <target_rate>  <!-- Incoming Target Rate Cap -->
           <cap>/cap>
           <default>10000</default>
           <lock>false</lock>
         </target_rate>
         <min_rate>  <!-- Incoming Minimum Rate Cap -->
           <cap>/cap>
           <default>/default>
           <lock>false</lock>
         </min_rate>
         <policy>  <!-- Incoming Policy Allowed -->
           <cap>/cap>
           <default>/default>
           <lock>false</lock>
         </policy>
         <priority>  <!-- Incoming Priority Allowed -->
           <cap>/cap>
           <default>/default>
           <lock>false</lock>
         </priority>
         <network_rc>  <!-- Incoming Rate Control Module -->
           <module>/module>
           <tcp_friendly>no</tcp_friendly>  <!-- Incoming TCP Friendly Mode -->
         </network_rc>
       </flow>
     </in>
     <out>  <!-- Outgoing VLink ID -->
   </transfer>
   ```

The following table explains all configuration options:
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Values</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incoming Vlink ID</td>
<td>The value sets Vlink ID for incoming transfers. Vlinks are a mechanism to define aggregate transfer policies. The default setting of 0 disables Vlinks. One Vlink—the virtual equivalent of a network trunk—represents a bandwidth allowance that may be allocated to a node, a group, or a user. Vlink ID is defined in each Vlink created in Aspera Console. Vlink ID is a unique numeric identifier.</td>
<td>pre-defined value</td>
<td>0</td>
</tr>
<tr>
<td>Incoming Target Rate Cap (Kbps)</td>
<td>The value sets the Target Rate Cap for incoming transfers. The Target Rate Cap is the maximum target rate that a transfer can request, in kilobits per second. No transfer may be adjusted above this setting, at any time. The default setting of Unlimited signifies no Target Rate Cap. Clients requesting transfers with initial rates above the Target Rate Cap will be denied.</td>
<td>positive integer</td>
<td>unlimited</td>
</tr>
<tr>
<td>Incoming Target Rate Default (Kbps)</td>
<td>This value represents the initial rate for incoming transfers, in kilobits per second. Users may be able to modify this rate in real time as allowed by the software in use. This setting is not relevant to transfers with a Policy of Fixed.</td>
<td>positive integer</td>
<td>10000</td>
</tr>
<tr>
<td>Incoming Target Rate Lock</td>
<td>After an incoming transfer is started, its target rate may be modified in real time. The default setting of false gives users the ability to adjust the transfer rate. A setting of true prevents real-time modification of the transfer rate.</td>
<td>• true • false</td>
<td>false</td>
</tr>
<tr>
<td>Incoming Minimum Rate Cap (Kbps)</td>
<td>The value sets the Minimum Rate Cap for incoming transfers. The Minimum Rate Cap is a level specified in kilobits per second, below which an incoming transfer will not slow, despite network congestion or physical network availability. The default value of Unlimited effectively turns off the Minimum Rate Cap.</td>
<td>positive integer</td>
<td>unlimited</td>
</tr>
<tr>
<td>Incoming Minimum Rate Default (Kbps)</td>
<td>This value represents the initial minimum rate for incoming transfers, in kilobits per second. Users may be able to modify this rate in real time as allowed by the software in use. This setting is not relevant to transfers with a Policy of Fixed.</td>
<td>positive integer</td>
<td>0</td>
</tr>
<tr>
<td>Incoming Minimum Rate Lock</td>
<td>After an incoming transfer is started, its minimum rate may be modified in real time. The default setting of false gives users the ability to adjust the transfer's minimum rate. A setting of true prevents</td>
<td>• true • false</td>
<td>false</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
<td>Values</td>
<td>Default</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>Field</td>
<td>real-time modification of the transfer rate. This setting is not relevant to transfers with a Policy of Fixed.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| Incoming Bandwidth Policy Allowed   | The value chosen sets the allowed Bandwidth Policy for incoming transfers. Aspera transfers use fixed, high, fair and low policies to accommodate network-sharing requirements. When set to any, the server will not deny any transfer based on policy setting. When set to high, transfers with a Policy of high and less aggressive transfer policies (e.g. fair or low) will be permitted. When set to fair, transfers of fair and low will be permitted, while fixed transfers will be denied. When set to low, only transfers with a Bandwidth Policy of low will be allowed. | • fixed  
• high  
• fair (regular)  
• low | any |
| Incoming Bandwidth Policy Default   | The value chosen sets the default Bandwidth Policy for incoming transfers. The default policy value may be overridden by client applications initiating transfers. | • fixed  
• high  
• fair (regular)  
• low | fair |
| Incoming Bandwidth Policy Lock      | After an incoming transfer is started, its Policy may be modified in real time. The default setting of false gives users the ability to adjust the transfer's Policy. A setting of true prevents real-time modification of the Policy. | • true  
• false | false |
| Incoming Priority Allowed           | The highest priority your client can request. Use the value 0 to unset this option; 1 to allow high priority, 2 to enforce normal priority. | • 0  
• 1  
• 2 | 1 |
| Incoming Priority Default           | The initial priority setting. Use the value 0 to unset this option, 1 to allow high priority; 2 to enforce normal priority | • 0  
• 1  
• 2 | 2 |
| Incoming Priority Lock              | To disallow your clients change the priority, set the value to true | • true  
• false | false |
| Module (for incoming rate control)  | Located within the incoming `<network_rc>` stanza, this hidden setting is meant for advanced users to select an incoming rate control module (which will only be applied at the local "receiver" side). It should only be used with special instructions for debugging. Options include:  
• delay-odp: queue scaling controller  
• delay-adv: advanced rate controller  
• air: FASP air | • delay-odp  
• delay-adv  
• air | blank |
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Values</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP Friendly (for <em>incoming</em> rate control)</td>
<td>Located within the incoming <code>&lt;network_rc&gt;</code> stanza, this hidden setting is <em>meant for advanced users</em> to turn TCP-friendly mode on or off (which will only be applied at the local &quot;receiver&quot; side when the transfer policy is set to <em>fair</em>). <em>It should only be used with special instructions for debugging.</em> If turned on (<em>&quot;yes&quot;</em>), this mode allows an incoming FASP transfer to maintain relative fair bandwidth share with a TCP flow under congestion.</td>
<td>• <em>yes</em> • <em>no</em></td>
<td><em>no</em></td>
</tr>
<tr>
<td>Outgoing Vlink ID</td>
<td>The value sets Vlink ID for outgoing transfers. Vlinks are a mechanism to define aggregate transfer policies. The default setting of 0 disables Vlinks. One Vlink—the virtual equivalent of a network trunk—represents a bandwidth allowance that may be allocated to a node, a group, or a user. Vlink ID is defined in each Vlink created in Aspera Console. The Vlink ID is a unique numeric identifier.</td>
<td>pre-defined value</td>
<td>0</td>
</tr>
<tr>
<td>Outgoing Target Rate Cap (Kbps)</td>
<td>The value sets the Target Rate Cap for outgoing transfers. The Target Rate Cap is the maximum target rate that a transfer can request, in kilobits per second. No transfer may be adjusted above this setting, at any time. The default setting of <em>Unlimited</em> signifies no Target Rate Cap. Clients requesting transfers with initial rates above the Target Rate Cap will be denied.</td>
<td>positive integer</td>
<td><em>unlimited</em></td>
</tr>
<tr>
<td>Outgoing Target Rate Default (Kbps)</td>
<td>This value represents the initial rate for outgoing transfers, in kilobits per second. Users may be able to modify this rate in real time as allowed by the software in use. This setting is not relevant to transfers with a Policy of <em>Fixed</em>.</td>
<td>positive integer</td>
<td>10000</td>
</tr>
<tr>
<td>Outgoing Target Rate Lock</td>
<td>After an outgoing transfer is started, its target rate may be modified in real time. The default setting of <em>false</em> gives users the ability to adjust the transfer rate. A setting of <em>true</em> prevents real-time modification of the transfer rate.</td>
<td>• <em>true</em> • <em>false</em></td>
<td><em>false</em></td>
</tr>
<tr>
<td>Outgoing Minimum Rate Cap (Kbps)</td>
<td>The value sets the Minimum Rate Cap for outgoing transfers. The Minimum Rate Cap is a level specified in kilobits per second, below which an outgoing transfer will not slow, despite network congestion or physical network availability. The default value of <em>Unlimited</em> effectively turns off the Minimum Rate Cap.</td>
<td>positive integer</td>
<td><em>unlimited</em></td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
<td>Values</td>
<td>Default</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>Outgoing Minimum Rate</td>
<td>This value represents the initial minimum rate for outgoing transfers, in kilobits per second. Users may be able to modify this rate in real time as allowed by the software in use. This setting is not relevant to transfers with a Policy of Fixed.</td>
<td>positive integer</td>
<td>0</td>
</tr>
<tr>
<td>Default</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Outgoing Minimum Rate</td>
<td>After an outgoing transfer is started, its minimum rate may be modified in real time. The default setting of <code>false</code> gives users the ability to adjust the transfer's minimum rate. A setting of <code>true</code> prevents real-time modification of the transfer rate. This setting is not relevant to transfers with a Policy of Fixed.</td>
<td>• <code>true</code> • <code>false</code></td>
<td><code>false</code></td>
</tr>
<tr>
<td>Lock</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Outgoing Bandwidth Policy</td>
<td>The value chosen sets the allowed Bandwidth Policy for outgoing transfers. Aspera transfers use fixed, high, fair and low policies to accommodate network-sharing requirements. When set to <code>any</code>, the server will not deny any transfer based on policy setting. When set to <code>high</code>, transfers with a Policy of high and less aggressive transfer policies (e.g. fair or low) will be permitted. When set to <code>fair</code>, transfers of fair and low will be permitted, while fixed transfers will be denied. When set to <code>low</code>, only transfers with a Bandwidth Policy of <code>low</code> will be allowed.</td>
<td>• <code>fixed</code> • <code>high</code> • <code>fair (regular)</code> • <code>low</code></td>
<td><code>any</code></td>
</tr>
<tr>
<td>Allowed</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Outgoing Bandwidth Policy</td>
<td>The value chosen sets the default Bandwidth Policy for outgoing transfers. The default policy value may be overridden by client applications initiating transfers.</td>
<td>• <code>fixed</code> • <code>high</code> • <code>fair (regular)</code> • <code>low</code></td>
<td><code>fair</code></td>
</tr>
<tr>
<td>Default</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Outgoing Bandwidth Policy</td>
<td>After an outgoing transfer is started, its Policy may be modified in real time. The default setting of <code>false</code> gives users the ability to adjust the transfer's Policy. A setting of <code>true</code> prevents real-time modification of the Policy.</td>
<td>• <code>true</code> • <code>false</code></td>
<td><code>false</code></td>
</tr>
<tr>
<td>Lock</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Outgoing Priority Allowed</td>
<td>The highest priority your client can request. Use the value 0 to unset this option; 1 to allow high priority, 2 to enforce normal priority.</td>
<td>• 0 • 1 • 2</td>
<td>1</td>
</tr>
<tr>
<td>Outgoing Priority Default</td>
<td>The initial priority setting. Use the value 0 to unset this option, 1 to allow high priority; 2 to enforce normal priority.</td>
<td>• 0 • 1 • 2</td>
<td>2</td>
</tr>
<tr>
<td>Lock</td>
<td>To disallow your clients change the priority, set the value to <code>true</code></td>
<td>• <code>true</code> • <code>false</code></td>
<td><code>false</code></td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
<td>Values</td>
<td>Default</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------</td>
<td>-----------</td>
</tr>
<tr>
<td>Module (for outgoing rate control)</td>
<td>Located within the outgoing <code>&lt;network_rc&gt;</code> stanza, this hidden setting is <em>meant for advanced users</em> to select an outgoing rate control module (which will only be applied at the local &quot;receiver&quot; side). <em>It should only be used with special instructions for debugging.</em> Options include:  • delay-odp: queue scaling controller  • delay-adv: advanced rate controller  • air: FASP air</td>
<td>• delay-odp  • delay-adv  • air</td>
<td>blank</td>
</tr>
<tr>
<td>TCP Friendly (for outgoing rate control)</td>
<td>Located within the outgoing <code>&lt;network_rc&gt;</code> stanza, this hidden setting is <em>meant for advanced users</em> to turn TCP-friendly mode on or off (which will only be applied at the local &quot;receiver&quot; side when the transfer policy is set to fair). <em>It should only be used with special instructions for debugging.</em> If turned on (&quot;yes&quot;), this mode allows an outgoing FASP transfer to maintain relative fair bandwidth share with a TCP flow under congestion.</td>
<td>• yes        • no</td>
<td>no</td>
</tr>
<tr>
<td>Bind IP Address</td>
<td>Specify an IP address for server-side ascp to bind its UDP connection. If a valid IP address is given, ascp sends and receives UDP packets only on the interface corresponding to that IP address. <strong>Important:</strong> The bind address should only be modified (changed to an address other than 127.0.0.1) if you, as the System Administrator, understand the security ramifications of doing so, and have undertaken precautions to secure the SOAP service.</td>
<td>valid IPv4 address</td>
<td>blank</td>
</tr>
<tr>
<td>Bind UDP Port</td>
<td>Prevent the client-side ascp process from using the specified UDP port.</td>
<td>integer between 1 and 65535</td>
<td>33001</td>
</tr>
<tr>
<td>Disable Packet Batching</td>
<td>When set to true, send data packets back to back (no sending a batch of packets). This results in smoother data traffic at a cost of higher CPU usage.</td>
<td>• true       • false</td>
<td>false</td>
</tr>
<tr>
<td>Batch Size</td>
<td>When set to &quot;0&quot; (default), the system uses a pre-computed batch size. Set this to &quot;1&quot; for high concurrency servers (senders) in order to reduce CPU utilization in aggregate.</td>
<td>Integer</td>
<td>0</td>
</tr>
<tr>
<td>Datagram Size</td>
<td>Sets the datagram size on the server side. If size is set with both -Z (client side)</td>
<td>Integer</td>
<td>1492</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
<td>Values</td>
<td>Default</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
<td>--------</td>
<td>---------</td>
</tr>
<tr>
<td>Maximum Socket Buffer (bytes)</td>
<td>Upper bound the UDP socket buffer of an ascp session below the input value. The default of 0 will cause the Aspera sender to use its default internal buffer size, which may be different for different operating systems.</td>
<td>positive integer</td>
<td>0</td>
</tr>
<tr>
<td>Minimum Socket Buffer (bytes)</td>
<td>Set the minimum UDP socket buffer size for an ascp session.</td>
<td>positive integer</td>
<td>0</td>
</tr>
<tr>
<td>RTT auto correction</td>
<td>Enable auto correction of base (minimum) RTT measurement. This feature is helpful for maintaining accurate transfer rates in hypervisor-based virtual environments.</td>
<td>• true • false</td>
<td>false</td>
</tr>
<tr>
<td>Reverse path congestion inference</td>
<td>Enable reverse path congestion inference, where the default setting of &quot;true&quot; prevents the transfer speed of a session from being adversely affected by congestion in the reverse (non data-sending) transfer direction. This feature is useful for boosting speed in bi-directional transfers.</td>
<td>• true • false</td>
<td>true</td>
</tr>
<tr>
<td>Strong Password Required for Content Encryption</td>
<td>When set to true, require the password for content encryption to contain at least 6 characters, of which at least 1 is non-alphanumeric, at least 1 is a letter, and at least 1 is a digit.</td>
<td>• true • false</td>
<td>false</td>
</tr>
<tr>
<td>Content Protection Required</td>
<td>Setting to true requires that content be left encrypted at the destination. Users will be required on upload to enter a password to encrypt the files on the server. Users will be given the option when downloading to decrypt during transfer.</td>
<td>• true • false</td>
<td>false</td>
</tr>
<tr>
<td>Encryption Allowed</td>
<td>Describes the type of transfer encryption accepted by this computer. When set to any the computer allows both encrypted and non-encrypted transfers. When set to none the computer restricts transfers to non-encrypted transfers only. When set to aes-128 the computer restricts transfers to encrypted transfers only.</td>
<td>• any • none • aes-128</td>
<td>any</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
<td>Values</td>
<td>Default</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
<td>--------</td>
<td>---------</td>
</tr>
<tr>
<td>Do encrypted transfers in FIPS-140-2-certified encryption mode</td>
<td>When set to <strong>true</strong>, ascp will use a FIPS 140-2-certified encryption module. Note: When this feature is enabled, transfer start is delayed while the FIPS module is verified.</td>
<td>• true • false</td>
<td>false</td>
</tr>
</tbody>
</table>

4. **Validate aspera.conf**

When you have finished updating aspera.conf, run the following command to validate it:

```
$ /opt/aspera/bin/asuserdata -b -v -a
```

---

**aspera.conf - File System**

The configuration options in `<file_system/>` section of aspera.conf.

This topic describes how to modify the `<file_system/>` section of aspera.conf in a Terminal.

1. **Open aspera.conf.**

```
/opt/aspera/etc/aspera.conf
```

You can also find the configuration example in this path:

```
/opt/aspera/etc/samples/aspera-everything.conf
```

2. **Add or locate the `<file_system/>` section using a template**

The following template includes all options:

```
<file_system>
<access>
  <paths>
    <path>
      <absolute_peer_ip="ip_address">/path/$(name)</absolute>
      <absolute>/path/$(name)</absolute> <!-- Absolute Path (conditional) -->
      <read_allowed>true</read_allowed> <!-- Read Allowed -->
      <write_allowed>true</write_allowed> <!-- Write Allowed -->
      <dir_allowed>true</dir_allowed> <!-- Browse Allowed -->
    </path>
  </paths>
</access>
<read_block_size>0</read_block_size> <!-- Read Block Size -->
<write_block_size>0</write_block_size> <!-- Write Block Size -->
<use_file_cache>true</use_file_cache> <!-- Use File Cache -->
<max_file_cache_buffer>0</max_file_cache_buffer> <!-- Max File Cache Buffer -->
<resume_suffix>.aspx</resume_suffix> <!-- Resume Suffix -->
<preserve_attributes/></preserve_attributes> <!-- Preserve Attributes -->
<overwrite>allow</overwrite> <!-- Overwrite -->
<file_manifest>disable</file_manifest> <!-- File Manifest -->
<file_manifest_path>path</file_manifest_path> <!-- File Manifest Path -->
<pre_calculate_job_size>any</pre_calculate_job_size> <!-- Pre-Calculate Job Size -->
<storage rc>
  <adaptive>true</adaptive> <!-- Storage Rate Control -->
</storage rc>
<file_create_mode> </file_create_mode> <!-- File Create Mode -->
<file_create_grant_mask>644</file_create_grant_mask> <!-- File Create Grant Mask -->
<directory_create_mode> </directory_create_mode> <!-- Directory Create Mode -->
<directory_create_grant_mask>755</directory_create_grant_mask> <!-- Directory Create Grant Mask -->
<excludes> <!-- Exclude Pattern -->
  <exclude></exclude>
  <exclude></exclude>
  ...
</excludes>
<partial_file_suffix>.partial</partial_file_suffix> <!-- Partial File Suffix -->
```

The following table lists all configuration options:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Values</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Absolute Path</td>
<td>The Absolute Path is a path to the docroot, the area of the file system that is accessible to Aspera users. The default empty value gives users access to the entire file system. In aspera.conf, you can set multiple docroots and make them conditional based on the IP address from which the connection is made. To do so, set the absolute path as follows:</td>
<td>file path or Amazon S3 URI</td>
<td>blank</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;absolute</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>peer_ip=&quot;ip_address&quot;&gt;path&lt;/absolute</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** You may also specify an Amazon S3 docroot in the following URI format: s3://MY_ACCESS_ID:MY_SECRET_KEY@s3.amazonaws.com/my_bucket/my_path (where each of the MY_ACCESS_ID, MY_SECRET_KEY and my_bucket/my_path parts must be url_encoded).

S3 server side options are specified through an additional query part in the URI, as shown below.

s3://MY_ACCESS_ID:MY_SECRET_KEY@s3.amazonaws.com/my_bucket/my_path?storage-class=REDUCED_REDUNDANCY&server-side-encryption=AES256

Valid values are as follows:

- **For storage-class**: STANDARD (default if not specified) or REDUCED_REDUNDANCY.
- **For server-side-encryption**: AES256 is the only valid value.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Values</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read Allowed</td>
<td>Setting this to true allows users to transfer from the designated area of the file system as specified by the Absolute Path value.</td>
<td>true, false</td>
<td>blank</td>
</tr>
<tr>
<td>Write Allowed</td>
<td>Setting this to true allows users to transfer to the designated area of the file system as specified by the Absolute Path value.</td>
<td>true, false</td>
<td>blank</td>
</tr>
<tr>
<td>Browse Allowed</td>
<td>Setting this to true allows users to browse the directory.</td>
<td>true, false</td>
<td>blank</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
<td>Values</td>
<td>Default</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>Read Block Size (bytes)</td>
<td>This is a performance-tuning parameter for an Aspera sender (which only takes effect if the <em>sender</em> is a server). It represents the maximum number of bytes that can be stored within a block as the block is being transferred from the source disk drive to the receiver. The default of zero will cause the Aspera sender to use its default internal buffer size, which may be different for different operating systems.</td>
<td>positive integer, where 500MB or 524,288,000 bytes is the maximum block size.</td>
<td>0</td>
</tr>
<tr>
<td>Write Block Size (bytes)</td>
<td>This is a performance-tuning parameter for an Aspera receiver (which only takes effect if the <em>receiver</em> is a server). It represents the maximum bytes within a block that an ascp receiver can write to disk. The default of zero will cause the Aspera receiver to use its default internal buffer size, which may be different for different operating systems.</td>
<td>positive integer, where 500MB or 524,288,000 bytes is the maximum block size.</td>
<td>0</td>
</tr>
<tr>
<td>Use File Cache</td>
<td>This is a performance tuning parameter for an Aspera receiver. Enable or disable per-file memory caching at the data receiver. File level memory caching improves data write speed on Windows platforms in particular, but will use more memory. We suggest using a file cache on systems that are transferring data at speeds close to the performance of their storage device, and disable it for system with very high concurrency (because memory utilization will grow with the number of concurrent transfers).</td>
<td>• true • false</td>
<td>true</td>
</tr>
<tr>
<td>Max File Cache Buffer</td>
<td>This is a performance tuning parameter for an Aspera receiver. This value corresponds to the maximal size allocated for per-file memory cache (see Use File Cache). Unit is bytes. The default of zero will cause the Aspera receiver to use its internal buffer size, which may be different for different operating systems.</td>
<td>positive integer</td>
<td>0</td>
</tr>
<tr>
<td>Resume Suffix</td>
<td>File name extension for temporary metadata files used for resuming incomplete transfers. Each data file in progress will have a corresponding metadata file with the same name plus the resume suffix specified by the receiver. Metadata files in the source of a directory transfer are skipped if they end with the sender's resume suffix.</td>
<td>text string</td>
<td>.aspx</td>
</tr>
<tr>
<td>Preserve Attributes</td>
<td>Configure file creation policy. When set to <em>none</em>, do not preserve the timestamp of source files. When set to <em>times</em>, preserve the timestamp of the source files at destination.</td>
<td>• none • times</td>
<td>blank</td>
</tr>
<tr>
<td>Overwrite</td>
<td>Overwrite is an Aspera server setting that determines whether Aspera clients are allowed to overwrite files on the server. By default it is set to <em>allow</em>, meaning that clients uploading files to the servers will be allowed to overwrite existing files as long</td>
<td>• allow • deny</td>
<td>allow</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
<td>Values</td>
<td>Default</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------------</td>
<td>---------</td>
</tr>
<tr>
<td></td>
<td>as file permissions allow that action. If set to deny, clients uploading files to the server will not be able to overwrite existing files, regardless of file permissions.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>File Manifest</td>
<td>When set to text a text file &quot;receipt&quot; of all files within each transfer session is generated. If set to disable, no File Manifest is created. The file manifest is a file containing a list of everything that was transferred in a given transfer session. The filename of the File Manifest itself is automatically generated based on the transfer session's unique ID. The location where each manifest is written is specified by the File Manifest Path value. If no File Manifest Path is specified, the file will be generated under the destination path at the receiver, and under the first source path at the sender.</td>
<td>• text</td>
<td>none</td>
</tr>
<tr>
<td></td>
<td>• disable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>File Manifest Path</td>
<td>Specify the location to store manifest files. Can be an absolute path or a path relative to the transfer user's home.</td>
<td>text string</td>
<td>blank</td>
</tr>
<tr>
<td></td>
<td>Note: File manifests can only be stored locally. Thus, if you are using S3, or other non-local storage, you must specify a local manifest path.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pre-Calculate Job Size</td>
<td>Configure the policy of calculating total job size before data transfer. If set to any, follow client configurations (-o PreCalculateJobSize={yes</td>
<td>no}). If set to no, disable calculating job size before transferring. If set to yes, enable calculating job size before transferring.</td>
<td>• any</td>
</tr>
<tr>
<td></td>
<td>• yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• no</td>
<td></td>
<td></td>
</tr>
<tr>
<td>File Create Mode</td>
<td>Specify file creation mode (permissions). If specified, create files with these permissions (for example, 0755), irrespective of File Create Grant Mask and permissions of the file on the source computer. Only takes effect when the server is a non-Windows receiver.</td>
<td>positive integer (octal)</td>
<td>undefined</td>
</tr>
<tr>
<td>File Create Grant Mask</td>
<td>Used to determine mode for newly created files if File Create Mode is not specified. If specified, file modes will be set to their original modes plus the Grant Mask values. Only takes effect when the server is a non-Windows receiver and when File Create Mode is not specified.</td>
<td>positive integer (octal)</td>
<td>644</td>
</tr>
<tr>
<td>Directory Create Mode</td>
<td>Specify directory creation mode (permissions). If specified, create directories with these permissions irrespective of Directory Create Grant Mask and permissions of the directory on the source computer. Only takes effect when the server is a non-Windows receiver.</td>
<td>positive integer (octal)</td>
<td>undefined</td>
</tr>
<tr>
<td>Directory Create Grant Mask</td>
<td>Used to determine mode for newly created directories if Directory Create Mode is not specified. If specified, directory modes will be set to their original modes plus the Grant Mask values. Only takes effect when the</td>
<td>positive integer (octal)</td>
<td>755</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
<td>Values</td>
<td>Default</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
<td>--------</td>
<td>---------</td>
</tr>
<tr>
<td>server is a non-Windows receiver and when Directory Create Mode is not specified.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| File Exclude Pattern List | Exclude files or directories with the specified pattern in the transfer. Add multiple entries for more exclusion patterns. Two symbols can be used in the setting of patterns:  
  • "*" (Asterisk) Represents zero to many characters in a string, for example, *.tmp matches .tmp and abcde.tmp.  
  • "?" (Question Mark) Represents one character, for example, t?p matches tmp but not temp. 
This option applies only when the server is acting as a client. Servers cannot exclude files or directories uploaded or downloaded by remote clients. | text entries | blank |
| Partial File Name Suffix | Filename extension on the destination computer while the file is being transferred. Once the file has been completely transferred, this filename extension is removed.  
If hot folders will be used as the upload destination, the partial filename suffix should be set even if it means setting it to the default value .partial. Setting it prevents partial files from being downloaded from a hot folder.  

⚠️ Note: This option only takes effect when it is set on the receiver side. | text string | blank |

4. Validate aspera.conf.

When you have finished updating aspera.conf, run the following command to validate it:

```
$ /opt/aspera/bin/asuserdata -b -v -a
```
Global Transfer Settings

The system-wide and default FASP transfer settings for your computer.

Global Bandwidth Settings

Allocate the global bandwidth for FASP file transfers.

Aspera’s FASP transport has no theoretical throughput limit. Other than the network capacity, the transfer speed may be limited by rate settings and resources of the computers. This topic describes how to optimize the transfer rate by setting up the global rate settings.

To set global FASP bandwidth, bring up the application and select Tools > Global Preferences. Global bandwidth can be set by administrators only.

In the Global Preferences dialog select Transfers, and enter the download and upload bandwidth values in the System-Wide Settings field and click the checkboxes to enable the settings.
<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>System-Wide Settings</td>
<td>The aggregated bandwidth cap for all FASP transfers on this computer. For more advanced bandwidth settings, see <em>Bandwidth</em> on page 72.</td>
</tr>
<tr>
<td>Default Target Rate</td>
<td>The initial download and upload rates for all transfers.</td>
</tr>
<tr>
<td>Maximum Active Transfers</td>
<td>The maximum number of concurrent upload transfers and download transfers.</td>
</tr>
</tbody>
</table>

**Note:**

When setting the global bandwidth, the application is in fact creating virtual links (Vlink) and applying them to the default transfer settings. For more information about Vlinks, see *Setting Up Virtual Links* on page 100.

To create global bandwidth using the command line, open the `aspera.conf` (/opt/aspera/etc/aspera.conf) with a text editor. The following example sets the global bandwidth with these values:

<table>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upload bandwidth limit (outgoing):</td>
<td>88 Mbps (88000 Kbps)</td>
</tr>
<tr>
<td>Download bandwidth limit (incoming):</td>
<td>99 Mbps (99000 Kbps)</td>
</tr>
</tbody>
</table>

```xml
<?xml version='1.0' encoding='UTF-8'?>
<CONF version="2">
  ...
  <trunks>
    <trunk> <!-- Create a Vlink with 88000 Kbps bandwidth cap. -->
      <id>108</id>  <!-- ID: 108 -->
      <capacity><value>88000</value></capacity>
      <on>true</on>
    </trunk>
    <trunk> <!-- Create a Vlink with 99000 Kbps bandwidth cap. -->
      <id>109</id>  <!-- ID: 109 -->
      <capacity><value>99000</value></capacity>
      <on>true</on>
    </trunk>
  </trunks>
  <default>  <!-- Global settings.-->
    <transfer>
      <out> <!-- Use Vlink ID: 108 for global outgoing bandwidth.-->
        <bandwidth><aggregate><trunk_id>108</trunk_id></aggregate></bandwidth>
      </out>
      <in> <!-- Use Vlink ID: 109 for global incoming bandwidth.-->
        <bandwidth><aggregate><trunk_id>109</trunk_id></aggregate></bandwidth>
      </in>
    </transfer>
  </default>
</CONF>
```

The global settings for download and upload bandwidth limits cannot be reset by non-admin users. However, users can view the global limit from the *My Preferences > Transfers* dialog. They can also adjust the default target rate and maximum number of active transfers.

*My Preferences* can be opened from *Tools > Preferences* or from the *Preferences* button in the upper-right corner of the application window.
Setting Up Virtual Links

Create and apply the aggregate bandwidth cap.

Virtual link (Vlink) is a feature that allows "virtual" bandwidth caps. Transfer sessions assigned to the same "virtual" link conform to the aggregate bandwidth cap and attain an equal share of it. This section first shows you how to set up Vlinks, then explains how to apply it to computers or users.

Follow these steps to configure Vlinks:

1. Create Vlinks (GUI)

   Note:
   
   Skip this step if you are setting it up in a Terminal.

   To configure Vlinks, execute `aserascp` in a Terminal (as a root user) to launch the application, and click **Configuration**. Select **Vlinks** tab in the left panel.

   Click + to add a new Vlink entry; enter a number between 1 and 255 and click **OK**.

   ![Vlink Configuration](image)

   Here is a list of all Vlink configuration options:

<table>
<thead>
<tr>
<th>#</th>
<th>Field</th>
<th>Description</th>
<th>Values</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Vlink Name</td>
<td>The Vlink name. This value has no impact on actual bandwidth capping.</td>
<td>text string</td>
<td>blank</td>
</tr>
<tr>
<td>2</td>
<td>On</td>
<td>Select <strong>true</strong> to activate this Vlink; select <strong>false</strong> to deactivate it.</td>
<td>• true • false</td>
<td>false</td>
</tr>
<tr>
<td>3</td>
<td>Capacity (kbps)</td>
<td>This value reflects the virtual bandwidth cap in Kbps. When applying this Vlink to a transfer (e.g. Default outgoing), the transfer's bandwidth will be restricted by this value.</td>
<td>positive integer in Kbps</td>
<td>50000</td>
</tr>
</tbody>
</table>

2. Apply a Vlink to a transfer (GUI)

   Note:
   
   Skip this step if you are setting it up in a Terminal

   You can assign a Vlink to global, user, or group settings. This example assigns a Vlink to a user's incoming transfer session.

   Bring up the **Configuration** window and select the **Users** tab, select the user to apply Vlink. In the right panel, select the **Bandwidth** tab, check the option **Incoming Vlink ID** and select the Vlink to apply (choose ID from drop-down list):
3. Create Vlinks in `aspera.conf` (Terminal)

To create Vlinks in a Terminal, open `aspera.conf` with a text editor:

```
/opt/aspera/etc/aspera.conf
```

You can refer to the configuration example:

```
/opt/aspera/etc/samples/aspera-everything.conf
```

Locate or create the section `<trunks>...</trunks>`. For each vlink, add a `<trunk>...</trunk>`:

```
<CONF version="2">
...
<trunks>
<trunk>
  <id>108</id>                   <!-- Vlink ID -->
  <name>50Mbps cap</name>        <!-- Vlink Name -->
  <capacity>
    <value>50000</value>        <!-- Capacity -->
  </capacity>
  <on>true</on>                  <!-- On -->
  <mcast_port>55001</mcast_port> <!-- Multicast Port -->
  <mcast_ttl></mcast_ttl>        <!-- Multicast TTL -->
</trunk>
</trunks>
</CONF>
```

Here is a description of the Vlink tags:

<table>
<thead>
<tr>
<th>#</th>
<th>Tag</th>
<th>Description</th>
<th>Values</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Vlink ID</td>
<td>The Vlink ID. Sessions assigned with the same trunk ID share the same bandwidth cap.</td>
<td>positive integer between 1 and 255.</td>
<td>N/A</td>
</tr>
<tr>
<td>2</td>
<td>Vlink Name</td>
<td>The Vlink name. This value has no impact on actual bandwidth capping.</td>
<td>text string</td>
<td>blank</td>
</tr>
<tr>
<td>3</td>
<td>Capacity</td>
<td>This value reflects the virtual bandwidth cap in Kbps. When applying this Vlink to a transfer (e.g. Default outgoing), the transfer's bandwidth will be restricted by this value.</td>
<td>positive integer in Kbps</td>
<td>50000</td>
</tr>
<tr>
<td>4</td>
<td>On</td>
<td>Select true to activate this Vlink; select false to deactivate it.</td>
<td>true/false</td>
<td>false</td>
</tr>
<tr>
<td>5</td>
<td>Multicast Port</td>
<td>This sets the UDP port through which virtual link sends and receives multicast communication messages. Sessions</td>
<td>positive integer between 1 and 65535</td>
<td>55001</td>
</tr>
<tr>
<td>#</td>
<td>Tag</td>
<td>Description</td>
<td>Values</td>
<td>Default</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td></td>
<td></td>
<td>sharing the same virtual bandwidth cap needs to have the same port number. To avoid port conflicts, it is recommended to use the default UDP port 55001. Do NOT set the port number to the same one used by FASP data transfer (33001).</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Multicast TTL</td>
<td>This sets the Time-to-Live (TTL) field in the IP header for Vlink multicast packets.</td>
<td>positive integer between 1 and 255</td>
<td>blank</td>
</tr>
</tbody>
</table>

4. Apply a Vlink to a transfer (Terminal)

You can assign a Vlink to a global, a user, or a group settings in `aspera.conf`.

In this example, assuming we have created three vlinks: 108, 109, and 110, apply these vlinks to the outgoing bandwidth globally or to specific users:

```xml
<CONF version="2">
  ...
  <default>
    <transfer>
      <out>
        <bandwidth><aggregate>
          <trunk_id>108</trunk_id> <!-- Vlink #108 for the default outgoing sessions. -->
        </aggregate></bandwidth>
      </out>
      <in>...
      </in>
    </out>
    <transfer>
      <in>...
      </in>
    </transfer>
  </default>
  <aaa><realms><realm>
    <users>
      <user>
        <name>aspera_user_1</name>
        <transfer>
          <out>
            <bandwidth><aggregate>
              <trunk_id>109</trunk_id> <!-- Vlink #109 to the user aspera_user_1's outgoing sessions. -->
            </aggregate></bandwidth>
          </out>
          <in>...
          </in>
        </transfer>
      </user>
    </users>
    </realm></realms></aaa>
  </CONF>
```

**Important:** If you have a local firewall on your server (for example, Windows firewall, Linux iptables, or Mac ipfw), you will need to allow the Vlink UDP port (55001, by default) for multicast traffic.

**Linux iptables Example:**

```
$ iptables -A INPUT -p udp --dport 55001 -j ACCEPT
```
Transfer Server Configuration

Set up the transfer server and more global/default settings.

Note: To configure the transfer server, you must run the application with admin or root privileges in order to enable the Configuration screen.

To configure IBM Aspera Enterprise Server, in the application (asperascp) click Configuration.

To configure the computer's Aspera Central transfer server, click Global tab in the left panel and select the Transfer Server.

To configure the Aspera Central transfer server in a Terminal, open aspera.conf with a text editor (/opt/aspera/etc/aspera.conf), locate or create the transfer server's section <central_server>...

<CONF version="2">
  ...
  <central_server>
    <address>127.0.0.1</address>                     <!-- Address -->
    <port>40001</port>                               <!-- Port -->
    <persistent_store>enable</persistent_store>     <!-- Persistent store -->
    <files_per_session>1000</files_per_session>     <!-- Files per session -->
    <persistent_store_path>blank</persistent_store_path> <!-- Persistent store path -->
    <persistent_store_max_age>86400</persistent_store_max_age> <!-- Max Age (sec) -->
    <persistent_store_on_error>ignore</persistent_store_on_error> <!-- Ignore on error -->
  </central_server>
  ...
</CONF>
The Aspera Central transfer server's configuration options:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Values</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address</td>
<td>This is the network interface address on which the transfer server listens. The default value 127.0.0.1 enables the transfer server to accept transfer requests from the local computer; The value 0.0.0.0 allows the transfer server to accept requests on all network interfaces for this node. Alternatively, a specific network interface address may be specified.</td>
<td>Valid IPv4 address</td>
<td>127.0.0.1</td>
</tr>
<tr>
<td>Port</td>
<td>The port at which the transfer server accepts transfer requests.</td>
<td>Positive integer between 1 and 65535</td>
<td>40001</td>
</tr>
<tr>
<td>Persistent Storage</td>
<td>Retain data that is stored in the database between reboots of Aspera Central.</td>
<td>• Enable</td>
<td>Enable</td>
</tr>
<tr>
<td></td>
<td>Path to store data between reboots of Aspera Central. If the path is currently a directory, then a file is created with the default name central-store.db. Otherwise, the file will be named as specified in the path.</td>
<td>Valid system path</td>
<td>/opt/aspera/var/ (assuming the product is installed in the default location)</td>
</tr>
<tr>
<td>Files per session</td>
<td>The maximum number of files that can be retained for persistent storage.</td>
<td>Positive integer</td>
<td>1000</td>
</tr>
<tr>
<td>Persistent Storage</td>
<td>Path to store data between reboots of Aspera Central. If the path is currently a directory, then a file is created with the default name central-store.db. Otherwise, the file will be named as specified in the path.</td>
<td>Valid system path</td>
<td>/opt/aspera/var/ (assuming the product is installed in the default location)</td>
</tr>
<tr>
<td>Maximum age (Seconds)</td>
<td>Maximum allowable age (in seconds) of data to be retained in the database.</td>
<td>Positive integer</td>
<td>86400</td>
</tr>
<tr>
<td>Exit Central on storage error</td>
<td>Terminate the Aspera Central server if an error writing to the database occurs.</td>
<td>• Ignore</td>
<td>Ignore</td>
</tr>
<tr>
<td></td>
<td>Compact database on startup Enable or disable compacting (vacuuming) the database when the transfer server starts.</td>
<td>• Enable</td>
<td>Enable</td>
</tr>
</tbody>
</table>

For the general configuration options (Authorization, Bandwidth, Network, File Handling, and Docroot), see the following sections:

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Authorization</strong></td>
<td>Connection permissions, token key, and encryption requirements.</td>
</tr>
<tr>
<td><strong>Bandwidth</strong></td>
<td>Incoming and outgoing transfer bandwidth and policy settings.</td>
</tr>
<tr>
<td><strong>Network</strong></td>
<td>Network IP, port, and socket buffer settings.</td>
</tr>
<tr>
<td><strong>File Handling</strong></td>
<td>File handling settings, such as file block size, overwrite rules, and exclude pattern.</td>
</tr>
<tr>
<td><strong>Document Root</strong></td>
<td>The document root settings.</td>
</tr>
</tbody>
</table>
If you are configuring the Enterprise Server from the command line, see the following sections:

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>aspera.conf - Authorization</code> on page 83</td>
<td>Connection permissions, token key, and encryption requirements.</td>
</tr>
<tr>
<td><code>aspera.conf - Transfer</code> on page 85</td>
<td>Incoming and outgoing transfer bandwidth and policy settings.</td>
</tr>
<tr>
<td><code>aspera.conf - File System</code> on page 93</td>
<td>Docroot, file and directory creation, access permissions, block sizes, and so on.</td>
</tr>
</tbody>
</table>

If you have modified these settings from the command line, run the following command to restart Aspera Central:

```
$ /etc/init.d/asperacentral restart
```
Configuring for Other Aspera Products

Configuring for Faspex

The steps below describe configuring IBM Aspera Enterprise Server as the transfer server for IBM Aspera Faspex.

1. Install Enterprise/Connect Server.

   If you haven't already, follow the steps in Standard Installation on page 6 to install Enterprise Server (the transfer server).

   The transfer server can be set up in either of the following configurations:

   • locally, on the same host as Faspex
   • remotely, on a separate host

   Note: For a local setup, most configuration is taken care of automatically when Faspex is installed in a later step. For this reason, Enterprise Server/Connect Server should be installed first.

   All steps must be performed as root.

2. (LOCAL SETUP ONLY) Check aspera.conf settings and adjust if necessary.

   In the aspera.conf file (/opt/aspera/etc/aspera.conf) check the following:

   • Look for <persistent_store> in the <central_server> section, and be sure that it is set to enable (default value). This setting allows the retention of historical transfer data used by the stats collector.
   • Look for the <dir_allowed> setting for the faspex user, and ensure that it's set to true.

   If you change settings, you must restart asperacentral and asperanoded.

   # /etc/init.d/asperanoded restart
   # /etc/init.d/asperacentral restart

   Note:
   If you are installing Enterprise Server locally (on the same machine as Faspex), continue by installing Faspex as described in the Aspera Faspex Admin Guide.
   If you are setting up Enterprise Server as a remote transfer server node, continue with the steps below.

3. Create the system user on the transfer-server host.

   The system user authenticates the actual ascp transfer and must be an operating system account. Run the following commands to: (1) create the group faspex; (2) create the system user faspex; and (3) configure it to use the Aspera secure shell, aspshell.

   # /usr/sbin/groupadd -r faspex
   # /usr/sbin/useradd -r faspex -s /bin/aspshell-r -g faspex

4. Create and configure the Faspex packages directory.

   Run the following commands to create the packages directory /home/faspex/faspex_packages:

   # mkdir -p /home/faspex/faspex_packages
   # chown -R faspex:faspex /home/faspex/faspex_packages

5. Add the faspex user to Enterprise/Connect Server.

   Launch the desktop application and click Configuration.
In Server Configuration, select the Users tab. Then click the + button.

In the Add User dialog that appears, fill in the name "faspex" and click OK; faspex is then added to the user list.

To specify a docroot, make sure faspex is selected in the user list, and click the Docroot tab in the right panel. For the Absolute Path setting, check the Override box, and under Effective Value fill in /Users/faspex/faspex_packages. For the read, write, and browse settings, check the Override boxes and select true.

You can also add and configure the faspex user for Enterprise Server by modifying aspera.conf, instead of using the application GUI. For details, see Setting Up Users on page 55.

6. Modify aspera.conf.

The aspera.conf file is found in the following location:

```
/opt/aspera/etc/aspera.conf
```

Below is a typical aspera.conf file. Yours may differ, particularly if you have installed other Aspera products. Copy any absent portions from the example below. Modify the following settings, as necessary:

- Add the Faspex package directory as a docroot. In the file below, look for the <absolute> tag to see how the docroot has been defined in this installation, and adjust yours accordingly.
• Look for the `<server_name>` tag, and ensure that `server_ip_or_name` has been replaced with the name or IP address of your server.

• Look for `<persistent_store>` in the `<central_server>` section, and be sure that it is set to `enable` (the default value).

• Look for the `<dir_allowed>` setting for the faspex user, and ensure that it's set to `true`.

```xml
<?xml version='1.0' encoding='UTF-8'?>
<CONF version="2">
  <central_server>
    <address>127.0.0.1</address>
    <port>40001</port>
    <compact_on_startup>enable</compact_on_startup>
    <persistent_store>enable</persistent_store>
    <persistent_store_on_error>ignore</persistent_store_on_error>
    <persistent_store_max_age>86400</persistent_store_max_age>
    <event_buffer_overrun>block</event_buffer_overrun>
  </central_server>
  <default>
    <file_system>
      <pre_calculate_job_size>yes</pre_calculate_job_size>
    </file_system>
  </default>
  <aaa>
    <realms>
      <realm>
        <user>
          <name>faspex</name>
          <file_system>
            <access>
              <paths>
                <path>
                  <absolute>./home/faspex/faspex_packages</absolute>
                  <show_as>/</show_as>
                  <dir_allowed>true</dir_allowed>
                </path>
              </paths>
            </access>
            <directory_create_mode>770</directory_create_mode>
            <file_create_mode>660</file_create_mode>
          </file_system>
          <authorization>
            <transfer>
              <in>
                <value>token</value>
              </in>
              <out>
                <value>token</value>
              </out>
            </transfer>
            <token>
              <encryption_key>af208360-dbdc-4033-a35b-2370941f37e9</encryption_key>
            </token>
          </authorization>
        </user>
      </realm>
    </realms>
  </aaa>
  <http_server>
```
<http_port>8080</http_port>
<enable_http>1</enable_http>
<https_port>8443</https_port>
<enable_https>1</enable_https>
</http_server>
<server>
  <server_name>server_ip_or_name</server_name>
</server>
</CONF>

After modifying aspera.conf, restart the asperacentral and asperanoded services.

# /etc/init.d/asperacentral restart
# /etc/init.d/asperanoded restart

7. Verify that you have a valid transfer server license installed.

Verify that the transfer server has a valid Faspex-enabled license for Enterprise Server. To check this from the command line, run `ascp -A` and review the enabled settings list. For example:

```
Enabled settings: connect, mobile, cargo, node, proxy, http_fallback_server, group_configuration, shared_endpoints, desktop_gui
```

If the list includes **connect** and **http_fallback_server**, you have a Faspex-enabled server license.

You can also check the license from the Enterprise Server desktop client GUI. The License dialog (Tools > License) includes the fields **Connect Clients Enabled** and **Http Fallback Server Enabled**. If both are set to Yes, you have a Faspex-enabled license.

Because this Faspex configuration uses Enterprise Server as a remote transfer service, it requires the Aspera Node API. For this reason, whenever you update your Enterprise Server license (see Updating the Product License on page 17), you must reload the asperanoded service afterwards. Reload the asperanoded service by running `asnodeadmin`, found in the following location:

```
# /opt/aspera/bin/asnodeadmin --reload
```

8. Set up the node user.

Set up the node user and associate it with the faspex user by running the asnodeadmin command, as in the following example--where node-admin is the node user, s3cur3_p433 is the node user's password, and faspex is the system user. Then run asnodeadmin again to reload asperanoded.

```
# /opt/aspera/bin/asnodeadmin -a -u node-admin -p s3cur3_p433 -x faspex
# /opt/aspera/bin/asnodeadmin --reload
```

9. Install the Connect key.

First, locate your Connect key:

```
/opt/aspera/var/aspera_id_dsa.pub
```

Then, create a .ssh folder (if it does not already exist) in the faspex user's home directory:

```
# mkdir -p /home/faspex/.ssh
```
Run the following commands to create the keyfile `authorized_keys` (if it does not already exist), and append the key text to it:

```
# cat /opt/aspera/var/aspera_id_dsa.pub >> /home/faspex/.ssh/authorized_keys
```

Run the following commands to change the key directory and keyfile's ownership to the `faspex` user, to allow access by the `faspex` group, and to set permission bits:

```
# chown -R faspex:faspex /home/faspex/.ssh/authorized_keys
# chmod 600 /home/faspex/.ssh/authorized_keys
# chmod 700 /home/faspex
# chmod 700 /home/faspex/.ssh
```

10. Ensure the firewall is set up correctly on your transfer server

For details, see `Configuring the Firewall` on page 9.

11. Configure your remote transfer server in the Faspex Web GUI.

Follow the instructions in `Aspera Faspex Admin Guide: Transfer Server` for configuring your remote transfer server in the Faspex Web GUI under `Server > File Storage`.

**Configuring for Shares**

The steps below show how to set up IBM Aspera Enterprise Server as a transfer server for IBM Aspera Shares. The procedure assumes you have already set up your Shares application. For general information on setting up a transfer server (using the Node API), see `Managing the Node API` on page 118.

1. Install Enterprise/Connect Server.

   Follow the instructions in `Standard Installation` on page 6 to install Enterprise Server either locally (on the same host as Shares) or remotely.

   The steps below must be performed as `root`.

2. Create a Node API username.

   Aspera's Web applications authenticate to the remote node service using a Node API username and password. The following command creates a Node API user/password and associates it with a file transfer user, `aspera_user_1`, which you will create in the next step. The Node API credentials can then be used to create nodes. Note that different nodes may use different Node API username/password pairs.

   ```
   # /opt/aspera/bin/asnodeadmin -a -u node_api_username -p node_api_passwd -x aspera_user_1
   ```

3. Create a file transfer user.

   The file transfer user authenticates the actual `ascp` transfer, and must be an operating system account on the node. To create a transfer user—for example, `aspera_user_1`—run the following command:

   ```
   # useradd aspera_user_1
   ```

   After you've created the operating system account, set up this user in Enterprise Server. For instructions on setting up a user, see `Setting Up Users`.

   **Note:** The file transfer user requires a `docroot`. After setting a user's docroot, be sure to perform a reload, as described in `asera.conf for Nodes`.

4. Copy the public key to the transfer user's SSH file.

   For example, if the file transfer user is `aspera_user_1`, the standard location for the public key is in the user's home directory, as follows: C:\Users\aspera_user_1\.ssh\authorized_keys
The Aspera-provided key file is located in:

```
/home/aspera_user_1/.ssh/authorized_keys
```

On the command line, run the following to create the user's public key folder (if it does not already exist):

```
# mkdir /home/aspera_user_1/.ssh
```

Run the following commands to create the keyfile `authorized_keys` (if it does not already exist), and append the key text to it. Update the directory permissions and ownership if necessary.

```
# cat /opt/aspera/var/aspera_id_dsa.pub >> /home/aspera_user_1/.ssh/authorized_keys
```

Run the following commands to change the key directory and keyfile's ownership to user `aspera_user_1`, to allow access by the `aspera_user_1` group, and to set permission bits:

```
# chown -R aspera_user_1:aspera_user_1 /home/aspera_user_1/.ssh/authorized_keys
# chmod 600 /home/aspera_user_1/.ssh/authorized_keys
# chmod 700 /home/aspera_user_1
# chmod 700 /home/aspera_user_1/.ssh
```

5. (Optional) Change HTTPS port and/or SSL certificate.

   The Aspera Node API provides an HTTPS interface for encrypted communication between node machines (on port 9092, by default). To modify the HTTPS port, see `aspera.conf for Nodes`. For information on maintaining and generating a new SSL certificate, see Setting up SSL for your Nodes on page 124.

6. Modify aspera.conf

   Make the following changes in the `aspera.conf` file, located in `/opt/aspera/etc`:

   - In the `<central_server>` section, look for `<persistent_store>` and be sure that it is set to `enable` (the default value). This setting allows the retention of historical transfer data used by the stats collector.
   - In the `<server>` section, look for the `<server_name>` tag, and replace `server_ip_or_name` with the name or IP address of your server. If the `<server>` section does not exist, create it.
   - Ensure there is an `<http_server>` section and that `<enable_http>` and `<enable_https>` are set to "1" (enabled).

   ```xml
   <central_server>
     <persistent_store>enable</persistent_store>
   </central_server>
   <server>
     <server_name>server_ip_or_name</server_name>
   </server>
   <http_server>
     <http_port>8080</http_port>
     <enable_http>1</enable_http>
     <https_port>8443</https_port>
     <enable_https>1</enable_https>
   </http_server>
   ```

   Whenever you change these settings, you must restart `asperacentral` and `asperanoded`.

   ```bash
   # /etc/init.d/asperanoded restart
   # /etc/init.d/asperacentral restart
   ```
7. In aspera.conf, enable token authorization for transfer users.  
   If you haven't done so already, set up the transfer user with an SSH public key as described in Setting Up Token Authorization on page 174. 
   In your aspera.conf file, add an authorization section for a transfer user as shown for the user aspera_user_1 in the example below. The authorization section should specify the following:
   • a <transfer> section specifying that both incoming and outgoing transfers (in and out) should use token encryption
   • a <token> section with an encryption key, which is a string of random characters (at least 20 characters recommended).

```xml
<user>
   <name>aspera_user_1</name>
   <authorization>
      <transfer>
         <in>
            <value>token</value>
         </in>
         <out>
            <value>token</value>
         </out>
      </transfer>
      <token>
         <encryption_key>gj5o930t78m34ejme9dx</encryption_key>
      </token>
   </authorization>
   <file_system>...
   </file_system>
</user>
```

Alternatively, you can configure token-authorization settings in a <group> section to be applied to all users in the group. Or, you can configure the settings in the <default> section to apply them globally for all users.  
For additional details on configuring token authorization, see Setting Up Token Authorization on page 174.

8. Ensure that the firewall is set up correctly on your transfer server  
For details, see Configuring the Firewall on page 9.

## Configuring for Aspera for SharePoint

This section describes how to set up IBM Aspera Enterprise Server as a transfer server for IBM Aspera for Microsoft SharePoint. It assumes that you have already set up your Microsoft SharePoint environment and configured (provisioned) it for SharePoint apps.

**Note:** In order to use IBM Aspera Enterprise Server as the transfer server for Aspera for SharePoint, you must run Enterprise Server on Windows 2012 or 2012 R2; or on Linux.

The basic steps are

1. Install the transfer server.  
2. Create a system user on the transfer server host.  
3. Create a directory to be the transfer user's docroot.  
4. Create the SSH key for the system user.  
5. Add the new system user as a transfer user to Enterprise/Connect Server.  
6. Specify a docroot for the new transfer user.  
7. Modify aspera.conf.
8. Ensure that the firewall is set up correctly on your transfer server host.
9. Verify your transfer server license.
10. Set up a node user.
11. Configure the Aspera for SharePoint application.

These steps are described in detail below.

1. **Install the transfer server.**

   If you haven't already, follow the steps in *Standard Installation* on page 6 to install Enterprise Server.

   The transfer server that you will use with Aspera for SharePoint must be installed on a host separate from your Microsoft SharePoint environment hosts.

   — *Note:* Aspera recommends you run Enterprise/Connect Server on a Linux host.

   All steps must be performed as root.

2. **Create a system user on the transfer server host.**

   The system user authenticates the actual `ascp` transfer and must be an operating system account.

   Run the following command to (1) create the system user `sharepoint`; (2) configure it to use the Aspera secure shell, `aspshell`; and (3) create the group `sharepoint` and assign the new system user to it.

   ```bash
   # /usr/sbin/useradd -r sharepoint -s /bin/aspshell-r -g sharepoint
   ```

3. **Create a directory to be the system user’s docroot.**

   This must be a location owned by the system user.

   You will use this location later in the configuration process, at Step 6 on page 114.

4. **Create the .ssh directory and public key file for the system user.**

   The standard location for the public key is in the user's home directory.

   `/home/sharepoint/.ssh/authorized_keys`

   The Aspera-provided key file is located in:

   `/opt/aspera/var/aspera_id_dsa.pub`

   1. On the command line, run the following to create the user's public key directory (if it does not already exist):

      ```bash
      # mkdir /home/aspera_user_1/.ssh
      ```

   2. Run the following commands to create the file `authorized_keys` (if it does not already exist), and append the key text to it.

      ```bash
      # cat /opt/aspera/var/aspera_id_dsa.pub >> /home/sharepoint/.ssh/authorized_keys
      ```

   3. Run the following commands to change the directory and file's ownership to system user `sharepoint`, to allow access by the `sharepoint` group, and to set permission bits:

      ```bash
      # chown -R sharepoint:sharepoint /home/sharepoint/.ssh/authorized_keys
      # chmod 600 /home/sharepoint/.ssh/authorized_keys
      # chmod 700 /home/sharepoint
      # chmod 700 /home/sharepoint/.ssh
      ```

5. **Add the `sharepoint` system user as a transfer user to Enterprise/Connect Server.**

   — *Note:* This step can also be done by modifying `aspera.conf`, instead of using the application GUI. For details, see *Setting Up Users* on page 55.

   1. Launch the Enterprise Server desktop application as administrator, and click **Configuration.**
2. In the **Server Configuration** dialog, select the **Users** tab. Then click the **+** button.

3. In the **Add User** dialog that appears, type **sharepoint** and click **OK**. The system user **sharepoint** is then added to the user list.

6. **Specify a docroot for the new transfer user sharepoint.**

   Still in the **Server Configuration** dialog, select the **Users** tab and do the following:
   
   1. Make sure **sharepoint** is selected in the user list.
   2. Open the **Docroot** tab in the right-hand panel.
   3. Set the following on the **Docroot** tab:

<table>
<thead>
<tr>
<th>Row</th>
<th>Override Setting</th>
<th>Effective Value Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Absolute Path</td>
<td>selected (checked)</td>
<td>/Users/sharepoint/ or whatever location you created in Step 3 on page 113</td>
</tr>
<tr>
<td>ad Allowed</td>
<td>selected (checked)</td>
<td>true</td>
</tr>
<tr>
<td>Write Allowed</td>
<td>selected (checked)</td>
<td>true</td>
</tr>
<tr>
<td>Browse Allowed</td>
<td>selected (checked)</td>
<td>true</td>
</tr>
</tbody>
</table>
Note: This step can also be done by modifying `aspera.conf`, instead of using the application GUI. For details, see Setting Up Users on page 55.

7. Modify `aspera.conf`.

The `aspera.conf` file is found in the following location:

```
/opt/aspera/etc/aspera.conf
```

Below is a typical `aspera.conf` file. Yours may differ, particularly if you have installed other Aspera products.

1. Modify the following settings, as necessary:

- `<persistent_store>`
  
  In the `<central_server>` section, find `<persistent_store>` and ensure that it is set to `enable` (the default value).
  
  This setting allows the retention of the historical transfer data that the stats collector uses.

- `<transfer>` and `<token>`
  
  To enable token authorization for the transfer user, add an authorization section that includes:
  
  - a `<transfer>` section specifying that both incoming and outgoing transfers (in and out) should use token encryption
  - a `<token>` section with an encryption key, which is a string of random characters (at least 20 characters recommended).
  
  See the example below.

  Note: Alternatively, you can configure token-authorization settings in a `<group>` section to be applied to all users in the group. Or, you can configure the settings in the `<default>` section to apply them globally for all users.

  For additional details on configuring token authorization, see Setting Up Token Authorization on page 174.

- `<dir_allowed>`
  
  In the `<file_system>` section, find `<dir_allowed>` and ensure that it is set to `true`.

- `<server_name>`
  
  In the `<server>` section, find `<server_name>` and ensure that `server_ip_or_name` is replaced with the name or IP address of your server.

```
<central_server>
    <persistent_store>enable</persistent_store>
</central_server>
...
<user>
    <name>sharepoint</name>
    <authorization>
```
2. After any change to aspera.conf, you must restart the asperacentral and asperanoded services.

   # /etc/init.d/asperanoded restart
   # /etc/init.d/asperacentral restart

8. Ensure that the firewall is set up correctly on your transfer server host.  
   For details, see Configuring the Firewall on page 9.

9. Verify your transfer server license.
   
   Verify that your transfer server license is Connect Server-enabled. (Aspera for SharePoint requires a Connect Server-enabled license.)
   
   To check this from the command line, run ascp -A and review the enabled settings list.
   
   For example:

   ```
   Enabled settings: connect, mobile, cargo, node, proxy,  
   http_fallback_server,  
   group_configuration, shared_endpoints, desktop_gui
   ```

   If the list includes connect, you have a Connect Server-enabled license.

   You can also check the license from the Enterprise Server desktop client GUI. The License dialog (Tools > License) includes the Connect Clients Enabled field. If it is set to Yes, you have a Connect Server-enabled license.

   Because this Faspex configuration uses Enterprise Server as a remote transfer service, it requires the Aspera Node API. For this reason, whenever you update your Enterprise Server license (see Updating the Product License on page 17), you must reload the asperanoded service afterwards. Reload the asperanoded service by running asnodeadmin, found in the following location:

   ```
   # /opt/aspera/bin/asnodeadmin --reload
   ```

10. Set up a node user.
A *node user* is the entity that Aspera's applications use for authentication between the Web application and the transfer server.

1. Create a node user, and then associate it with the system user that you created and configured in previous steps.

   Run the `asnodeadmin` command to
   - Create the `node_sharepoint` node user.
   - Assign a password of `s3cur3_p433` to the `node_sharepoint` node user.
   - Associate the `node_sharepoint` node user with the `sharepoint` transfer user.

   ```shell
   # /opt/aspera/bin/asnodeadmin -a -u node_sharepoint -p s3cur3_p433 -x sharepoint
   ```

2. Reload the `asperanoded` service.

   ```shell
   # /opt/aspera/bin/asnodeadmin --reload
   ```

3. Verify the node user.

   Run the `asnodeadmin` command to
   - Verify that the node user was created.
   - Verify the association between the node user and the transfer user.

   ```shell
   # /opt/aspera/bin/asnodeadmin -l
   ```

   The output for this command should resemble the following:

<table>
<thead>
<tr>
<th>List of node user(s):</th>
<th>system/transfer user</th>
<th>acls</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>system/transfer user</td>
<td>acls</td>
</tr>
<tr>
<td>sharepointnode</td>
<td>sharepoint</td>
<td>[]</td>
</tr>
<tr>
<td>spnode</td>
<td>aspera_user_1</td>
<td>[]</td>
</tr>
</tbody>
</table>

11. Configure the Aspera for SharePoint application.


   **Note:** The transfer server must be installed, configured, and running before you set up Aspera for SharePoint.
Managing the Node API

Overview: Aspera Node API

Capabilities of the Aspera Node API

The Aspera Node API is a feature of IBM Aspera Enterprise Server that provides a RESTful interface for full programmatic control of the Aspera transfer server environment. The Node API is a daemon that supports APIs for remote file operations, as well as initiating uploads and downloads.

The Node API includes the following features and functionality:

- An HTTPS (by default port 9092) and HTTP (by default port 9091) interface.
- An API in JSON data format.
- The API is authenticated and the node daemon uses its own application-level users (node users).
- A node admin utility called `asnodeadmin`, which can be utilized to add and manage node users.
- It logs to syslog, akin to asperacentral.

You can use the Node API to set up the following configurations:

- Set up a remote transfer server for Aspera Faspex. In this configuration, the Aspera Faspex Web UI is on Machine A, while the transfer server (an Enterprise Server node) is on Machine B. Machine A communicates with Machine B over HTTPS, by default.
- Set up nodes for Aspera Shares. In this configuration, the Aspera Shares Web UI is on Machine A, while content nodes (Enterprise Server nodes) are created on Machines B, C and D. Users can then be granted permission to access specific directories (shares) on nodes B, C and D.

Node API Setup

Setting up the Aspera Node API.

To set up the Aspera Node API, follow the instructions below. These instructions assumed that you have already installed Enterprise (or Connect) Server.

1. Create a Node API username.

   Aspera's Web applications authenticate to the remote node service using a Node API username and password. The following command creates a Node API user/password and associates it with a file transfer user, `aspera_user_1`, which you will create in the next step. The Node API credentials can then be used to create nodes. Note that different nodes may use different Node API username/password pairs.

   ```
   # /opt/aspera/bin/asnodeadmin -a -u node_api_username -p node_api_passwd -x aspera_user_1
   ```

2. Create a file transfer user.

   The file transfer user authenticates the actual `ascp` transfer, and must be an operating system account on the node. To create a transfer user—for example, `aspera_user_1`—run the following command:

   ```
   # useradd aspera_user_1
   ```

   After you've created the operating system account, set up this user in Enterprise Server. For instructions on setting up a user, see Setting Up Users.
**Note:** The file transfer user requires a *docroot*. After setting a user's docroot, be sure to perform a reload, as described in *aspera.conf for Nodes*.

3. **(Optional)** Change HTTPS port and/or SSL certificate.

The Aspera Node API provides an HTTPS interface for encrypted communication between node machines (on port 9092, by default). To modify the HTTPS port, see *aspera.conf for Nodes*. For information on maintaining and generating a new SSL certificate, see *Setting up SSL for your Nodes* on page 124.

### Setting up Node Users

Using `asnodeadmin` to set up node users

The `asnodeadmin` program can be used to manage (add, modify, delete, and list) node users. For each node user, you must indicate the following:

- Node username
- Node user's password
- Transfer/system username, which must be an operating system account on the node. This username is critical, since it's the user who authenticates the actual ascp transfer. If the transfer user is not mapped to the node user, then you will receive an error.

Recall in the topic *Node API Setup*, we created a node user and linked this user to file transfer user `aspera_user_1`. For `asnodeadmin` usage, please refer to the topic "Node Admin Tool."

**Important:** Note that adding, modifying or deleting a node-user triggers automatic reloading of the conf and license files, as well as the user database.

### Usage Examples

(All short options; use `asnodeadmin -h` to see the corresponding long options).

1. Add user “usr1” with password “pwd1” (will be prompted to enter if the `-p` option is not given) and associated transfer/system user “aspera”:

   ```
   # asnodeadmin -au usr1 -x aspera [-p pwd1]
   ```

2. Add user “usr2” with password “pwd2” and associated system/transfer user “root”:

   ```
   # asnodeadmin -au usr2 -p pwd2 -x root
   ```

3. Modify user “usr1” by assigning it a different password, “pwd1.1”:

   ```
   # asnodeadmin -mu usr1 -p pwd1.1
   ```

4. List users in the current user DB:

   ```
   # asnodeadmin -l
   ```

5. Delete user “usr1”:

   ```
   # asnodeadmin -du usr1
   ```

### Node Admin Tool

Usage Instructions for `asnodeadmin`

The help file below displays `asnodeadmin` options, which can be used to configure node users.
**Note:** Running `asnodeadmin` requires root privileges.

```
$ sudo /opt/aspera/bin/asnodeadmin -h
```

**Usage:** `asnodeadmin` [options]

Options:

- `-h,--help` Display usage.
- `-v,--version` Display version.
- `-f,--conf-file` Conf file pathname (default: `aspera.conf`).
- `--reload` Reload configuration settings, including the
  conf file and delete).
- `-a,--add` Add a user (also reloads configuration).
- `-d,--del[ete]` Delete an existing user (also reloads
  configuration).
- `-m,--mod[ify]` Modify an existing user (also reloads
  configuration).
- `--acl-add` Add new ACLs for a user. May be used with `-m`
  or `-a`.
- `--acl-set` Sets ACLs (clears old ACLs) for a user. May
  be used with `-m` or `-a`.
- `--acl-del` Deletes ACLs for a user. May be used with `-m`.
- `--acl-list` Lists all current ACLs for a user.
- `-l,--list` List users.
- `-u,--user=username` Specify username.
- `-p,--{pwd|password}=passwd` Specify password.
- `-x,--user=xfer_username` Specify system transfer user.
- `-b,--backup=filename` Back up user data to a file.
- `-r,--restore=filename` Restore user data from a file.
- `-P` Display hashed passwords as well when listing
  users.
- `-L local_log_dir` Local logging directory (default: no logging).
- `-D...` Debug level (default: no debug output).
- `--show-transfer-queue` Displays the current transfer queue
  log.
- `--access-key access_key` Specifies which `access_key`
  --transfer-log-del, --bearer-create, and --bearer-verify will use
- `--transfer-log-cleanup` Delete all transfers from the activity log
  older than `activity_retention_hrs`.
- `--show-transfer-log` Displays the output of data.
- `--bearer-create` Generate bearer token.
- `--key-file-path` Specifies the file path to read and write pem
  files for --bearer-create.
- `--token-key-length` Specifies the RSA key length for --bearer-
  create.
- `--user-id user_id` Specifies the user-id for --bearer-create.
- `--group-ids id1,id2,...` Specifies the group-id for --bearer-create.
- `--scope-role role` Specifies the scope role for --bearer-create.
- `--expires-at utc_date` Specifies the expiration date for --bearer-
  create.
- `--bearer-verify` Verify bearer token.
- `--db-shutdown` Shut down the database.
- `--db-update` Update KV store keys format to the latest
  version.
- `--db-update-local` Update KV store keys format to the latest
  version only if using the local redis.
aspera.conf for Nodes

Editing aspera.conf for your node configuration.

In your aspera.conf file, use the <server> section (shown below) to configure your node machines. The aspera.conf file is found in the following location:

```
/opt/aspera/etc/aspera.conf
```

**Note:** Each of the settings below requires certain services to be restarted in order for any changes to take effect. The services to restart are noted in the To Activate Changes column in the table below, and the commands to restart these services are given at the end of this topic.

```xml
<server>
  <server_name>your_hostname</server_name>
    <!-- hostname or IP address -->
  <http_port>9091</http_port>
    <!-- integer (1 - 65535) -->
  <https_port>9092</https_port>
    <!-- integer (1 - 65535) -->
  <enable_http>false</enable_http>
    <!-- true | false -->
  <enable_https>true</enable_https>
    <!-- true | false -->
  <cert_file>           <!-- full path; .chain file same /path/filename -->
    /opt/aspera/etc/aspera_server_cert.pem
  </cert_file>
  <max_response_entries>1000</max_response_entries>
    <!-- max entries to return in response -->
  <max_response_time_sec>10</max_response_time_sec>
    <!-- max seconds to wait for long operation -->
  <db_dir>/opt/aspera/var</db_dir>
    <!-- path to dir where DB file will be saved -->
  <db_port>31415</db_port>
    <!-- integer (1 - 65535) -->
  <proxy>
    ...
  </proxy>
  <enable_sslv2>true</enable_sslv2>
    <!-- boolean true or false -->
  <ssl_ciphers>TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA:...</ssl_ciphers>
    <!-- ssl_ciphers: see full default list in table below -->
  <ssl_protocol>sslv23</ssl_protocol>
    <!-- ss1v3, ss1v23, tls1v, tls1v1.1, or tls1v1.2 -->
</server>
```

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
<th>Default Value</th>
<th>To Activate Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;server_name&gt;</td>
<td>Hostname or IP address.</td>
<td>hostname</td>
<td>Restart node service</td>
</tr>
<tr>
<td>&lt;http_port&gt;</td>
<td>HTTP service port.</td>
<td>9091</td>
<td>Restart node service</td>
</tr>
<tr>
<td>Setting</td>
<td>Description</td>
<td>Default Value</td>
<td>To Activate Changes...</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>&lt;https_port&gt;</td>
<td>HTTPS service port.</td>
<td>9092</td>
<td>Restart node service</td>
</tr>
<tr>
<td>&lt;enable_http&gt;</td>
<td>Enable HTTP for the Node API services.</td>
<td>false</td>
<td>Restart node service</td>
</tr>
<tr>
<td>&lt;enable_https&gt;</td>
<td>Enable HTTPS for the Node API services.</td>
<td>true</td>
<td>Restart node service</td>
</tr>
<tr>
<td>&lt;cert_file&gt;</td>
<td>Full pathname of SSL certificate (.pem and existing support for .chain).</td>
<td>/opt/aspera/etc/aspera_server_cert.pem</td>
<td>Restart node service</td>
</tr>
<tr>
<td>&lt;max_response_entries&gt;</td>
<td>Maximum number of entries to return in a response.</td>
<td>1000</td>
<td>Reload node configuration.</td>
</tr>
<tr>
<td>&lt;max_response_time&gt;</td>
<td>Maximum amount of time to wait for a long-running operation.</td>
<td>10</td>
<td>Reload node configuration.</td>
</tr>
<tr>
<td>&lt;db_dir&gt;</td>
<td>Path to the directory where the database file is saved. Before changing this value, you should back up your database. See Redis DB Backup/Restore on page 124.</td>
<td>/opt/aspera/var</td>
<td>Restart the node and DB services.</td>
</tr>
<tr>
<td>&lt;db_port&gt;</td>
<td>Database service port. Before changing this value, you should back up your database. See Redis DB Backup/Restore on page 124.</td>
<td>31415</td>
<td>Restart the node and DB services.</td>
</tr>
<tr>
<td>&lt;ssl_ciphers&gt;</td>
<td>The SSL encryption ciphers that the server will allow, each separated by a colon (:). This option may also be set in the &lt;client&gt; section, in which case, when this machine functions as a client, the specified ciphers are requests to the server. If any of the ciphers in the server's allow list coincide with those in the client's request list, communication is allowed; otherwise it is denied. If you override this setting, the override is always used. However, if you do not override it, the default setting depends on the settings for &lt;ssl_protocol&gt;.</td>
<td>All of the following: <code>TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA</code>, <code>TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA</code>, <code>TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA</code>, <code>DHE-RSA-AES256-SHA</code>, <code>DHE-RSA-AES128-SHA</code>, <code>AES128-SHA256</code>, <code>AES256-SHA</code>, <code>AES128-SHA256</code>, <code>DHE-RSA-AES128-SHA</code>, <code>DHE-DSS-AES256-SHA</code>, <code>AES256-SHA</code>, <code>AES128-SHA256</code>, <code>DHE-RSA-AES128-SHA</code>, <code>DHE-DSS-AES128-SHA</code>, <code>RC2-CBC-MD5</code></td>
<td>Restart node service</td>
</tr>
</tbody>
</table>
### Setting | Description | Default Value | To Activate Changes...
--- | --- | --- | ---
| If `<ssl_protocol>` is set to `sslv23`, then a large, relatively weak selection of suites is allowed. If the protocol is anything else, then a smaller, stronger selection of suites is allowed. Many older web browsers cannot handle the stronger set of suites, in which case you may encounter compatibility issues. |  |  |  |
| `<ssl_protocol>` | The SSL protocol versions that the server will allow. This option may also be set in the `<client>` section, in which case, when this machine is a client, the specified protocols function as requests to the server. If any of the protocols in the server's allow list coincide with those in the client's request list, communication is allowed; otherwise it is denied. Supported values: `sslv3`, `tlsv1`, `tlsv1.1`, `tlsv1.2`, and `sslv23`. Despite its name, specifying `sslv23` (the default) allows all supported protocols, including all TLS versions. | `sslv23` | Restart node service. |
| `<enable_sslv2>` | Setting to true (default) enables SSLv2. If `<ssl_protocol>` is not set (or is explicitly set to its default `sslv23`), setting `<enable_sslv2>` to false allows only SSLv3 and TLSv1.x—that is, all protocols except SSLv2. If `<ssl_protocol>` is set to any value other than `sslv23`, settings for `<enable_sslv2>` settings have no effect. | true | Restart node service. |

**Note:** Running the commands below requires root privileges.

**Restarting the Node Service**

```sh
$ sudo /etc/init.d/asperanoded restart
```
### Managing the Node API

#### Reloading the Node Configuration

```bash
$ sudo /opt/aspera/bin/asnodeadmin --reload
```

#### Restarting the Node and DB Services

```bash
$ sudo /etc/init.d/asperanoded stop
$ sudo /opt/aspera/bin/asnodeadmin --db-shutdown
$ sudo /etc/init.d/asperanoded start
```

**Note:** The DB service is started automatically when you restart the node service.

---

### Redis DB Backup/Restore

Instructions for backing up and restoring the database.

To back up and restore the Redis database (and your user data up to the point-in-time of the backup operation), follow the instructions below. Note that the backup and restore operations should be used for the following scenarios:

- If you need to change the Redis database port number `<db_port/>` in `aspera.conf`, you should first back up the Redis database. Once you have changed the port number, you need to restore the database.
- Basic backup and restore (after a data-loss event).

1. **Back up the Redis database.**
   
   Use the following command to back up your Redis database *(before changing the port number):*
   ```bash
   $ sudo /opt/aspera/bin/asnodeadmin -b /your/backup/dir/database.backup
   ```

   **Important:** When backing up the Redis database, all user data up to that point-in-time will be saved to the backup file. *Restoring the database (see Step 2, below) does not delete users added after this snapshot was taken.* Thus, if you added any users after backing up the database, then they will still exist in the system and will not be affected by the restore operation.

2. **Restore the Redis database.**
   
   Use the following command to restore your Redis database:
   ```bash
   $ sudo /opt/aspera/bin/asnodeadmin -r /your/backup/dir/database.backup
   ```

   Recall the "Important Note" in Step 1, which stated that *restoring the database does not delete users added after the database snapshot was taken.* If you do not want to keep users that have been added since the last backup operation, you can delete them after performing the restore with the `asnodeadmin` command `-du username`.

3. **Restart the asperanoded service.**
   
   Use the following command(s) to restart the asperanoded service *(requires a restart rather than a reload):*
   ```bash
   $ sudo /etc/init.d/asperanoded restart
   ```

---

### Setting up SSL for your Nodes

#### Communicating with Aspera nodes over HTTPS

The Aspera Node API provides an HTTPS interface for encrypted communication between node machines (on Port 9092, by default). For example, if you are running the Faspex Web UI or the Shares Web UI on Machine A, you can encrypt the connection (using SSL) with your transfer server or file-storage node on Machine B. Enterprise Server
nodes are preconfigured to use Aspera's default, self-signed certificate (aspera_server_cert.pem), located in the following directory:

/opt/aspera/etc/

About PEM Files: The PEM certificate format is commonly issued by Certificate Authorities. PEM certificates have extensions that include .pem, .crt, .cer, and .key, and are Base-64 encoded ASCII files containing "-----BEGIN CERTIFICATE-----" and "-----END CERTIFICATE-----" statements. Server certificates, intermediate certificates, and private keys can all be put into the PEM format.

To generate a new certificate, follow the instructions below.

1. Enter the OpenSSL command to generate your Private Key and Certificate Signing Request (CSR).

   In this step, you will generate an RSA Private Key and CSR using OpenSSL. In a Terminal window, enter the following command (where my_key_name.key is the name of the unique key that you are creating and my_csr_name.csr is the name of your CSR):

   ```bash
   $ openssl req -new -nodes -keyout my_key_name.key -out my_csr_name.csr
   ```

2. Enter your X.509 certificate attributes.

   After entering the command in the previous step, you will be prompted to input several pieces of information, which are the certificate's X.509 attributes.

   🔄 Important: The Common Name field must be filled in with the fully qualified domain name of the server to be protected by SSL. If you are generating a certificate for an organization outside the U.S., go to https://www.iso.org/obp/ui/, select Country codes, and click to view a list of two-letter ISO country codes.

   Generating a 1024 bit RSA private key
   ........................+++.........
   ........................+++.........
   writing new private key to 'my_key_name.key'
   ------
   You are about to be asked to enter information that will be incorporated into your certificate request.
   What you are about to enter is what is called a Distinguished Name or a DN.
   There are quite a few fields but you can leave some blank.
   For some fields there will be a default value,
   if you enter '.', the field will be left blank.
   ------
   Country Name (2 letter code) [US]:Your_2_letter_ISO_country_code
   State or Province Name (full name) [Some-State]:Your_State_Province_or_County
   Locality Name (eg, city) [ ]:Your_City
   Organization Name (eg, company) [Internet Widgits Pty Ltd]:Your_Company
   Organizational Unit Name (eg, section) []:Your_Department
   Common Name (i.e., your server's hostname) []:secure.yourwebsite.com
   Email Address []:johndoe@yourwebsite.com

   You will also be prompted to input "extra" attributes, including an optional challenge password. Please note that manually entering a challenge password when starting the server can be problematic in some situations (for example, when starting the server from the system boot scripts). You can skip entering a challenge password by pressing Enter.

   ...
After finalizing the attributes, the private key and CSR are saved to your root directory.

**Important:** If you make a mistake when running the OpenSSL command, you may discard the generated files and run the command again. After successfully generating your key and CSR, be sure to guard your private key, as it cannot be re-generated.

3. Send the CSR to your signing authority.

You now need to send your unsigned CSR to a Certifying Authority (CA). Once completed, you will have a valid, signed certificate.

**Note:** Some certificate authorities provide a CSR generation tool on their website. For additional information, check with your CA.

4. Generate a self-signed certificate. (Optional)

At this point, you may need to generate a self-signed certificate for either of the following reasons:

- You don't plan on having your certificate signed by a CA.
- You plan to test your new SSL implementation while the CA is signing your certificate.

You can also generate a self-signed certificate through OpenSSL. To generate a temporary certificate (good for 365 days), run the following command:

```
openssl x509 -req -days 365 -in my_csr_name.csr -signkey my_key_name.key -out my_cert_name.crt
```

5. Create the .pem file.

After generating a new certificate, you must create a .pem file that contains both the private key and the certificate. To create the .pem file, copy and paste the entire body of the key and cert files into a single text file and save the file as aspera_server_cert.pem.

**Note:** Before overwriting the existing .pem file, be sure to back up this file as aspera_server_cert.old, in the following directory:

```
/opt/aspera/etc/
```

Case 1: If you have individual certificate files, the contents of the new .pem file should be added in the following sequence:

1. The private key.
2. The primary server's certificate.
3. The intermediate certificates, if any (if more than one, begin with the least authoritative and proceed in ascending order).
4. The root certificate.

Case 2: If you have a bundle of certificates, the contents of the new .pem file should be added in the following sequence:

1. The private key.
2. The primary server's certificate.
3. The entire bundle (as one file).

In this case (you have a certificate bundle), create a new file named aspera_server_cert.chain. This file must reside in the same installation directory as the .pem files. Place the root certificate in this file, followed by the bundle.

6. Enable SSL options in aspera.conf.

For information about enabling specific SSL protocols with `<ssl_protocol>` and enabling specific encryption ciphers with `<ssl_ciphers>`, see `aspera.conf for Nodes` on page 121.

7. Restart the node service.
You must restart (not reload) the Aspera node service after generating a new certificate. To do so, run the following command(s):

# /etc/init.d/asperanoded restart
Pre- and Post-Processing (Prepost)

Execute scripts before and after the FASP file transfers on your server.

Setting Up Pre/Post

Execute scripts before and after the FASP file transfers on your server.

Your Aspera server executes a shell script at a pre-defined location.

/opt/aspera/var/

This script is executed as a result of four (4) transfer events:

- Session start
- Session end
- Start of each individual file transfer in the session
- End of each individual file transfer in the session

The aspera-prepost script can also execute additional shell scripts, Perl scripts, native executables, and Java programs. Aspera also sets several environment variables for aspera-prepost that you can use in your own, custom scripts. These environment variables are described in detail in Pre/Post Variables on page 129. Depending on usage, pre- and post-processing may consume a large amount of system resources. Be sure to evaluate your own system performance and apply this feature appropriately.

Caution: When creating pre- and post-processing scripts, Be aware that an unsafe script can compromise a server. As with CGI scripts, you should take precautions in testing a pre/post script before placing it into use (such as taint checking, ensuring proper quotes, and so on). Also note that a pre/post script runs as the same user who authenticates the transfer. To prevent a pre/post script from performing an action with elevated or special user permissions, the script needs to check the $USER variable.

Follow the steps below to set up pre/post processing for your Aspera transfer product:

1. Set up the shell script file.

Locate the following file:

```
/opt/aspera/var/aspera-prepost.disable
```

This file runs the perl script aspera-notif.pl, which is an email notification script that sends emails (according to user-defined filters) to one or more recipients. Filters and lists are defined in the Aspera configuration file aspera.conf, which is located in /opt/aspera/etc.

Copy the contents of aspera-prepost.disable into a new file, and name it as follows:

```
/opt/aspera/var/aspera-prepost
```

Ensure that execute privileges are enabled (at least r-xr-xr-x).

2. Create your scripts.

The pre/post processing script, aspera-prepost, can contain the pre/post processing steps, as well as execute other programs (including other .bat scripts). Often, aspera-prepost checks for certain conditions (based on environment variables), and then calls a specific external executable based on those conditions. Recall that aspera-prepost is executed as a result of four (4) transfer events:

- session start
- session end
- start of each individual file transfer in the session
• end of each individual file transfer in the session

You can use the variables TYPE and STARTSTOP to specify a particular state. For the complete list of all variables, see Pre/Post Variables on page 129.

3. Include custom scripts in aspera-prepost.

Custom scripts can be written directly into the script file aspera-prepost. For example, to add the custom script "script1.pl" to your pre/post script, insert the following line (into aspera-prepost):

```perl
perl script1.pl
```

---

### Pre/Post Variables

The predefined variables for setting up the pre- and post-processing.

The following tables list all pre/post variables:

---

**Note:** Pre/post variables are case-sensitive.

#### For Type Session and Type File

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Values</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>COOKIE</td>
<td>The user-defined cookie string.</td>
<td>string</td>
<td>&quot;$COOKIE&quot; == cookie-string</td>
</tr>
<tr>
<td>DIRECTION</td>
<td>The transfer direction.</td>
<td>• send</td>
<td>&quot;$DIRECTION&quot; == send</td>
</tr>
<tr>
<td>ERRCODE</td>
<td>The error code.</td>
<td>string</td>
<td>&quot;$ERRCODE&quot; == 1</td>
</tr>
<tr>
<td>ERRSTR</td>
<td>The error string.</td>
<td>string</td>
<td>&quot;$ERRSTR&quot; == FASP error</td>
</tr>
<tr>
<td>MANIFESTFILE</td>
<td>The full path to the manifest file.</td>
<td>string or valid IPv4 address</td>
<td>&quot;$MANIFESTFILE&quot; == /log</td>
</tr>
<tr>
<td>PEER</td>
<td>The peer name or IP address.</td>
<td>string or valid IPv4 address</td>
<td>&quot;$PEER&quot; == 10.0.0.1</td>
</tr>
<tr>
<td>SECURE</td>
<td>Transfer encryption.</td>
<td>• yes</td>
<td>&quot;$SECURE&quot; == no</td>
</tr>
<tr>
<td>SESSIONID</td>
<td>The session id.</td>
<td>string</td>
<td>&quot;$SESSIONID&quot; == 1</td>
</tr>
<tr>
<td>STARTSTOP</td>
<td>The status start or stop.</td>
<td>• Start</td>
<td>&quot;$STARTSTOP&quot; == Start</td>
</tr>
<tr>
<td>STATE</td>
<td>The transfer state.</td>
<td>• started</td>
<td>&quot;$STATE&quot; == success</td>
</tr>
<tr>
<td>TOKEN</td>
<td>The user-defined security token.</td>
<td>string</td>
<td>&quot;$TOKEN&quot; == token-string</td>
</tr>
<tr>
<td>TYPE</td>
<td>The event type.</td>
<td>• Session</td>
<td>&quot;$TYPE&quot; == Session</td>
</tr>
<tr>
<td>USER</td>
<td>The user name</td>
<td>string</td>
<td>&quot;$USER&quot; == aspera_user_1</td>
</tr>
</tbody>
</table>

---
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Values</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>USERID</td>
<td>The user ID</td>
<td>string</td>
<td>&quot;$USERID&quot; == 501</td>
</tr>
<tr>
<td>USERSTR</td>
<td>The user string, such as additional variables.</td>
<td>string</td>
<td>&quot;$USERSTR&quot; == -q</td>
</tr>
</tbody>
</table>

**For Type Session**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Values</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILE_CSUM</td>
<td>Destination checksum of the most recently transferred file.</td>
<td>string</td>
<td>&quot;$FILE_CSUM&quot; == checksum</td>
</tr>
<tr>
<td>FILE1</td>
<td>The first file.</td>
<td>string</td>
<td>&quot;$FILE1&quot; == first-file</td>
</tr>
<tr>
<td>FILE2</td>
<td>The second file.</td>
<td>string</td>
<td>&quot;$FILE2&quot; == second-file</td>
</tr>
<tr>
<td>FILECOUNT</td>
<td>The number of files.</td>
<td>positive integer</td>
<td>&quot;$FILECOUNT&quot; &gt;= 5</td>
</tr>
<tr>
<td>FILELAST</td>
<td>The last file.</td>
<td>string</td>
<td>&quot;$FILELAST&quot; == last-file</td>
</tr>
<tr>
<td>LICENSE</td>
<td>The license account and serial number.</td>
<td>string</td>
<td>&quot;$LICENSE&quot; == license-string</td>
</tr>
<tr>
<td>MINRATE</td>
<td>The initial minimum rate, in Kbps.</td>
<td>positive integer</td>
<td>&quot;$MINRATE&quot; == 50</td>
</tr>
<tr>
<td>PEERLICENSE</td>
<td>The peer's license account and serial number.</td>
<td>string</td>
<td>&quot;$PEERLICENSE&quot; == license-string</td>
</tr>
<tr>
<td>RATEMODE</td>
<td>The transfer policy.</td>
<td>• adapt • fixed</td>
<td>&quot;$RATEMODE&quot; == adapt</td>
</tr>
<tr>
<td>SOURCE</td>
<td>The full path of the source file.</td>
<td>string</td>
<td>&quot;$SOURCE&quot; == /tmp</td>
</tr>
<tr>
<td>TARGET</td>
<td>The full path of the target directory.</td>
<td>string</td>
<td>&quot;$TARGET&quot; == .</td>
</tr>
<tr>
<td>TARGETRATE</td>
<td>The initial target rate, in Kbps.</td>
<td>positive integer</td>
<td>&quot;$TARGETRATE&quot; == 100</td>
</tr>
<tr>
<td>TOTALBYTES</td>
<td>The total bytes transferred.</td>
<td>positive integer</td>
<td>&quot;$TOTALBYTES&quot; &gt;= 100000000</td>
</tr>
<tr>
<td>TOTALSIZE</td>
<td>The total size of files being transferred in bytes.</td>
<td>positive integer</td>
<td>&quot;$TOTALSIZE&quot; &gt;= 500000000</td>
</tr>
</tbody>
</table>

**For Type File**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Values</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELAY</td>
<td>The measured network delay, in ms.</td>
<td>positive integer</td>
<td>&quot;$DELAY&quot; &lt;= 1</td>
</tr>
<tr>
<td>FILE</td>
<td>The file name.</td>
<td>string</td>
<td>&quot;$FILE&quot; == file-name</td>
</tr>
<tr>
<td>LOSS</td>
<td>The network loss in percentage.</td>
<td>double-digit fixed point value</td>
<td>&quot;$LOSS&quot; &gt;= 5.00</td>
</tr>
<tr>
<td>OVERHEAD</td>
<td>The total number of duplicate packets.</td>
<td>positive integer</td>
<td>&quot;$OVERHEAD&quot; &gt;= 1</td>
</tr>
<tr>
<td>RATE</td>
<td>The transfer rate in Kbps.</td>
<td>double-digit fixed point value</td>
<td>&quot;$RATE&quot; &gt;= 10.00</td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
<td>Values</td>
<td>Example</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
<td>--------</td>
<td>---------</td>
</tr>
<tr>
<td>REXREQS</td>
<td>The total number of retransmission requests.</td>
<td>positive integer</td>
<td>&quot;$REXREQS&quot; &gt;= 3</td>
</tr>
<tr>
<td>SIZE</td>
<td>The file size in bytes.</td>
<td>positive integer</td>
<td>&quot;$SIZE&quot; &gt;= 5000000</td>
</tr>
<tr>
<td>STARTBYTE</td>
<td>The start byte if resumed.</td>
<td>positive integer</td>
<td>&quot;$STARTBYTE&quot; &gt;= 100000</td>
</tr>
</tbody>
</table>

**Pre/Post Examples**

Pre- and post-processing script examples.

Pre- and post-processing script examples are shown below ("bash" syntax). To run these examples on your own system, do the following:

**Note:**
- Save the example to /opt/aspera/var/myscript.sh.
- Ensure that the script file is executable -- for example:
  ```bash
  chmod +x /opt/aspera/var/myscript.sh
  ```
- Add the line /opt/aspera/var/myscript.sh to /opt/aspera/var/aspera-prepost to call myscript.sh.
- Be sure there is no exit condition in aspera-prepost before you call your script.

1. **Shell** - Change file and directory permissions.

   In the shell script, change file and directory permissions after receiving, and log into the file /tmp/p.log:
   ```bash
   #!/bin/bash
   if [ $TYPE == File ]; then
     if [ $STARTSTOP == Stop ]; then
       echo "The file is: $FILE" >> /tmp/p.log
       chmod 777 $FILE
     fi
   fi
   fi
   ```

2. **Shell** - Forward files to another computer.

   In the shell script, transfer received files to a third computer 10.10.10.10, and remove the local copy.

   **Important:** For this example to work properly, the server's host key must be cached.
   ```bash
   #!/bin/bash
   TARGET=aspera@10.10.10.10:/tmp
   RATE=10m
   export ASPERA_SCP_PASS=aspera
   if [ $TYPE == File ]; then
   ```
if [ $STARTSTOP == Stop ]; then
  if [ $STATE == success ]; then
    if [ $DIRECTION == recv ]; then
      logger -p local2.info "Move file $FILE to $TARGET"
      ascp -T -o RemoveAfterTransfer=yes -l $RATE $FILE $TARGET
    fi
  fi
fi

3. Shell - Create a log of successfully transferred files.
In the shell script, store successfully transferred files as a list into the file /tmp/aspera.transfer.log:

#!/bin/bash
if [ $TYPE == File ]; then
  if [ $STARTSTOP == Stop ]; then
    if [ $SIZE -gt 0 ]; then
      if [ `expr $SIZE - $STARTBYTE` -gt 0 ]; then
        echo `date` >> /tmp/aspera.transfer.log
        echo "$STATE $FILE $SIZE bits transferred" >> /tmp/aspera.transfer.log
      fi
    fi
  fi
fi

Setting Up Email Notification

Configure the email notification, a prepost application.

The email notification feature is a built-in pre- and post-processing application that generates customized emails based on transfer events. Your server should have pre- and post-processing configured in order to run this application. For details, see Setting Up Pre/Post on page 128. Email notification requires an SMTP server that matches the following configurations:

- An open SMTP server you can reach on your network.
- The SMTP Server must not use any external authentication or SSL.

The following steps explain how to set up email notification:

1. Prepare the email notification configuration template.
   Open the aspera.conf file:

   /opt/aspera/etc/aspera.conf

   Locate or create the section <EMAILNOTIF>...</EMAILNOTIF>:

   <CONF version="2">
   ...
   <EMAILNOTIF>
   <MAILLISTS
   mylist = "asperausers@example.com, admin@example.com"
   myadminlist = "admin@example.com"
   />
   <FILTER
   MAILLISTS = "mylist"
   TARGETDIR = "/content/users"
   />
You can find the aspera.conf example in the following location:

/opt/aspera/etc/sample/aspera-sample.email.conf

2. Set up the basic Notification function in <MAILCONF/>

<MAILCONF/> defines the general email configuration, including the sender, the mail server, and the body text. In the SUBJECT and BODYTEXT options, the pre- and post-processing variables can be used with the format %{variable}, such as %{STATE} for the variable STATE. For the complete list of the variables, see Pre/Post Variables on page 129.

<table>
<thead>
<tr>
<th>MAILCONF Field</th>
<th>Description</th>
<th>Values</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>FROM</td>
<td>The email address to send notifications from. (Required)</td>
<td>a valid email address</td>
<td>FROM=&quot;<a href="mailto:admin@example.com">admin@example.com</a>&quot;</td>
</tr>
<tr>
<td>MAILSERVER</td>
<td>The outgoing mail server (SMTP). (Required)</td>
<td>A valid URL</td>
<td>MAILSERVER=&quot;mail.example.com&quot;</td>
</tr>
<tr>
<td>SUBJECT</td>
<td>General subject of the email.</td>
<td>text string</td>
<td>SUBJECT=&quot;Transfer:%{STATE}&quot;</td>
</tr>
<tr>
<td>BODYTEXT</td>
<td>General body of the email.</td>
<td>text string</td>
<td>BODYTEXT=&quot;Transfer has %{STATE}.&quot;</td>
</tr>
<tr>
<td>DEBUG</td>
<td>Print debugging info and write to the logs.</td>
<td>&quot;0&quot; = off, &quot;1&quot; = on</td>
<td>DEBUG=&quot;0&quot;</td>
</tr>
</tbody>
</table>

3. Create mailing lists in <MAILLISTS />.

<MAILLISTS/> defines sets of mailing lists. For example, to create the following mailing list:

<table>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mailing list name</td>
<td>list1</td>
</tr>
<tr>
<td>Emails to include</td>
<td><a href="mailto:janedoe@companymail.com">janedoe@companymail.com</a>, <a href="mailto:johndoe@companymail.com">johndoe@companymail.com</a></td>
</tr>
</tbody>
</table>

Specify the mailing list in the following form:

<MAILLISTS
    list1 = "janedoe@companymail.com, johndoe@companymail.com"
/>  

4. Set up mailing filters in <FILTER />.

<FILTER/> defines email notification conditional filters. When the conditions are met, a customized email is sent to the indicated mailing list. Multiple filters are allowed.

The values in the filter are matched as substrings, for example, USER = root means the value would match strings like root, treeroot, and root1. The pre- and post-processing variables can be used with the format
%{variable}, such as %{STATE} for the variable STATE. For the complete list of the variables, see Pre/Post Variables on page 129.

<table>
<thead>
<tr>
<th>FILTER Field</th>
<th>Description</th>
<th>Values</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAILLISTS</td>
<td>Required The email lists to send to. Separate lists with comma (,).</td>
<td>text string</td>
<td>MAILLISTS=&quot;mylist&quot;</td>
</tr>
<tr>
<td>USER</td>
<td>Login name of the user who transferred the files.</td>
<td>text string</td>
<td>USER=&quot;asera_user_1&quot;</td>
</tr>
<tr>
<td>SRCIP</td>
<td>Source IP of the files.</td>
<td>a valid IPv4 address</td>
<td>SRCIP=&quot;10.0.1.1&quot;</td>
</tr>
<tr>
<td>DESTIP</td>
<td>Destination IP of the files.</td>
<td>a valid IPv4 address</td>
<td>DESTIP=&quot;10.0.1.5&quot;</td>
</tr>
<tr>
<td>SOURCE</td>
<td>The top-level directories and files that were transferred.</td>
<td>text string</td>
<td>SOURCE=&quot;/folder1&quot;</td>
</tr>
<tr>
<td>TARGETDIR</td>
<td>The directory that the files were sent to.</td>
<td>text string</td>
<td>TARGETDIR=&quot;/folder2&quot;</td>
</tr>
<tr>
<td>SUBJECTPREFIX</td>
<td>The email subject, preceded by the SUBJECT in &lt;MAILCONF /&gt;.</td>
<td>text string</td>
<td>SUBJECTPREFIX=&quot;Sub&quot;</td>
</tr>
<tr>
<td>BODYPREFIX</td>
<td>The email body, preceded by the BODYTEXT in &lt;MAILCONF /&gt;.</td>
<td>text string</td>
<td>BODYPREFIX=&quot;Txt&quot;</td>
</tr>
<tr>
<td>TOTALBYTESOVER</td>
<td>Send email when total bytes transferred is over this number. This only applies to emails sent at the end of a transfer.</td>
<td>positive integer</td>
<td>TOTALBYTESOVER=&quot;9000&quot;</td>
</tr>
<tr>
<td>SENDONSESSION</td>
<td>Send email for the entire session.</td>
<td>yes / no</td>
<td>SENDONSESSION=&quot;yes&quot;</td>
</tr>
<tr>
<td>SENDONSTART</td>
<td>Send email when transfer is started. This setting is dependent on SENDONSESSION=&quot;yes&quot;.</td>
<td>yes / no</td>
<td>SENDONSTART=&quot;yes&quot;</td>
</tr>
<tr>
<td>SENDONSTOP</td>
<td>Send email when transfer is stopped. This setting is dependent on SENDONSESSION=&quot;yes&quot;.</td>
<td>yes / no</td>
<td>SENDONSTOP=&quot;yes&quot;</td>
</tr>
<tr>
<td>SENDONFILE</td>
<td>Send email for each file within a session.</td>
<td>yes / no</td>
<td>SENDONFILE=&quot;yes&quot;</td>
</tr>
</tbody>
</table>

**Email Notification Examples**

Email Notification configuration examples.

This topic demonstrates the email notification setup with the examples below.

1. Notify when a transfer session is completed.

   When a transfer session is finished, an email with brief session summary is sent to "list1".

   ```
   <EMAILNOTIF>
   <MAILLISTS
     list1 ="janedoe@companyemail.com, johndoe@companyemail.com"
   />
   <MAILCONF
   ```
2. Notify when a session is initiated and completed.

When a transfer is initiated, send a transfer notice email message. Send a summary email when finished.

```
<EMAILNOTIF>
<MAILLISTS>
  list1 = "johndoe@companyemail.com, janedoe@companyemail.com"
</MAILLISTS>
<MAILCONF>
  FROM="Aspera Notifier &lt;admin@companyemail.com&gt;"
  MAILSERVER="smtp.companyemail.com"
  SUBJECT=" by %{USER}"
  BODYTEXT="Powered by Aspera Inc."
</MAILCONF>
<FILTER>
  MAILLISTS="list1"
  SENDONSTART="yes"
  SENDONSTOP="no"
  SUBJECTPREFIX="Transfer Started"
  BODYPREFIX="Source: %{PEER}%{NEWLINE} Target: %{TARGET}"
</FILTER>
<FILTER>
  MAILLISTS="list1"
  SENDONSTART="no"
  SENDONSTOP="yes"
  SUBJECTPREFIX="Transfer Completed"
  BODYPREFIX="Status: %{STATE}%{NEWLINE} File Count: %{FILECOUNT}%{NEWLINE} Source: %{PEER}%{NEWLINE} Target: %{TARGET}%{NEWLINE} Bytes Transferred: %{TOTALBYTES} Bytes"
</FILTER>
</EMAILNOTIF>
```

3. Send different email text for regular transfers and for Aspera Sync transfers.

When Aspera Sync triggers a transfer (assuming only Aspera Sync uses the folder `/sync-folder`), an email message is sent to "mediaGroup". When a regular transfer occurs (files are sent to `/upload`), a different notification is sent to "mediaLead" and "adminGroup".

```
<EMAILNOTIF>
<MAILLISTS>
  mediaGroup = "johndoe@companyemail.com, janedoe@companyemail.com"
  mediaLead = "janedoe@companyemail.com"
  adminGroup = "admin@companyemail.com, root@companyemail.com"
</MAILLISTS>
```
<MAILCONF>
    FROM="Aspera Notifier &lt;admin@companyemail.com&gt;"
    MAILSERVER="smtp.companyemail.com"
    BODYTEXT="%{NEWLINE}Powered by Aspera Inc."
</MAILCONF>

<FILTER>
    MAILLISTS="mediaGroup"
    SENDONSESSION="yes"
    DESTIP="192.168.1.10"
    TARGETDIR="/sync-folder"
    SUBJECTPREFIX="Aspera Sync #1 - From %{PEER}" 
    BODYPREFIX="Status: %{STATE}%{NEWLINE} File Count: %{FILECOUNT}"
</FILTER>

<FILTER>
    MAILLISTS="mediaLead,adminGroup"
    SENDONSESSION="yes"
    TARGETDIR="/upload"
    SUBJECTPREFIX="Transfer - %{USER}"
    BODYPREFIX="Status: %{STATE}%{NEWLINE} Source: %{PEER}%{NEWLINE} File Count: %{FILECOUNT}%{NEWLINE} Bytes Transferred: %{TOTALBYTES} Bytes%{NEWLINE}"
</FILTER>
</EMAILNOTIF>
Ascp Command Reference

The executable ascp (Aspera secure copy) is a command-line FASP transfer program. This topic covers the complete command usage, including general syntax guidelines, supported environment variables, a synopsis, and command options.

General Syntax Guidelines

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>symbols used in the paths</td>
<td>Use single-quote (') and forward-slashes (/) on all platforms.</td>
</tr>
<tr>
<td>Characters to avoid</td>
<td>/ \ &quot; : ' ? &gt; &lt; &amp; *</td>
</tr>
</tbody>
</table>

Environment Variables

If needed, you can set the following environment variables for use with the ascp command:

<table>
<thead>
<tr>
<th>Item</th>
<th>Initiation Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password</td>
<td>ASPERA_SCP_PASS=password</td>
</tr>
<tr>
<td>Token</td>
<td>ASPERA_SCP_TOKEN=token</td>
</tr>
<tr>
<td>Cookie</td>
<td>ASPERA_SCP_COOKIE=cookie</td>
</tr>
<tr>
<td>Content Protection Password</td>
<td>ASPERA_SCP_FILEPASS=password</td>
</tr>
<tr>
<td>Proxy Server Password</td>
<td>ASPERA_PROXY_PASS=proxy_server_password</td>
</tr>
</tbody>
</table>

Ascp Usage

```
ascp options [[user@]srcHost:]source_file1[,source_file2,...] [[user@]destHost:]target_path
```

**Important:** If you do not specify a username for the transfer, the local username will be authenticated (by default). In the case of a Windows machine and a domain user, the transfer server will strip the domain from the username (for example, authenticating Administrator, rather than DOMAIN\Administrator). Thus, you will need to specify a domain explicitly, if applicable to the user.

Special Considerations for URI Paths

URIs are supported in paths, but only under the following restrictions:

- URIs can only be specified on the command line.
- If source paths are specified with a URI, all source paths specified on the command line must be from the same cloud storage account, and all must include URIs.
- If source paths are specified with a URI, no docroot (download), local docroot (upload), or source prefix can be specified.
- If a destination path is specified with a URI, no docroot (upload) or local docroot (download) can be specified.
• The special schemes stdio:// and stdio-tar:// are supported on the client only. Usage as a destination (upload) or source (download) is undefined.
• If required, URI passphrases can either be embedded in the URI or specified with the applicable environment variable ASPERA_SRC_PASS or ASPERA_DST_PASS.

Ascp Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Display usage.</td>
</tr>
<tr>
<td>-A, --version</td>
<td>Display version and license information; then exit.</td>
</tr>
<tr>
<td>-T</td>
<td>Disable encryption for maximum throughput.</td>
</tr>
<tr>
<td>-d</td>
<td>Create target directory if it doesn't already exist.</td>
</tr>
<tr>
<td>-q</td>
<td>Quiet mode (to disable progress display).</td>
</tr>
<tr>
<td>-v</td>
<td>Verbose mode (prints connection and authentication debug messages in the log file). For information on log files, see Log Files on page 179.</td>
</tr>
<tr>
<td>-6</td>
<td>Enable IPv6 address support. When using IPv6, the numeric host can be written inside brackets. For example, [2001:0:4137:9e50:201b:63d3:ba92:da] or [fe80::21b:21ff:felc:5072%eth1].</td>
</tr>
<tr>
<td>-D</td>
<td>-DD</td>
</tr>
<tr>
<td>-1 max_rate</td>
<td>Set the target transfer rate in Kbps (default: 10000 Kbps). If the ascp client does not specify a target rate, it will be acquired from aspera.conf (server-side, as the local aspera.conf target rate setting doesn't apply). If local or server aspera.conf rate caps are specified, the &quot;starting&quot; (default) rates will be not higher than the cap.</td>
</tr>
<tr>
<td>-m min_rate</td>
<td>Set the minimum transfer rate in Kbps (default: 0). If the ascp client does not specify a minimum rate, it will be acquired from aspera.conf (server-side, as the local aspera.conf minimum rate setting doesn't apply). If local or server aspera.conf rate caps are specified, the &quot;starting&quot; (default) rates will be not higher than the cap.</td>
</tr>
<tr>
<td>-u user_string</td>
<td>Apply a user string, such as variables for pre- and post-processing.</td>
</tr>
<tr>
<td>-i private_key_file</td>
<td>Use public key authentication and specify the private key file. Typically, the private key file is in the directory $HOME/.ssh/id_[algorithm].</td>
</tr>
<tr>
<td>-w{r</td>
<td>f}</td>
</tr>
<tr>
<td>-K probe_rate</td>
<td>Set probing rate (Kbps) when measuring bottleneck bandwidth.</td>
</tr>
<tr>
<td>-k{0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>• 0 – Always retransfer the entire file.</td>
</tr>
<tr>
<td></td>
<td>• 1 – Check file attributes and resume if the current and original attributes match.</td>
</tr>
<tr>
<td></td>
<td>• 2 – Check file attributes and do a sparse file checksum; resume if the current and original attributes/checksums match.</td>
</tr>
<tr>
<td></td>
<td>• 3 – Check file attributes and do a full file checksum; resume if the current and original attributes/checksums match.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>Note that when a complete file exists at the destination (no .aspx), the source file size is compared with the destination file size. When a partial file and a valid .aspx file exist at the destination, the source file size is compared with the file size recorded inside the .aspx file.</td>
<td></td>
</tr>
<tr>
<td>-Z dgram_size</td>
<td>Specify the datagram size (MTU) for FASP. By default, the detected path MTU is used. (Range: 296 - 10000 bytes)</td>
</tr>
<tr>
<td><strong>Note:</strong> As of version 3.3, datagram size can also be enforced by the server using <code>&lt;dgram_size&gt;</code> in <code>aspera.conf</code>. If size is set with both -Z (client side) and <code>&lt;dgram_size&gt;</code> (server side), the <code>&lt;dgram_size&gt;</code> setting is used. If the client-side is pre-3.3, datagram size is determined by the --dgram-size setting, regardless of the server-side setting for <code>&lt;dgram_size&gt;</code>. In this case, if there is no --dgram-size setting, datagram size is based on the discovered MTU and the server logs the message &quot;LOG Peer client doesn't support alternative datagram size&quot;.</td>
<td></td>
</tr>
<tr>
<td>-g read_size</td>
<td>Set the read-block size, in bytes. A read_size of 1M is 1 MB. The maximum block size is 500 MB. The default of 0 causes the Aspera sender to use its default internal buffer size.</td>
</tr>
<tr>
<td>This is a performance-tuning parameter for an Aspera sender, which takes effect only if the sender is a server. It specifies the maximum number of bytes that can be stored within a block as the block is transferred from the source disk to the receiver. This option overrides the client's configuration file setting for this feature if set. The server uses its configuration file setting for this feature if it's set, otherwise it uses read_size if set; however, it does not use settings in the client configuration file.</td>
<td></td>
</tr>
<tr>
<td>-G write_size</td>
<td>Set the write-block size, in bytes. A write_size of 1M is 1 MB. The maximum block size is 500 MB. The default of 0 causes the Aspera receiver to use its default internal buffer size.</td>
</tr>
<tr>
<td>This is a performance-tuning parameter for an Aspera receiver, which takes effect only if the receiver is a server. It specifies the maximum number of bytes within a block that an <code>ascp</code> receiver can write to disk. This option overrides the client's configuration file setting for this feature if set. The server uses its configuration file setting for this feature if it's set, otherwise it uses write_size if set; however, it does not use settings in the client configuration file.</td>
<td></td>
</tr>
<tr>
<td>-L local_log_dir</td>
<td>Specify a logging directory in the local host, instead of using the default directory.</td>
</tr>
<tr>
<td>-R remote_log_dir</td>
<td>Specify a logging directory in the remote host, instead of using the default directory.</td>
</tr>
<tr>
<td>-S remote_ascp</td>
<td>Specify the name of the remote <code>ascp</code> binary (if different).</td>
</tr>
<tr>
<td>-e prepost</td>
<td>Specify an alternate pre/post command. Be sure to use the complete path and file name.</td>
</tr>
<tr>
<td>-O fasp_port</td>
<td>Set the UDP port to be used by FASP for data transfer. (Default: 33001)</td>
</tr>
<tr>
<td>-P ssh-port</td>
<td>Set the TCP port to be used for FASP session initiation. (Default: 33001)</td>
</tr>
<tr>
<td>-C nid:ncount</td>
<td>Enable multi-session transfers (also known as parallel transfers) on a multi-node/multi-core system. Specify the node ID (nid) and count (ncount) in the format 1:2, 2:2. Assign each participant to an independent UDP port. Large files can also be split using --multi-session-threshold option. (See below.)</td>
</tr>
<tr>
<td>-E pattern</td>
<td>Exclude files or directories with the specified pattern from the transfer. This option can be used multiple times to exclude many patterns. Up to 16 patterns can be used by using -E. Two symbols can be used in the pattern, as shown below.</td>
</tr>
<tr>
<td><strong>•</strong> (asterisk) represents zero or more characters in a string, for example * .tmp matches .tmp and abcde .tmp.</td>
<td></td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>-?</td>
<td>(question mark) represents a single character, for example t?p matches tmp but not temp.</td>
</tr>
<tr>
<td>-f config_file</td>
<td>Specify an alternate Aspera configuration file (default is aspera.conf).</td>
</tr>
<tr>
<td>-W token_string</td>
<td>Specify the token string for the transfer.</td>
</tr>
<tr>
<td>-@ [range_low:range_high]</td>
<td>Transfer only part of a file. This option only works when downloading a single file and does not support resuming. The argument to &quot;-@&quot; may omit either or both numbers, and the &quot;:&quot; delimiter. For example, -@3000:6000 transfers bytes between positions 3000 to 6000; -@1000: transfers from 1000 to the end of the file; and -@:1000 transfers from beginning to 1000.</td>
</tr>
<tr>
<td>-X rexmsg_size</td>
<td>Adjust the maximum size in bytes of a retransmission request. (Max: 1440).</td>
</tr>
<tr>
<td>--mode=mode</td>
<td>Specify the transfer direction, where mode is either send or recv.</td>
</tr>
<tr>
<td>--user=username</td>
<td>The user name to be authenticated by the transfer server.</td>
</tr>
<tr>
<td><strong>Important</strong>: If you do not specify a user name for the transfer, the local username will be authenticated (by default). In the case of a Windows machine and a domain user, the transfer server will strip the domain from the username (e.g. authenticating &quot;Administrator,&quot; rather than &quot;DOMAIN\Administrator&quot;). Thus, you will need to explicitly specify a domain, if applicable to the user.</td>
<td></td>
</tr>
<tr>
<td>--host=hostname</td>
<td>The server's address.</td>
</tr>
<tr>
<td>--policy=fixed</td>
<td>Set the FASP transfer policy.</td>
</tr>
<tr>
<td>fixed</td>
<td>Attempts to transfer at the specified target rate, regardless of the actual network capacity. This policy transfers at a constant rate and finishes in a guaranteed time. This policy typically occupies most of the network's bandwidth, and is not recommended in most file transfer scenarios. In fixed mode, a maximum (target) rate value is required.</td>
</tr>
<tr>
<td>high</td>
<td>Monitors the network and adjusts the transfer rate to fully utilize the available bandwidth up to the maximum rate. When congestion occurs, it transfers at a rate twice of a session with fair policy. In this mode, both the maximum (target) and the minimum transfer rates are required.</td>
</tr>
<tr>
<td>fair</td>
<td>Monitors the network and adjusts the transfer rate to fully utilize the available bandwidth up to the maximum rate. When other types of traffic build up and congestion occurs, it shares bandwidth fairly by transferring at an even rate. In this mode, both the maximum (target) and the minimum transfer rates are required.</td>
</tr>
<tr>
<td>low</td>
<td>Similar to fair mode, the low policy uses the available bandwidth up to the maximum rate, but is much less aggressive when sharing bandwidth with other network traffic. When congestion builds up, the transfer rate is reduced to the minimum rate until other traffic retreats.</td>
</tr>
<tr>
<td><strong>Important</strong>: If --policy is not set, ascp uses the server-side policy setting (fair by default).</td>
<td></td>
</tr>
<tr>
<td>--file-list=filename</td>
<td>Take the list of sources to transfer from filename. The file list supports UTF-8 files and input from standard input through &quot;-&quot;. The sources can exist on either the local host or the remote host (in terms of download), but not on both. Each source must be specified on a separate line: src src2 ... srcN</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>Use with URIs:</strong> The sources listed in the file list cannot be URIs. To use <code>--file-list</code> with cloud storage, you must specify the cloud storage with a URI in either a root, source prefix, or command-line destination parameter, subject to the limitations described above. <strong>Important:</strong> Multiple <code>--file-list</code> and <code>--file-pair-list</code> options are not supported in a single <code>ascp</code> command. If multiple file lists are specified, all but the last will be ignored. In addition, you cannot also include file names on the command line when you use <code>--file-list</code>. Only files from the file list will be transferred.</td>
<td></td>
</tr>
<tr>
<td><code>--file-pair-list=filename</code></td>
<td>Take the list of sources and corresponding destinations from <code>filename</code>. Note that there is no command-line equivalent. Source and destination arguments in the file list cannot be URIs. Each source and each destination must be specified on a separate line:</td>
</tr>
<tr>
<td>src1</td>
<td>dst1</td>
</tr>
<tr>
<td>src2</td>
<td>dst2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>srcN</td>
<td>dstN</td>
</tr>
<tr>
<td><strong>Use with URIs:</strong> The sources and destinations listed in the file list cannot be URIs. To use <code>--file-pair-list</code> with cloud storage, you must specify the cloud storage with a URI in either a root, source prefix, or command-line destination parameter, subject to the limitations described above. <strong>Important:</strong> Multiple <code>--file-list</code> and <code>--file-pair-list</code> options are not supported in a single <code>ascp</code> command. If multiple file lists are specified, all but the last will be ignored. In addition, you cannot also include file names on the command line when you use <code>--file-pair-list</code>. Only files from the file-pair list will be transferred.</td>
<td></td>
</tr>
<tr>
<td><code>--symbolic-links=method</code></td>
<td>Specify rule to handle symbolic links. This option takes following values: (Default: follow)</td>
</tr>
<tr>
<td>• follow</td>
<td>Follow symbolic links and transfer the linked files.</td>
</tr>
<tr>
<td>• copy</td>
<td>Copy only the alias file. If a file with the same name exists on the destination, the symbolic link will not be copied.</td>
</tr>
<tr>
<td>• copy+force</td>
<td>Copy only the alias file. If a file with the same name exists on the destination, the symbolic link will replace the file. If the file of the same name on the destination is a symbolic link to a directory, it will not be replaced.</td>
</tr>
<tr>
<td>• skip</td>
<td>Skip the symbolic links.</td>
</tr>
<tr>
<td><code>--remove-after-transfer</code></td>
<td>Add this option to remove all source files (excluding the source directory) once the transfer has completed.</td>
</tr>
<tr>
<td><code>--move-after-transfer=archivedir</code></td>
<td>Move source files and copy source directories to <code>archivedir</code> after they are successfully transferred. Because directories are copied, the original source tree remains in place. The <code>archivedir</code> is created if it does not already exist. If the archive directory cannot be created, the transfer proceeds and the source files remain in their original location. <strong>Example upload:</strong></td>
</tr>
<tr>
<td><code>ascp --move-after-transfer=C:\Users\Bob\Archive C:\Users\Bob \srcdir\file0012 bob@10.0.0.1:</code></td>
<td>Result:</td>
</tr>
<tr>
<td>• file0012 is transferred to bob's docroot on 10.0.0.1</td>
<td></td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>• file0012 is moved (not copied) from its original location to C:\Users\Bob\Archive</td>
<td>Example download: \n  ascp --move-after-transfer=Archive bob@10.0.0.1:/srcdir C:\Users\Bob\Archive \n  \n  Result: \n  • srcdir is downloaded to C:\Users\Bob on the current machine. \n  • srcdir is moved (not copied) from its original location to the archive directory bob@10.0.0.1:/Archive on the server. \n  As with transfers, by default, no portion of the path above the transferred file or directory is included when the file or directory is moved to the archive (unless --src-base is specified). \n  The --src-base=prefix option preserves paths in the archive directory the same way as it preserves them with transfers. That is, when --src-base=prefix is specified, files are moved to the archivedir including the portion of the path that remains when prefix is removed. \n  Example: \n  ascp --src-base=C:\Users\Bob --move-after-transfer=C:\Users\Bob\Archive C:\Users\Bob\srcdir\file0012 bob@10.0.0.1:/ \n  Result: \n  • file0012 is transferred to bob's docroot on 10.0.0.1. The file is transferred and includes the path minus the prefix — that is, to srcdir/file0012. \n  • file0012 is moved to C:\Users\Bob\Archive. The file is moved and includes the path minus the prefix — that is, to C:\Users\Bob\Archive\srcdir\file0012. \n  Once files have been moved to the archive, the original source directory tree remains intact. To remove empty source directories that remain after files have been moved, include the flag --remove-empty-directories on the command line. This removes empty source directories except for those that are specified as the source to transfer. \n  Restrictions: \n  • archivedir must be on the same file system as the source. If the specified archive is on a separate file system, it will be created (if it does not exist), but an error will be generated and files will not be moved to it. For cloud storage, archivedir must be in the same cloud storage account. \n  • archivedir is subject to the same docroot restrictions as the source. \n  • --remove-after-transfer and --move-after-transfer are mutually exclusive; including both in the same command generates an error. \n  • Empty directories are not saved to archivedir. \n  --remove-empty-directories</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>--skip-special-files</code></td>
<td>Skip special files (for example, devices and pipes).</td>
</tr>
<tr>
<td><code>--file-manifest=output</code></td>
<td>Generate a list of all transferred files, where <code>output</code> is <code>none</code> or <code>text</code> (Default: <code>none</code>).</td>
</tr>
<tr>
<td><code>--file-manifest-path=directory</code></td>
<td>Specify the path to the file manifest. <strong>Important:</strong> File manifests can only be stored locally. Thus, if you are using S3, or other non-local storage, you must specify a <code>local</code> manifest path.</td>
</tr>
<tr>
<td><code>--file-manifest-inprogress-suffix=suffix</code></td>
<td>Specify the suffix of the file manifest's temporary file.</td>
</tr>
<tr>
<td><code>--precalculate-job-size</code></td>
<td>Add this option to calculate total size before transfer. Note that the server side <code>aspera.conf</code> setting overrides the <code>ascp</code> command-line option.</td>
</tr>
</tbody>
</table>
| `--overwrite=method` | Overwrite files with the same name. This option takes following values (Default: `diff`):  
  - `always` – Always overwrite the file.  
  - `never` – Never overwrite the file. However, note that if the parent folder is not empty, its access, modify, and change times may still be updated.  
  - `diff` – Overwrite if file is different from the source (i.e., if a complete file exists at the destination (no .aspx file) and is the same as the source file, then leave it unmodified (no change on timestamp/attributes either); otherwise re-transfer the whole source file). Note this policy interacts with the `resume` policy.  
  - `older` – Overwrite if file is older than the source. **Important:** When `--overwrite=diff`, you must also consider the `resume policy` (`-k{0|1|2|3}`). If `-k0` (or no `-k` specified), the source and destination files are always deemed to be different, thereby implying always overwrite. If `-k1`, the source and destination files are compared based on file attributes (currently, just file size). If `-k2`, the source and destination files are compared based on `sparse` checksum. If `-k3`, the source and destination files are compared based on `full` checksum. |
<p>| <code>--save-before-overwrite</code> | If a transfer will overwrite an existing file, move the existing file to <code>file.yyyy.mm.dd.hh.mm.ss.index.ext</code>, where <code>index</code> is set to 1 at the beginning of each new second and incremented for each file saved in this manner during the same second. File attributes are maintained in the renamed file. |
| <code>--file-crypt=crypt</code> | Encrypt or decrypt files for client-side encryption-at-rest (EAR). Valid values for <code>crypt</code> are <code>encrypt</code> and <code>decrypt</code>. Set the passphrase (required) with the environment variable <code>ASPERA_SCP_FILEPASS</code>. Encrypted files have the file extension <code>.aspera-env</code>. Note that if a client-side encrypted file is downloaded with an incorrect password, the download is successful but the file is still encrypted and still has the file extension <code>.aspera-env</code>. |
| <code>--file-checksum=hash</code> | Report checksums for transferred files, where <code>hash</code> is <code>sha1</code>, <code>md5</code>, or <code>none</code>. |
| <code>--partial-file-suffix=suffix</code> | Filename extension on the destination computer while the file is being transferred. Once the file has been completely transferred, this filename extension will be removed. (Default: blank) <strong>Note:</strong> This option only takes effect when it is set on the receiver side. |</p>
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--source-prefix=pref</code></td>
<td>Add <code>prefix</code> to the beginning of each source path. This can be either a conventional path or a URI; however, it can only be a URI if there is no root defined.</td>
</tr>
</tbody>
</table>
| `--src-base=pref` | Specify the prefix to be stripped off from each source object. The remaining portion of the source path is kept intact at the destination. Special care must be taken when using this option with cloud storage.  

Example: The "clips" directory on the remote computer contains the following folders and files:  

```
/clips/outgoing/file1  
/clips/outgoing/folderA/file2  
/clips/outgoing/folderB/file3  
```

In this case, to transfer all folders and files within the "outgoing" folder (but not the "outgoing" folder, itself), run the following command:

```
$ ascp -d --src-base=/clips/outgoing/ root@10.0.0.1:/clips/outgoing/ /incoming  
```

Result: The following folders and files appear in the "incoming" directory at the destination:

```
(docroot)/incoming/file1  
(docroot)/incoming/folderA/file2  
(docroot)/incoming/folderB/file3  
```

Files outside of the source base (for example, `/temp/file4`) are not transferred, and warnings are generated.

**Without `--src-base`**

If `--src-base` is not used, and the source item is a folder, the contents of the folder are transferred, along with the folder itself. For example:

```
$ ascp -d root@10.0.0.1:/clips/outgoing/ /incoming  
```

Result:

```
(docroot)/incoming/outgoing/file1  
(docroot)/incoming/outgoing/folderA/file2  
(docroot)/incoming/outgoing/folderB/file3  
```

If `--src-base` is not used, and the source item is a file, only the file is transferred, not the folders in the file's path. For example:

```
$ ascp -d root@10.0.0.1:/clips/outgoing/file1 root@10.0.0.1:/clips/outgoing/ folderA/file2 /incoming  
```

Result:

```
(docroot)/incoming/file1  
(docroot)/incoming/file2  
```

For further examples, with and without `--src-base`, see *Ascp File Manipulation Examples* on page 148.

**Use with URIs**

The `--src-base` option performs a character-to-character match with the source path specifying a file or directory. Hence for cloud storage, it is necessary that `--src-base` specify the URI.
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--proxy=proxy_url</td>
<td>Specify the address of the Aspera proxy server. <code>proxy_url</code> takes the form of: <code>dnat[s]://[username]@server:port</code></td>
</tr>
<tr>
<td></td>
<td>The default ports for DNAT and DNATS protocols are 9091 and 9092.</td>
</tr>
</tbody>
</table>
| --preserve-file-owner-uid | (OS X and Linux/UNIX systems only.) Preserve transferred files' owner information (uid).  
|                         | **Note:** This option requires the transfer user be authenticated as a superuser.                                                        |
| --preserve-file-owner-gid | (OS X and Linux/UNIX systems only.) Preserve transferred files' group information (gid).  
|                         | **Note:** This option requires the transfer user be authenticated as a superuser.                                                        |
| --preserve-creation-time | Preserve creation time: Set the file/directory creation time at the destination to that of the source.                                   |
| --preserve-modification-time | Available on Windows clients only. If the destination is a non-Windows host, this option is ignored.                                    |
|                         | (Note: Do not confuse this with UNIX ctime, which represents "change time", indicating the time when metadata was last updated.)       |
| --preserve-access-time  | Preserve modification time: Set the file/directory modification time at the destination to that of the source.                        |
|                         | Preserve access time: Set the file/directory access time (the last time the file was read or written) at the destination to that of the source. This results in the destination file having the access time that the source file had prior to the copy operation. The act of copying the source file to the destination results in an update to the source file's access time. |
| --preserve-source-access-time | Preserve source access time: Restore the access time of the file at the source once the copy operation is complete (because the file system at the source regards the transfer operation as an access). |
| -p                      | --p is equivalent to setting both `--preserve-modification-time` and `--preserve-access-time`.                                           |
|                         | On Windows, modification time may be affected when the system automatically adjusts for Daylight Savings Time (DST). For details, see the Microsoft KB article, `http://support.microsoft.com/kb/129574`. |
|                         | Cloud storage support for timestamp settings depends on the cloud storage implementation. See the documentation for your cloud storage option to determine which of these settings are supported. For Limelight, only the preservation of modification time (`mtime`) is supported. |
| --ignore-host-key       | If you are prompted to accept a host key when connecting to a remote host, `ascp` ignores the request.                                      |
| --check-sshfp=fingerprint | Check against the server SSH host key fingerprint (for example, `5de9c0d62eafa0616ed1e851133c42a0082`).                                        |
| --apply-local-docroot   | Apply the local docroot. This option is equivalent to setting the environment variable `ASPERA_SCP_DOCROOT`.                          |
This option augments the existing `-C` option, which enables multi-session transfers (also known as parallel transfers). With the threshold option, if the size of the files to be transferred is greater than or equal to `threshold`, files will be split. Files with sizes less than the threshold will not be split for the multi-session transfer. If not set, the default is that no files are split.

The client node API can also specify the multi-session-threshold, and this will be passed to the ascp command line. A default value for the threshold can be specified in `aspera.conf` by setting `multi_session_threshold_default`. By default, the threshold default is "unspecified" (NULL). Setting it to 0 (zero) means "do not split". The command-line setting overrides the `aspera.conf` setting. If the client doesn't specify a value, then the `multi_session_threshold_default` is taken from the server.

Note: For cloud transfers, file-splitting is currently (3.6.0) supported for S3 only.

For cloud transfers, the process of splitting files in a multi-session transfer differs from a regular (non-cloud) multi-session transfers. For cloud transfers, files are sent in chunks, and the chunk size is specified by `<chunk_size>` in `aspera.conf`:

```xml
<central_server>
  ...
  <transfer>
    <protocol_options>
      <transfer>
        <chunk_size>0</chunk_size>
      </transfer>
    </protocol_options>
  </transfer>
</central_server>
```

Multi-session uploads to cloud storage with file splitting must have the `ascp` chunk size and the cloud-storage part size set to equal values. File-splitting needs to respect a minimum split size, which for cloud storage is a `part`. That is, each `ascp` call needs to deliver full parts. However, note that if the file size is greater than the multi-session threshold and smaller than the chunk size, then there is no split. Set chunk size and part size as follows:

1. In `aspera.conf` set the chunk size to some value greater than 5 MB; for example:
   ```xml
   <chunk_size>67108864</chunk_size> <!-- 64 MB -->
   ```

2. In `/opt/aspera/etc/trapd/s3.properties`:
   - Set the upload part size (default 64 MB) to the same value as the chunk size.
   - Use a `ONE_TO_ONE` gathering policy:
     ```
     aspera.transfer.upload.part-size=64MB
     aspera.transfer.gathering-policy=ONE_TO_ONE
     ```

---

Delete files that exist at the destination but not at the source, before any files are transferred. Do not use with multiple sources, keepalive, or HTTP fallback. The utility `asdelete` provides the same capability.

---

Preserve extended attributes (xattrs) and/or access control lists (ACLs) when transferring files between different types of file systems. `mode` can be one of `native`, `metafile`, or `none` (default):

- **native**: xattrs and ACLs are preserved using native capabilities of the file system.
- **metafile**: however, this storage mode is not supported on all file systems.
### Ascp Options for HTTP Fallback

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-y {0</td>
<td>1}</td>
</tr>
<tr>
<td>-j {0</td>
<td>1}</td>
</tr>
<tr>
<td>-Y key_file</td>
<td>The HTTPS transfer's key file name.</td>
</tr>
<tr>
<td>-I cert_file</td>
<td>The HTTPS certificate's file name.</td>
</tr>
<tr>
<td>-t port</td>
<td>Specify the port for HTTP Fallback Server.</td>
</tr>
<tr>
<td>-x proxy_server</td>
<td>Specify the proxy server address used by HTTP Fallback.</td>
</tr>
</tbody>
</table>

### Ascp General Examples

Examples of initiating FASP file transfers using the `ascp` command.

This topic demonstrates the `ascp` command with the following examples:

- **Fair-policy transfer, without encryption**
  Transfer with fair rate policy, with maximum rate 100 Mbps and minimum at 1 Mbps:

```bash
$ ascp -T --policy=fair -l 100m -m 1m /local-dir/files root@10.0.0.2:/remote-dir
```

- **Fixed-policy transfer, without encryption**
  Transfer all files in `\local-dir\files` to `10.0.0.2` with target rate 100 Mbps and encryption OFF:

```bash
$ ascp -T -l 100m /local-dir/files root@10.0.0.2:/remote-dir
```
• Specify a UDP port
To perform a transfer with UDP port 42000:

$ ascp -l 100m -O 42000 /local-dir/files user@10.0.0.2:/remote-dir

• Authenticate with public key
To perform a transfer with public key authentication with key file <home dir>/ssh/aspera_user_1-key local-dir/files:

$ ascp -T -l 10m -i ~/.ssh/aspera_user_1-key local-dir/files root@10.0.0.2:/remote-dir

• Authenticate with a login that contains space
Enclose the target in double-quotes when spaces are present in the username and remote path:

$ ascp -l 100m local-dir/files "User Name@10.0.0.2:/remote directory"

• Transfer with a network shared location
Send files to a network shares location \\1.2.3.4\nw-share-dir, through the computer 10.0.0.2:

$ ascp local-dir/files root@10.0.0.2:"\\1.2.3.4\nw-share-dir/"

• Parallel transfer on a multi-core system
Use parallel transfer on a dual-core system, together transferring at the rate 200Mbps, using UDP ports 33001 and 33002. Two commands are executed in different Terminal windows:

$ ascp -C 1:2 -O 33001 -l 100m /file root@10.0.0.2:/remote-dir &
$ ascp -C 2:2 -O 33002 -l 100m /file root@10.0.0.2:/remote-dir

• Use content protection
Upload the file space\file to the server 10.0.0.2 with password protection (password: secRet):

$ set ASPERA_SCP_FILEPASS=secRet ascp -l 10m --file-crypt=encrypt local-dir/file root@10.0.0.2:/remote-dir/

Download from the server 10.0.0.2 and decrypt while transferring:

$ set ASPERA_SCP_FILEPASS=secRet ascp -l 10m --file-crypt=decrypt root@10.0.0.2:/remote-dir
/local-dir

If the password-protected file is downloaded without decrypting (file1.aspera-env, with aspera-env appended), on the local computer, decrypt the file as file1:

$ set ASPERA_SCP_FILEPASS=secRet asunprotect -o file1 file1.aspera-env

Ascp File Manipulation Examples

Examples of manipulating files using the ascp command.

This topic demonstrates file manipulation using the ascp command with the following examples:

1. Upload directory contents to remote computer
   Upload the "/content/" directory to the remote server.

   $ ascp /data/content/ root@10.0.0.1:/storage/

   Result => /storage/content/*
Upload the "/content/" directory to the remote server, but strip the srcbase path and preserve the rest of the file structure.

```
$ ascp --src-base=/data/content /data/content/ root@10.0.0.1:/storage
```

Result => /storage/*

2. Upload directory contents to remote computer and create the destination folder if it does not already exist

Upload the "/content/" directory to the remote server and create the "/storage2" folder since it does not exist.

```
$ ascp -d /data/content/ root@10.0.0.1:/storage2/
```

Result => /storage2/content/*

3. Download directory contents from remote computer

Download the "/content/" directory to the remote server, but strip the srcbase path and preserve the rest of the file structure.

```
$ ascp --src-base=/storage/content root@10.0.0.1:/storage/content/ /data
```

Result => /data/*

4. Upload selected files and directories to a remote computer and preserve directory structure

Upload the selected file and directory to the remote server, but strip the srcbase path and preserve the rest of the file structure.

```
$ ascp --src-base=/data/content /data/content/monday/file1 /data/content/ tuesday/ root@10.0.0.1:/storage
```

Results => /storage/monday/file1 AND /storage/tuesday/*

5. Download selected files and directories from a remote computer and preserve directory structure

Download the selected file and directory from the remote server, but strip the srcbase path and preserve the rest of the file structure.

```
$ ascp --src-base=/storage/content root@10.0.0.1:/storage/content/monday/ file1 root@10.0.0.1:/storage/content/tuesday/ /data
```

Results => /data/monday/file1 AND /data/tuesday/*

6. Remove source files from the local computer after transferring them to the remote computer

Remove the "/content/" directory of the local computer after the contents (excluding partial files) have been transferred to the remote computer.

```
$ ascp -k2 -E ".*\.partial" --remove-after-transfer --remove-empty-directories /data/content root@10.0.0.1:/storage
```

Result => /storage/content/*

Remove the "/content/" directory of the local computer after the contents (excluding partial files) have been transferred to the remote computer. Strip the srcbase path and preserve the rest of the file structure

```
$ ascp -k2 -E ".*\.partial" --src-base=/data/content --remove-after-transfer --remove-empty-directories /data/content root@10.0.0.1:/storage
```

Result => /storage/*
**Important:** For version 2.7.1, the "-d" option is required when specifying the "--src-base" option if the target directory does not exist. As of version 2.7.3+, this constraint has been removed.

## Ascp Transfers to Cloud Storage

Examples of using the ascp command to initiate FASP transfers to cloud storage.

If you have access to cloud storage that is hosted by Aspera On Demand, you can use `ascp` to transfer to it.

### With Docroot Already Configured

If your transfer server account already has a docroot set up, `ascp` transfers to S3 storage, Google storage, Akamai, Softlayer, and Azure are the same as regular `ascp` transfers:

```
ascp options myfile username@server:/targetpath
```

For examples, see *Ascp General Examples* on page 147.

In some cases, `ascp` transfers to cloud storage can be made without a preconfigured docroot. See the examples below.

### With No Docroot Configured: S3

If the transfer server account does not have a docroot, you can still transfer to S3 as long as you know your S3 Access ID and Secret Key and you have an S3 bucket. The syntax is:

```
ascp options --mode=send --user=username --host=s3_server_addr files
s3://access_id:secret_key@s3.amazonaws.com/s3_bucket
```

For example:

```
ascp --mode=send --user=bob --host=s3.asperasoft.com myfiles
s3://1K3C18FBWF9902:GEyU...AqXuxtTVHWtc@s3.amazonaws.com/demos2014
```

### With No Docroot Configured: Softlayer

If the transfer server account does not have a docroot, you can still transfer with the following syntax:

```
ascp options --mode=send --user=root --
host=ip_addr files swift://softlayer user:
api_key@location.objectstorage.service.networklayer.com/container
```

Example Upload:

```
ascp --mode=send --user=root --host=192.155.218.130 bigfile.txt
```
Example Download:

```
ascp --mode=recv --user=root --host=192.155.218.130 swift://XYZO...46-2:bob:437e...bc16@sjc01.objectstorage.service.networklayer.com/test/bigfile.txt /tmp/
```

With No Docroot Configured: Azure

If the transfer server account does not have a docroot, you can still transfer. First set an Aspera environment variable with the password:

Windows Command Prompt: `set ASPERA_SCP_PASS = password`

Linux: `export ASPERA_SCP_PASS=password`

Then run `ascp` with the following syntax:

```
ascp options --mode=send --user=uname --host=server files azu://storage:key@blob.core.windows.net/abc
```

For example:

Windows Command Prompt: `set ASPERA_SCP_PASS = fslk47CLwlj`

Linux: `export ASPERA_SCP_PASS=fslk47CLwlj`

```
ascp --mode=send --user=AS037d8eda429737d6 --host=dev920350144d2.azure.asperaondemand.com bigfile.txt azu://astransfer:zNfMt...nBTkhB@blob.core.windows.net/abc
```

Configuring the Aspera Watch Service

The Aspera Watch Service (asperawatchd) is a file system change detection and snapshot service for speed, scale and distributed sources. It discovers changes (new files and directories, deleted items, renames, and so on) in source file systems immediately as they occur, eliminating the need to scan the file system. It can be used on any local or shares (CIFS, NFS) host.

When used in conjunction with `ascp` commands, the Aspera Watch Service allows for fast detection and send of new and deleted items. By comparing snapshots of the file directory it is watching, `asperawatchd` generates file lists for `ascp` commands for transfers and delete requests.

1. To configure `asperawatchd`, open the `aspera.conf` file with an editor located at `/opt/aspera/etc/aspera.conf` and add the `<watchd>` section.

   - Set `<enabled>` to true.
   - Set `<master>` to true.
   - Set a directory for `<log_level>`.
   - Set the data storage for `<db_spec>` (`.datastore:host:port`). By default, `asperawatchd` uses Redis with port 31415.
   - Set directories in `<watches>` for `asperawatchd` to monitor.

```
<server>
  <watchd>
    <enabled>true</enabled>
    <log_level>log</log_level>
    <log_directory>/tmp</log_directory>
  </watchd>
</server>
```
The `<watch>` section specifies directories for asperawatchd to monitor. When starting a session with Aspera Sync, Sync requests snapshots from asperawatchd, a process which takes longer depending on the number of files in the directory. The asperawatchd service periodically scans the directories specified in aspera.conf, reducing time spent waiting for a new completed scan.

**Tip:** You can configure asperawatchd to watch multiple directories, but Aspera recommends specifying a single `<watch>` at the top level directory. Though you can specify as many watches as needed, (for example, one for each directory), specifying a single watch is the most efficient use of memory. To monitor both the `/data` and `/data/D1` directories, specify `/data` as the path for the `<watch>`.

2. After properly configuring the asperawatchd service, you must manually start the service.

```
# service asperawatchd start
```

Check to make sure the asperawatchd service is running.

```
# ps aux | grep watchd
root 2306  0.1  0.5 732968 11220 ?      Sl    Nov13  1:17 /opt/aspera/sbin/asperawatchd
```

---

### Transferring and Deleting Files with the Aspera Watch Service

When used in conjunction with `ascp` commands, the Aspera Watch Service (`asperawatchd`) allows for fast detection and sending of new and deleted items. By comparing snapshots of the file directory it is watching, `asperawatchd` generates file lists for `ascp` commands for transfers requests. To generate snapshots and file lists, you must first configure and start `asperawatchd`. For more information, see *Configuring the Aspera Watch Service*.

**Process Overview**

1. Retrieve the `uuid` of the watcher for the directory configured in `aspera.conf`.
2. Take a snapshot of the current state of the directory.
3. Generate a snapshot differential between the previous state and the current state and save it in a file list.
4. Use the file list in an `ascp` command (to send created and modified files) or in an `asdelete` command (to delete files).

**Retrieving the Watcher uuid**

To generate a file list for all the changes in a directory, first retrieve the `uuid` of the watcher for the directory you configured in the `aspera.conf` file under the `<watches>` section.

```
# /opt/aspera/bin/aswatchadmin --db-spec=datastore:host:port --tool daemon
```

The `<max_directories>` tag allows for the maximum number of directories to be monitored, the `<max_queue>` tag allows for the maximum number of files to be monitored in a queue, the `<max_snapshots>` tag allows for the maximum number of snapshots to be generated, the `<master>` tag allows the directory to be monitored as master, the `<db_spec>` tag allows setting the database specification, the `<scan_period>` tag allows the scan period to be set, and the `<watches>` tag allows the watches to be specified.
For example, if the directory set in the `<watches>` section is `/data/D1`, run this command:

```
# /opt/aspera/bin/aswatchadmin --db-spec=redis:localhost:31415 --tool daemon 
--get-by-path=/data/D1

[aswatchadmin::daemon] Successfully retrieved watcher
[ct=[uid=e565a3e3-e25e-44c2-9d68-db1d59cef4c7,p=/data/D1,i_c=true,f=[]],
in=[t=CHANNEL,n='wt:e565a3e3-e25e-44c2-9d68-db1d59cef4c7:input'],
in_b=[t=CHANNEL,n='wt:e565a3e3-e25e-44c2-9d68-db1d59cef4c7:broadcast'],
t=[c=[ss.max=10000],p=[t=REDIS,db=wt:e565a3e3-e25e-44c2-9d68-db1d59cef4c7@localhost:31415],c=1]]
```

In the results, find the string for the `uuid` (for example, `uuid=e565a3e3-e25e-44c2-9d68-db1d59cef4c7`) and copy the `uuid`.

**Taking a Snapshot**

A snapshot represents the structure of the file system at a particular point of time; `asperawatchd` compares two snapshots to detect file system changes and generate a file list of those changes. To take a snapshot, run the following command with the watcher `uuid` associated with the file system.

```
# /opt/aspera/bin/aswatchadmin --db-spec=redis:localhost:31415 --tool=instance --uuid=watched_uuid --snapshot

For example, to take a snapshot with the watcher with the `uuid` we retrieved in the previous section:

```
# /opt/aspera/bin/aswatchadmin --db-spec=redis:localhost:31415 --tool=instance --uuid=e565a3e3-e25e-44c2-9d68-db1d59cef4c7 --snapshot

[aswatchadmin::instance] Successfully created snapshot 1.
```

**Sending New and Modified Files**

Take a snapshot of the directory. Generate and save a file list of all files that have been created or modified. Then, run an `ascp` command with that file list to send those files to another location.

```
# /opt/aspera/bin/aswatchadmin --db-spec datastore:host:port --tool tree --uuid=uuid_no --diff=snapshot_version:snapshot_version --format \${$PATH} --exclude-remove > /filelist

# ascp --file-list /filelist --source-prefix=/prefix --mode=send --user=username --host=host /target_directory\target_directory
```

For example:

```
# /opt/aspera/bin/aswatchadmin --db-spec=redis:localhost:31415 --tool tree --uuid=e565a3e3-e25e-44c2-9d68-db1d59cef4c7 --diff=0:1 --format \${$PATH} --exclude-remove > /filelist

# ascp --file-list ~/filelist --source-prefix=/data/D1 --mode=send --user=aspera --host=10.0.0.1 /data/R2

file1 100% 10MB 9.7Mb/s 00:07
file2 100% 10MB 9.7Mb/s 00:16
file3 100% 10MB 9.7Mb/s 00:26
file4 100% 10MB 9.7Mb/s 00:33
file5 100% 10MB 9.7Mb/s 00:42
file6 100% 10MB 9.7Mb/s 00:51

Completed: 61440K bytes transferred in 52 seconds (96454 bits/sec), in 6 files.
```
Tip: Aspera recommends using the source prefix option in conjunction with file lists to keep the file smaller. Alternatively, you can format the generated file list with a source prefix by specifying the prefix in the `--format` option. For example, specifying `--format /data/D1\$PATH` prefixes "/data/D1" to every entry in the file list.

Removed Files
The asdelete utility compares the source directory with the target directory and deletes extraneous files from the target directory. Use asdelete utility to remove files from a target directory.

```
# /opt/aspera/bin/asdelete --host host --auth-name username --auth-pass password /source_directory /target_directory
```

For example:

```
# /opt/aspera/bin/asdelete --host 10.0.0.1 --auth-name root --auth-pass !XF345lui@0 /data/D1 /data/R1
```

Check the target directory to confirm deletion of the correct files.

Token Generation
Usage and examples for astokengen

Overview
A token authorizes the download of one or more files, or an upload of one or more files into a directory (called destination root). It supports the traditional “cp” paradigm of ascp (copy file1, file2, file3 to directory) or source/destination pairs (ascp `--file-pair-list`).

Functionality
- Authorizes uploads of one or more files to a destination
- Authorizes downloads of one or more files or directories
- Authorizes uploads of one or more files as source/destination pairs
- Authorizes downloads of one or more files as source/destination

Usage
1. astokengen --mode=send [options] -u user --dest=path [-v token]
2. astokengen --mode=send [options] -u user --file-pair-list=filename --dest=destination [-v token]
3. astokengen --mode=recv [options] -u user -p path [-p path ...] [-v token]
4. astokengen --mode=recv [options] -u user --file-list=filename [-v token]
5. astokengen --mode=recv [options] -u user --file-pair-list=filename [-v token]
6. astokengen -t token [options]

<table>
<thead>
<tr>
<th>Option (short form)</th>
<th>Option (long form)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-A</td>
<td>--version</td>
<td>Print version information.</td>
</tr>
<tr>
<td></td>
<td>--mode=mode</td>
<td>Direction of the transfer mode (send</td>
</tr>
<tr>
<td>-p</td>
<td>--path=path</td>
<td>Source path</td>
</tr>
<tr>
<td></td>
<td>--dest=destination</td>
<td>Destination path</td>
</tr>
<tr>
<td>Option (short form)</td>
<td>Option (long form)</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>-u</td>
<td>--user=user</td>
<td>Generate the token for this user name. This name is embedded in the token and also used to retrieve further information from \texttt{aspera.conf}(user_value and \texttt{token_life_seconds}).</td>
</tr>
<tr>
<td></td>
<td>--file-list=filename</td>
<td>Specifies a file name that contains a list of sources for a download token. Each line of the file contains a single source and blank lines are ignored.</td>
</tr>
<tr>
<td></td>
<td>--file-pair-list=filename</td>
<td>Specifies a file name that contains a multiplexed list of source and destination pairs for an upload or download token. Each pair of lines encodes one source and one destination and blank lines are ignored.</td>
</tr>
<tr>
<td>-v token</td>
<td></td>
<td>Verify token against user and path parameters.</td>
</tr>
<tr>
<td>-t token</td>
<td></td>
<td>Display the contents of the token.</td>
</tr>
<tr>
<td>-k passphrase</td>
<td></td>
<td>Passphrase to decrypt token. For use with \texttt{-t}.</td>
</tr>
<tr>
<td>-b</td>
<td></td>
<td>Assume user name and paths are encoded in base64.</td>
</tr>
</tbody>
</table>

**Examples**

<table>
<thead>
<tr>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
</table>
| Example file list                  | /monday/first\_thing\.txt  
/monday/next\_thing\.txt  
/monday/last\_thing\.txt          |
| Example file-pair list             | /monday/first\_thing\.txt  
/archive/monday/texts/first\_thing  
/monday/next\_thing\.txt  
/archive/monday/texts/next\_thing  
/monday/last\_thing\.txt  
/archive/monday/texts/last\_thing|
| Common upload                      | In a common upload, only the destination is encoded into the token.  
\texttt{astokengen --user=user --dest=path --mode=send}  
The destination is encoded into the token. Source paths are not allowed and will cause \texttt{astokengen} to fail. \texttt{--path} and \texttt{--file-list} are illegal in this case. |
| Paired upload                      | The destination is prepended to each of the destinations in the paired list file and they are all encoded into the token. The destinations are in each odd numbered line of the file (1, 3, 5, 7, and so on).  
\texttt{astokengen --user=user --dest=path --file-pair-list=filename --mode=send}  
In this case, \texttt{--path} and \texttt{--file-list} are illegal. |
| Common download                    | The specified paths are encoded into the token.  
\texttt{astokengen --user=user --path=file1 --path=file2 --mode=recv} |
Creating SSH Keys (Command Line)

Create a key pair for your computer.

If you are using this machine as a client to connect to other Aspera servers with public key authentication, you can also create key-pairs in command line. Follow these instructions:

**Note:** You can also use the application GUI to create SSH keys or import existing keys for use with a selected user account. For instructions, see Creating SSH Keys on page 26.

1. Create a `.ssh` in your home directory if it does not already exist:

   ```shell
   $ mkdir /home/username/.ssh
   
   Go to the `.ssh` folder and continue:
   2. Run `ssh-keygen` to generate an SSH key-pair.

   Run the following command in the `.ssh` folder. The program prompts you for the key-pair's filename. Press ENTER to use the default name `id_rsa`. For a passphrase, you can either enter a password, or press return twice to leave it blank:

   ```shell
   $ ssh-keygen -t rsa
   
   3. Retrieve the public key file.

   When created, the key-pair can be found in your home directory's `.ssh` folder (assuming you generated the key with the default name `id_rsa`):

   ```shell
   /home/username/.ssh/id_rsa.pub
   
   Provide the public key file (for example, `id_rsa.pub`) to your server administrator, so that it can be set up for your server connection. The instructions for installing the public key on the server can be found in the Setting Up a User's Public Key on page 63; however, the server may be installed on an operating system that is different from the one where your client has been installed.

   4. Start a transfer using public key authentication with the `ascp` command.

   To transfer files using public key authentication on the command line, use the option `-i private_key_file`. For example:

   ```shell
   $ ascp -T -l 10M -m 1M -i ~/.ssh/id_rsa myfile.txt jane@10.0.0.2:/space
   
   In this example, you are connecting to the server (10.0.0.2, directory /space) with the user account jane and the private key ~/.ssh/id_rsa.
Ascp FAQs

This topic lists frequently asked questions regarding `ascp` command:

1. How do I control the transfer speed?

You can specify a transfer policy that determines how a FASP transfer utilizes the network resource, and you can specify target and minimum transfer rates where applicable. With the `ascp` command, use the following flags to specify transfer policies that are fixed, fair, high, and low:

<table>
<thead>
<tr>
<th>Policy</th>
<th>Command template</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed</td>
<td><code>--policy=fixed -l target_rate</code></td>
</tr>
<tr>
<td>Fair</td>
<td><code>--policy=fair -l target_rate -m min_rate</code></td>
</tr>
<tr>
<td>High</td>
<td><code>--policy=high -l target_rate -m min_rate</code></td>
</tr>
<tr>
<td>Low</td>
<td><code>--policy=low -l target_rate -m min_rate</code></td>
</tr>
</tbody>
</table>

2. What should I expect in terms of transfer speed? How do I know if something is "wrong" with the speed?

Aspera's FASP transport has no theoretical throughput limit. Other than the network capacity, the transfer speed may be limited by rate settings and resources of the computers. To verify that your system's FASP transfer can fulfill the maximum bandwidth capacity, prepare a client machine to connect to this computer, and test the maximum bandwidth.

Note: This test typically occupies most of a network's bandwidth. Aspera recommends this test be performed on a dedicated file transfer line or during a time of low network activity.

On the client machine, start a transfer with fixed bandwidth policy. Start with a lower transfer rate and increase gradually toward the network bandwidth (for example, 1m, 5m, 10m, and so on). Monitor the transfer rate and make sure it fulfills your bandwidth:

```bash
$ ascp -l 1m source-file destination
```

To improve the transfer speed, also consider upgrading the following hardware components:

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hard disk</td>
<td>The I/O throughput, the disk bus architecture (such as RAID, IDE, SCSI, ATA, and Fiber Channel).</td>
</tr>
<tr>
<td>Network I/O</td>
<td>The interface card, the internal bus of the computer.</td>
</tr>
<tr>
<td>CPU</td>
<td>Overall CPU performance affects the transfer, especially when encryption is enabled.</td>
</tr>
</tbody>
</table>

3. How do I ensure that if the transfer is interrupted or fails to finish, it will resume without retransferring the files?

Use the `--k` flag to enable resume, and specify a resume rule:

- `-k 0` Always retransfer the entire file.
- `-k 1` Check file attributes and resume if they match.
- `-k 2` Check file attributes and do a sparse file checksum; resume if they match.
- `-k 3` Check file attributes and do a full file checksum; resume if they match.

4. How does Aspera handle symbolic links?
The `ascp` command follows symbolic links by default. There is a `-o SymbolicLink` flag that offers handling options:

- `--symbolic-links=follow`: Follow symbolic links and transfer the linked files.
- `--symbolic-links=copy`: Copy only the alias file.
- `--symbolic-links=skip`: Skip the symbolic links.

5. What are my choices regarding file overwrites on the destination computer?

In `ascp`, you can specify the overwriting rule with the following flags:

- `--overwrite=always`: Always overwrite the file.
- `--overwrite=never`: Never overwrite the file.
- `--overwrite=diff`: Overwrite if file is different from the source.
- `--overwrite=older`: Overwrite if file is older than the source.

**Note:** For `--overwrite=diff`, if a complete file (that is, no `.aspx` file) exists on the destination computer and is the same as the source file, then the destination file will remain unmodified (no change on timestamp/attributes either). Otherwise the entire source file will be retransferred. Note this policy interacts with the resume policy.
Configuring for the Cloud

Cloud Configuration for Enterprise Server Nodes

Configuring aspera.conf for S3

The following example explains how to modify aspera.conf for AWS S3 transfers. You must meet the following prerequisites before modifying aspera.conf:

• You have permissions to access the S3 bucket.
• You know your username's S3 Access ID and Secret Key.
• The Aspera server is located in the same region as your S3 bucket.

Note: For Aspera on Demand, you can also enter these settings from Console.

```xml
<?xml version='1.0' encoding='UTF-8'?>
<CONF version="2">
<server>
  <server_name>aspera.example.com</server_name>
</server>
<aaa>
  <realms><realm><users>
    <user>
      <name>UserName</name>
      <authorization>
        <transfer>
          <in>
            <value>token</value>
          </in>
          <out>
            <value>token</value>
          </out>
        </transfer>
        <token>
          <encryption_key>YourSuperSecretKey</encryption_key>
        </token>
      </authorization>
      <file_system>
        <access>
          <paths>
            <path>
              <absolute></absolute>
              <read_allowed>true</read_allowed> <!-- Read Allowed: boolean true or false -->
              <write_allowed>true</write_allowed> <!-- Write Allowed: boolean true or false -->
              <dir_allowed>true</dir_allowed> <!-- Browse Allowed: boolean true or false -->
              <restrictions>
                <!-- File access restrictions. Multiple entries are allowed. -->
                <restriction>s3://*/</restriction>
                <restriction>!azu://*/</restriction>
              </restrictions>
            </path>
          </paths>
        </access>
      </file_system>
    </user>
  </users>
</realm>
</realms>
</aaa>
</CONF>
```
Docroot Restrictions for URI Paths

A configuration with both a docroot absolute path (docrooted user) and a restriction is not supported.

The primary purpose of restrictions is to allow access to certain storage (for example, Amazon S3) for clients that have their own storage credentials. In this case, instead of using docroots in aspera.conf, use a docroot restriction.

Configuration:

```
<paths>
  <path>
    <restrictions>
      <restriction>s3://*</restriction>
    </restrictions>
  </path>
</paths>
```

You can also configure restrictions once for all users by setting `<restriction>` in the default section.

Functionality:

A docroot restriction limits the files a client is allowed to access for browsing and transfers. Files are rejected unless they match any restrictions that are present. Restrictions work for URI paths (for example, s3://*) and are processed in the following order:

1. If a restriction starts with "!", any files that match are rejected.
2. If a restriction does not start with a "!", any files that match are kept.
3. If any restrictions other than "!" exist, and the file does not match any of them, the file is rejected.
4. Files that fail restrictions during directory iteration are ignored as if they do not exist.
Authentication and Authorization

Access Key Authentication

Access key authentication provides an alternative to the security credentials of a node user or system user. Because an access key is restricted to its own storage (local or cloud), it allows access control and usage reporting to be segregated by storage. This offers significant benefits to multi-tenant service providers and enterprise installations with multiple departments. Access key authentication supports Aspera client products, such as Desktop Client, Point-to-Point Client, Enterprise Server, Connect, and Drive. It also supports Faspex, Shares, and Aspera Files. For details about using access key authentication with these products, see the documentation for these products.

Node Access through SSH and HTTPS

A node (a transfer server) is accessed over SSH or HTTPS:

- SSH services are best suited for access when all computers are part of the same administrative domain (in the same organization and with the same administrators).
- HTTPS is best suited for services offered to arbitrary clients or the Internet at large.

Access through SSH and HTTPS use various types of authentication:

- SSH services (ascp and async)
  - SSH user and password or user and key
  - Token - SSH authentication bypass using the Aspera web private key protected by requiring an authorization token (ATM).
  - Access key - SSH authentication bypass using the Aspera web private key protected by requiring an access key and secret.
- HTTPS (node API)
  - Basic authentication using a node user and password.
  - Basic authentication using an access key and secret.

Creating and Testing Access Keys

Set up a node user and associate it with a system user by running the `asnodeadmin` command, as in the following example, where `asp1` is the node user, `aspera` is the node user's password, and `xfer` is the system user. Then run `asnodeadmin` again to reload `asperanoded`. Running `asnodeadmin` requires root or administrator permissions.

```
# /opt/aspera/bin/asnodeadmin -a -u asp1 -p aspera -x xfer
# /opt/aspera/bin/asnodeadmin --reload
```

Run `curl` commands as in the following examples to create access keys. To create an access key with local storage, run the following:

```
$ curl -d @access_key-make-local.json -ki -u 'asp1:aspera' https://localhost:9092/access_keys
```

where:

- `@access_key-make-local.json` Indicates the next argument is the data to send. The "@" identifies `access_key-make-local.json` as a file containing the data, in this case, a JSON payload file.
-i
Includes the HTTP header in the output.

-k
Allows curl to perform "insecure" SSL connections and transfers.

-u 'asp1:aspera'
Specifies the user name asp1 and password aspera to use for server authentication.

https://localhost:9092/access_keys
Indicates where to store the access keys.

Create an access key with cloud storage for Swift:

```bash
$ curl -d @access_key-make-swift.json -ki -u 'asp1:aspera' https://localhost:9092/access_keys
```

Check the access keys:

```bash
$ curl -ki -u asp1:aspera https://localhost:9092/access_keys
HTTP/1.1 200 OK
Cache: no-cache
Connection: close
Content-Type: application/json; charset=utf-8
[
    {
        "id": "diDeuFLcpG9IYdsvxj0SCq4mOohNJTKvp5Q2nRWjDgIA",
        "uri": "file:///home/asp1/data",
        "file_id": "1",
        "token_verification_key": null,
        "license": null,
        "storage": {
            "type": "local",
            "path": "/home/asp1/data"
        }
    },
    {
        "id": "Yc6Q4VuvaYA9mMRf55NyNsiVGC-HHSBh0FTuqMH8aHsA",
        "uri": "swift://sjc01.objectstorage.softlayer.net/wallball",
        "file_id": "1",
        "token_verification_key": null,
        "license": null,
        "storage": {
            "type": "softlayer_swift",
            "path": "/",
            "container": "wallball",
            "credentials": {
                "authentication_endpoint": "https://sjc01.objectstorage.softlayer.net/auth/v1.0",
                "username": "IBMOS303446-2%3AIBM303446",
                "api key": "e0a8987b571cca4e475c8dd816c2d2db71b6d6e060f2a75ce23b1832c12d6706"
            }
        }
    }
]
```

Test whether you can browse the storage for each key. To specify the user (-u) enter the access key ID.

Testing the local storage:

```bash
$ curl -ki -u 'diDeuFLcpG9IYdsvxj0SCq4mOohNJTKvp5Q2nRWjDgIA:aspera' https://localhost:9092/files/1/files
```
Testing the Swift storage:

```bash
$ curl -ki -u 'Yc6Q4VuvaYA9mMRf55NyNsiVGC-HHSBh0FTuqMH8aHsA:aspera' https://localhost:9092/files/1/files
```

Examples of JSON payload files for various storage types:

**access_key-make-local.json**

```json
{
  "id": "diDeuFLcpG9IYdsvxj0SCq4mOohNJTKvp5Q2nRWjDgIA",
  "secret": "aspera",
  "storage": {
    "type": "local",
    "path": "/home/asp1/data"
  }
}
```

**access_key-make-aws.json**

```json
{
  "id": "AWSQ4VuvaYA9mMRf55NyNsiVGC-HHSBh0FTuqMH8aHsA",
  "secret": "aspera",
  "storage": {
    "type": "aws_s3",
    "path": "/",
    "endpoint": "s3.amazonaws.com",
    "bucket": "aspera-demo",
    "storage_class": "STANDARD",
    "server_side_encryption": null,
    "credentials": {
      "access_key_id": "AKI...............KHQ",
      "secret_access_key": "KScx...............................PHcm1"
    }
  }
}
```

**access_key-make-azure-sas.json**

```json
{
  "secret": "aspera",
  "storage": {
    "type": "azure_sas",
    "path": "/",
    "credentials": {
      "shared_access_signature": "https://asperadev.blob.core.windows.net/temp?sv=2014-02-14&sr=c&sig=yfew...79uXE%3D&st=2015-07-29T07%3A00%3A00Z&se=2018-08-06T07%3A00%3A00Z"
    }
  }
}
```

**access_key-make-azure.json**

```json
{
  "secret": "aspers",
  "storage": {
    "type": "azure",
    "container": "temp",
  }
}
```
Client-Server Authentication Using Basic Auth with Access Key

Basic authentication is used by Aspera Faspex and Aspera Shares.

Server setup:

1. Create a node user by running `asnodeadmin` as described above in Creating and Testing Access Keys.
2. Identify the storage for this access key, either local or cloud, as described above in Creating and Testing Access Keys.
3. Create a system user for the SSH bypass (asp1).

   ```
   $ cat /opt/aspera/var/aspera_id_dsa.pub > /home/asp1/.ssh/authorized_keys
   chown asp1 /home/asp1/.ssh/*
   chmod 755 /home/asp1/authorized_keys
   ```

Server configuration:

When using access key authentication, which includes storage, consider the following:

- Access keys cannot be used with docroots. If a docroot is configured (in `aspera.conf`), access key creation and use will fail.
- "Restriction" is required. No restriction results in failure.
- Although access keys can be created with no storage, using that access key with a transfer will result in failure.

Creating a basic auth token:

```
$ echo -n diDeuFLcpG91Ydsvxj0SCq4mOohNJTKvp5Q2nRWjDgIA:aspera | base64
2G1EZXVGTGmRz1JWWRzdnhqMFnDcTRtT29oTkpUS3ZwNVEyblJXakRnSUE6YXNwZXJh
```
$ export ASPERA_SCP_TOKEN="Basic ZGlEZXVGTGNwRzl1JWWRzdnhqMFNDcTRtT29oTkpUS3ZwNVEyb1JXakRnSUE6YXNwZXJh"

For 3.5 clients:

$ export ASPERA_SCP_TOKEN="Basic ZGlEZXVGTGNwRzl1JWWRzdnhqMFNDcTRtT29oTkpUS3ZwNVEyb1JXakRnSUE6YXNwZXJh"
$ ascp -i $PWD/asperaweb_id.openssh testfile asp1@node.aspera.us: /

Client-Server Authentication Using Bearer Token and File IDs

Bearer token authentication is a requirement for Aspera Files.

Server setup:

1. Use the node user to create an access key.
2. Identify the storage for this access key, either local or cloud, as described above in Creating and Testing Access Keys.
3. Set the token verification key for the access key.
4. Create permissions for user ID (access ID) luke@aspera.us.
5. Give permissions to top-level storage root:

```
$ curl -d '"file_id":"1",
"access_type":"user","access_id":"luke@aspera.us","access_level":"edit"'
\ -ki -u diDeuFLcpG9IYdsvxj0Scq4mOohNJTKvp5Q2nRwJdgIA:aspera https://localhost:9092/permissions
HTTP/1.1 200 OK
Cache: no-cache
Connection: close
Content-Type: application/json; charset=utf-8
{
"id" : "1",
"file_id" : "1"
}
```

6. Inspect the sample token:

```
$ cat token-bearer-luke@aspera.us.json
{
"user_id": "luke@aspera.us",
"group_ids": ["eng", "emeryville"],
"scope": "node.diDeuFLcpG9IYdsvxj0Scq4mOohNJTKvp5Q2nRwJdgIA:all",
"expires_at": ""
}
```

7. Create a bearer token:

```
$ cat token-bearer-luke@aspera.us.json  | openssl zlib | base64 -w0eJwlkNuOokAARN/5CsMrkx1FLmJisi1jAi1iXoDNZsJAAw3NxFYG2L/
vr7XcenkvVNTSYOaQd
+HbG9nNC10BNn0NQA7z9IQ7+94gRxpH7mzR4RYYmoy8mNCrgAfningQIjUd//
kc1YyWl6Ks1rCm0ALIdh/Wgr5R5UdMN2tZ3NcqtKDFja6mrfZ8nTL1ERbLwOE/
jeBoYyN9J+XlR1B9qtX109bA+X04fqxX16xc49wx/
```
Adding a Node to Aspera Files

Aspera Enterprise Server (the transfer server) provides the "Node API", which is required to connect standalone, on-premise systems to Aspera Files. Basic principles:

- Computer owners access the Node API using a node API user and password.
- Aspera Files users access the Node API using access keys.
- The node API user creates and revokes access keys.
- All transfers and system operations on the computer are run under a local system account called xfer.

Important: To complete this procedure successfully, both the Files application and the browser must be able to verify the node SSL certificate. For details, see Verifying the Certificate on page 168 below.

Prerequisites

- Enterprise Server minimum version: 3.6.0.117909
- SSHD running on port 33001
- Firewall inbound access:
  - TCP/443
  - TCP/33001
  - UDP/33001
- System user:
  - Must be named xfer
  - Must use aspshell
  - Must have read/write access to the local directories or mounts that the application must access
- aspera.conf: no docroot
- aspera.conf: requires docroot restrictions to allow access to the intended storage only. Set the <restriction/> tag as follows:

```xml
  <users>
    <user>
      <name>xfer</name>
      <file_system>
        <access>
          <paths>
            <path>
              <restrictions>
                path
              </restriction>
            </path>
          </paths>
        </access>
      </file_system>
    </user>
  </users>
```

Or:

```
$ sudo /opt/aspera/bin/asnodeadmin --bearer-create --access-key VJDUP5xpZjvq7U8i8U-OVT1HR8TWGKHWOM5tKE84MA --user-id tperrie@aspera.us --group-ids eng,emeryville --expires-at 2020-06-23T13:21:58Z | ruby -rzlib -rbase64 -e 'print Zlib.inflate(Base64.decode64(STDIN.read))'
```
Example path settings for `<restriction/>`:

<table>
<thead>
<tr>
<th>Storage Type</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local storage</td>
<td><code>&lt;restriction&gt;file:///*/&lt;/restriction&gt;</code></td>
</tr>
<tr>
<td>S3 storage</td>
<td><code>&lt;restriction&gt;s3:///*/&lt;/restriction&gt;</code></td>
</tr>
<tr>
<td>Swift storage</td>
<td><code>&lt;restriction&gt;swift:///*/&lt;/restriction&gt;</code></td>
</tr>
<tr>
<td>Azure storage</td>
<td><code>&lt;restriction&gt;azu:///*/&lt;/restriction&gt;</code></td>
</tr>
</tbody>
</table>

Example using `asconfigurator` to set `<restriction>` for S3:

```
$ asconfigurator -x "set_user_data;user_name,xfer;file_restriction,s3://*"
```

- **aspera.conf**: Configure the system user for token authorization. For details, see Configuring Token Authorization With aspera.conf on page 176.
- The system user must use the SSH public key that matches the private key used by the Connect plug-in. Place the key in the `authorized_keys` file and ensure that it’s secure and readable by `xfer`:

```
$ mkdir /home/xfer/.ssh
$ cp /opt/aspera/var/aspera_id_dsa.pub /home/xfer/.ssh/authorized_keys
$ chmod 700 /home/xfer/.ssh
$ chmod 600 /home/xfer/.ssh/authorized_keys
$ chown -R xfer:xfer /home/xfer/.ssh
```

- The public DNS name of the system must be accessible from the following:
  - all internal clients
  - all external clients
  - this computer itself
- **Configuration in `aspera.conf`**:
  - Configure a special extension (for example, `.partial`) to identify a file still being transferred:

```
<default>
  <file_system>
    <partial_file_suffix>.partial</partial_file_suffix>
</file_system>
```

To set this using `asconfigurator`, run the following command, replacing `partial_extension` with your preferred partial file extension:

```
# asconfigurator -x "set_node_data;partial_file_suffix,partial_extension"
```

For example:

```
$ asconfigurator -x "set_node_data;partial_file_suffix,.partial"
```

For more information, see the Aspera Knowledgebase article How to set a file extension for partially transferred files

- Configure activity logging:

```
<server>
  <activity_logging>true</activity_logging>
</server>
```

To set this using `asconfigurator`, run the following command:

```
$ asconfigurator -x "set_server_data;activity_logging,true"
```
• Configure the HTTPS service to use port 443:

```xml
<server>
  <https_port>443</https_port>
</server>
```

To set this using asconfigurator, run the following command:

```bash
# asconfigurator -x "set_server_data;https_port,443"
```

**Note:** You must restart the node services (Aspera NodeD and Aspera Central) after making these changes to the aspera.conf file. For more information, see Transfer Server Configuration on page 103.

Restart these services using the following commands:

```bash
# service asperanoded restart
# service asperacentral restart
```

### Creating a Node User

On the transfer node, create a node user and associate it with the system user `xfer`. The asnodeadmin command requires root or administrator permissions.

**1.** List the current node users:

```bash
# /opt/aspera/bin/asnodeadmin -l
```

**2.** Create a node user by running asnodeadmin as follows:

```bash
# /opt/aspera/bin/asnodeadmin -a -u node_user -p node_user_passwd -x xfer
```

For example:

```bash
# /opt/aspera/bin/asnodeadmin -a -u nuser-001 -p !472830x4n -x xfer
```

**3.** List node users again to verify that step 2 configuration was successful:

```bash
# /opt/aspera/bin/asnodeadmin -l
```

You have now created node user `nuser-001` with password `!472830x4n` and associated it with the system user `xfer`.

**Note:** Before deploying Files in a production environment, be sure to create a node-user password that is secure.

### Verifying the Certificate

Both the Files application and the browser in use must be able to validate the SSL certificate of the node you are adding, including any and all intermediate certificates.

Because an intermediate certificate may be known to your browser but not to the Files application, it is possible for an add-node procedure to seem successful at first, but for workspace operations (for example, create.workspace, create.membership) to fail; in this case, Files displays a warning symbol in the Operations tab of a workspace you attempt to create on this node.

Best practices require that you provide a bundle of chained certificates, typically provided by the certificate authority, concatenated to the signed server certificate. Note that you must list the server certificate before the chained certificates. For example:

```bash
```
This ensures that both the Files application and the browser can verify the node certificates. Use the concatenated file in the `ssl_certificate` directive. For example:

```
server {
  listen              443 ssl;
  server_name         www.example.com;
  ssl_certificate     www.example.com.chained.crt;
  ssl_certificate_key www.example.com.key;
  ...
}
```

For procedures to verify the SSL certificates, see *Installing SSL Certificates* on page 171.

### Creating a New Access Key

On the transfer node, create a new user access key with access to the local directory `/data` by running the following `curl` command:

```
# curl -X POST -ki -u 'node_user:node_user_passwd'
   https://node_host:port/access_keys
   -d json_data
```

To list existing access keys:

```
# curl -X GET -ki -u 'node_user:node_user_passwd'
   https://node_host:port/access_keys
```

To delete an existing access key:

```
# curl -X DELETE -ki -u 'node_user:node_user_passwd'
   https://node_host:port/access_keys/access_key_id
```

Examples of the `curl` command are shown below for local storage, Swift storage, and AWS storage. For examples of other storage types, see *Access Key Authentication* on page 161.

### Using Local Storage

Example of creating a new access key with access to the local directory `/data`:

```
# curl -ki -u 'nuser-001:!472830x4n' https://localhost:9092/access_keys -d
   '{"secret": "873826c61fffeeb0f5a1746af8808cb823651458", "storage":{"type":"local",
   "path":"/data"}}'
```

Example response:

```
{
  "id": "xwXhChBkVppSCgn-o_hoSudXNZHLMQJqFFHGg0LiUA",
  "uri": "file:///data",
  "file_id": "1",
  "system_user": "xfer",
  "license": null,
  "storage": {
    "type": "local",
    "path": "/data"
  }
}
```

### Using Swift Storage
This example uses a file (swift_tenant.json) because the required data is extensive compared to the data required for local storage.

Example request:

```bash
# curl -ki -u 'node_user:node_user_passwd'
https://node_host:port/access_keys
-d @swift_tenant.json
```

Example JSON payload file `swift_tenant.json`:

```json
{
  "secret": "873826c61fffeeb0f5a1746af8808cb823651458",
  "storage": {
    "type": "softlayer_swift",
    "path": "/data",
    "container": "wallball",
    "credentials": {
      "authentication_endpoint": "https://sjc01.objectstorage.softlayer.net/auth/v1.0",
      "username": "RMC05303446-2%3Asascoopdev",
      "api_key": "42f977c63d4a177b381ef6be42d6e3469a99fbeb32a85b3e55e4298c7fe593afb714"
    }
  }
}
```

Example response:

```json
{
  "id": "i9ckg14RHEOEo1pFp2PHFg31C1aE20XZWgAXL48c194A",
  "uri": "swift://sjc01.objectstorage.softlayer.net/wallball/",
  "file_id": "1",
  "system_user": "xfer",
  "license": null,
  "storage": {
    "type": "softlayer_swift",
    "path": "/",
    "container": "wallball",
  }
}
```

**Using AWS Storage**

Use either of the following methods to create an access key:

- Create an access key using S3 credentials. Example request:

  ```bash
  # curl -ki -u 'xfer:passwd' https://localhost:443/access_keys -d
  '{"secret":"!Q@345","storage":
  {"type":"aws_s3","path":"/test","bucket":"aspera","storage_class":"STANDARD","credentials":
  {"access_key_id":"AAKKKL9876RJM7JJJJJ","secret_access_key":"IoppkoijuhytmVw88ikou/hlg9H9MOeiopq4B28"}}}
  ```

- Create an access key using IAM Assume Role. Example request:

  ```bash
  # curl -ki -u 'xfer:passwd' https://localhost:443/access_keys -d
  '{"secret":"!Q@345","storage":
  {"type":"aws_s3","path":"/test","bucket":"aspera","storage_class":"STANDARD","credentials":
  {"assume_role_arn":"arn:aws:iam::2233333444:role/access-s3"}}}
  ```
Example response for both methods:

```plaintext
HTTP/1.1 200 OK
Cache: no-cache
Connection: close
Content-Type: application/json; charset=utf-8
{
  "id" : "xCIOPKlpfw9ok78uKIOHOMn_-N98KGFHc234OPNjR5AdPAA",
  "root_file_id" : "1",
  "license" : null,
  "storage" : {
    "type" : "aws_s3",
    "path" : "/test",
    "endpoint" : "s3.amazonaws.com",
    "bucket" : "aspera",
    "storage_class" : "STANDARD"
  }
}
```

Testing the Access Key

Test the access key by browsing the top-level directory:

```bash
# curl -ki -u 'access_key_id:secret' https://node_host:port/files/1/files
```

For example:

```bash
# curl -ki -u 'xwXhCh...g01iUA:873826c...651458' https://localhost:9092/files/1/files
```

Test by writing to storage:

```bash
# curl -ki -u 'access_key_id:secret' https://node_host:port/files/1/files -d json_commands
```

For example:

```bash
# curl -ki -u 'xwXhCh...g01iUA:873826c...651458' https://localhost:9092/files/1/files -d {"type":"folder", "name", "TEST-001"}
```

Changing the Access Key Secret

Given an access key (my_access_key) and a secret (my_secret), change the secret to my_new_secret by running the following command:

```bash
# curl -ki -u 'my_access_key:my_secret' hostname:port/access_keys/my_access_key -X PUT -d '{"secret":"my_new_secret"}'
```

Installing SSL Certificates

This topic assumes you have a signed root certificate or certificate bundle (root certificate with chained or intermediary certificates) from an authorized Certificate Authority to configure on your Aspera transfer server. If you need to request a certificate from a Certificate Authority (CA), see the article How to Generate a Certificate Signing Request (CSR) in the Aspera Support Knowledgebase.
Procedure Overview

This procedure describes how to install SSL certificates for an Aspera transfer server.

The procedure uses three files:

aspera_server_key.pem
- Created automatically during transfer server installation.
- Resides in the default Aspera installation directory: /opt/aspera/etc
- Contains the default private key.
- In this procedure, you replace the default private key with the new private key generated with the certificate signing request (CSR).

aspera_server_cert.pem
- Created automatically during transfer server installation.
- Resides in the default Aspera installation directory: /opt/aspera/etc
- Contains the default self-signed certificate.
- In this procedure, you replace the default self-signed certificate with the content described in step 3 below.

aspera_server_cert.chain
- You create this file as described in the procedure.
- You place the file in the same directory as aspera_server_key.pem and aspera_server_cert.pem.
- You place the certificate bundle (chained or intermediary certificates) from the CA in this file.

The default filenames and their default locations can be changed by means of settings in the transfer server's aspera.conf file located as follows:

/opt/aspera/etc/aspera.conf

The aspera.conf tags are as follows:

```xml
<http_server>
    ...
    <key_file>path/keyfile.pem</key_file> <!-- key file for asperahttpd -->
    <cert_file>path/certfile.pem</cert_file> <!-- cert file for asperahttpd -->
    ...
</http_server>

<server>
    ...
    <cert_file>path/certfile.pem</cert_file> <!-- cert file for asperanoded -->
    ...
</server>
```

The chain file for asperanoded must always match the location and name of the asperanoded certfile but with the .chain extension.
Install Certificates

1. Back up the default private key and self-signed certificate, using the following commands:

   # cd /opt/aspera/etc
   # cp aspera_server_key.pem aspera_server_key.pem.bak
   # cp aspera_server_cert.pem aspera_server_cert.pem.bak

2. In `aspera_server_key.pem`, replace the existing content with the new private key generated with the certificate signing request (CSR).

3. In `aspera_server_cert.pem`, replace the existing content with the following, in the order shown:
   
   a. the new private key
   b. the server certificate
   c. any chained or intermediary certificates from the CA in order of ascending authority, for example:
      
      intermediary certificate 1
      intermediary certificate 2
      intermediary certificate 3
   d. the root certificate from the CA

4. Create a new file named `aspera_server_cert.chain`. This file must reside in the same directory as the `.pem` files.

   If you have a certificates bundle from the CA, the contents of `aspera_server_cert.chain` must consist of the following, in the order shown:
   
   a. the server certificate
   b. the certificates bundle, which includes the root certificate

   If you do not have a certificates bundle from the CA, the contents of `aspera_server_cert.chain` must consist of the following, in the order shown:
   
   a. the server certificate
   b. any chained or intermediary certificates from the CA in order of ascending authority, for example:
      
      intermediary certificate 1
      intermediary certificate 2
      intermediary certificate 3
   c. the root certificate from the CA

Restart Services

Restart the services `asperacentral`, `asperahttpd`, and `asperanoded` using the following commands:

   # service asperacentral restart
   # service asperahttpd restart
   # service asperanoded restart

Verify Proper Installation

To verify the root certificate and the certificate chain, run the command-line tool `openssl` to connect to the `asperanoded` service. For example, assuming you are using the default node port (HTTPS 9092):

   # /opt/aspera/bin/openssl s_client -connect myserver:9092

The output returned from this command will show a return value of 0 for success or 1 for failure.

Success: The following sample output shows that verification was successful because `verify return` is 0.
Failure: The following sample output shows that verification failed because `verify return` is 1.

```
depth=0 C = US, ST = California, L = Emeryville, O = IBM, OU = Aspera Inc IT Department, CN = *.asperafiles.com
verify error:num=20:unable to get local issuer certificate
verify return:1
```

Note: You must see as many elements in the output as there are certificates in the chain. In the example below, there is one root certificate and two chained certificates, and therefore the output must show three elements to prove the installation was successful.

Success: The following example shows a successful verification for one root certificate and two intermediary certificates in the chain:

```
Certificate chain
  0 s:/C=US/ST=California/L=Emeryville/O=IBM/OU=Aspera Inc IT Department/
    CN=*.asperafiles.com
  i:/C=US/O=Symantec Corporation/OU=Symantec Trust Network/CN=Symantec Class
    3 Secure Server CA - G4
  1 s:/C=US/O=Symantec Corporation/OU=Symantec Trust Network/CN=Symantec Class
    3 Secure Server CA - G4
  i:/C=US/O=VeriSign, Inc./OU=VeriSign Trust Network/OU=(c) 2006 VeriSign,
    Inc. - For authorized use only/CN=VeriSign Class 3 Public Primary
    Certification Authority - G5
  2 s:/C=US/O=VeriSign, Inc./OU=VeriSign Trust Network/OU=(c) 2006 VeriSign,
    Inc. - For authorized use only/CN=VeriSign Class 3 Public Primary
    Certification Authority - G5
  i:/C=US/O=VeriSign, Inc./OU=Class 3 Public Primary Certification Authority
```

Failure: The following example shows an unsuccessful verification, since only the root certificate is displayed.

```
Certificate chain
  0 s:/C=US/ST=California/L=Emeryville/O=IBM/OU=Aspera Inc IT Department/
    CN=*.asperafiles.com
  i:/C=US/O=Symantec Corporation/OU=Symantec Trust Network/CN=Symantec Class
    3 Secure Server CA - G4
```

If verification is unsuccessful, run the following command to inspect your certificate content:

```
# /opt/aspera/bin/openssl x509 -in certificate.crt -text -noout
```

**Setting Up Token Authorization**

When accounts on a transfer server are configured to require token authorization, only transfers initiated with a valid token are allowed to transfer to or from the server. The token authorization requirement can be set for individual
users, entire user groups, or globally for all users. Token authorization can be set independently for incoming transfers and outgoing transfers.

Token authorization is a requirement for initiating transfers with the Shares product.

Set up token authorization for a transfer user as follows:

1. Choose or create the transfer user on the server.
   The examples below use the transfer user aspera_user_1.

2. Log in as the user to ensure that any created files are owned by the user.
   Create the directory .ssh and the file authorized_keys if they don't already exist. For example:

   `/home/aspera_user_1/.ssh/authorized_keys`

3. Append the token-authorization public key to the user's authorized_keys file.
   Aspera provides a public key in the file aspera_id_dsa.pub stored in the following location:

   `/opt/aspera/var/aspera_id_dsa.pub`

4. Ensure that .ssh and .ssh/authorized_keys are owned by the user.
   For example:

   `drwxr-xr-x  2  aspera_user_1  xgroup  4096  Mar  20  2013  .ssh
   -rw-r--r--  1  aspera_user_1  xgroup   674  Mar  20  2013  .ssh/authorized_keys`

5. Make sure the user has no password.
   If the system does not allow this, create a very large password.

6. Make sure the user's login shell is aspshell.
   For information on setting this, see Securing your SSH Server on page 10.

7. Configure the user for token authorization
   To configure user authorization from the GUI, see Configuring Token Authorization from the GUI on page 175.
   To configure user authorization from aspera.conf, see Configuring Token Authorization With aspera.conf on page 176.

   Note:
   Instead of setting authorization for each user individually, you can set it for a group, or set it globally for all users.

8. Create a node user and associate it with the transfer user.
   The examples below use the Node API user nuser.

   `% asnodeadmin -au nuser -x aspera_user_1 -p nuser_passwd`

9. Test the node user:

   `% curl -ki -u nuser:nuser_passwd https://hostname_or_ip:9092/info`

---

**Configuring Token Authorization from the GUI**

Requirements:

- You have created a transfer user on your server.
- You have set up the transfer user with an SSH public key as described in Setting Up Token Authorization on page 174.
The examples below use a transfer user called aspera_user_1.

1. On the main screen of the desktop client, click the Configuration link (upper right).
   This opens the Server Configuration dialog.

2. Select the Users tab and choose a user to configure.
   Alternatively, select the Groups tab and choose a group to configure, or select the Global tab to configure options for all users.

3. In the right panel of the Server Configuration dialog, select the Authorization tab.

4. For Incoming Transfers check the override box. Under Effective Value, select token from the dropdown menu.

5. Similarly, do the same for Outgoing Transfers.

6. For Token Encryption Key, check the override box, and under Effective Value, enter the token encryption key. The encryption key should be a string of random characters (at least 20 recommended).

7. When you're done, click Apply to save the changes, or click OK to save the changes and close the dialog.

   Alternatively, instead of configuring token authorization individually for each user, you can select the Groups tab and apply these settings to groups of users. Or, you can select the Global tab and apply these settings to all users.

---

**Configuring Token Authorization With aspera.conf**

Requirements:
- You have created a transfer user on your server.
- You have set up the transfer user with an SSH public key as described in [Setting Up Token Authorization](#) on page 174.

The examples below use a transfer user called aspera_user_1.

1. Locate aspera.conf and open it with a plain-text editor

   ```
   /opt/aspera/etc/aspera.conf
   ```
2. Add an authorization section for the user

In the following example, the user section for `aspera_user_1` contains an `<authorization>` section that specifies the following:

- a `<transfer>` section specifying that both incoming and outgoing transfers (in and out) should use token encryption
- a `<token>` section with an encryption key, which should be string of random characters (at least 20 recommended)

Alternatively, you can configure token-authorization settings in a `<group>` section to be applied to all users in the group. Or, you can configure the settings in the `<default>` section to apply them globally for all users.

```xml
<user>
  <name>aspera_user_1</name>
  <authorization>
    <transfer>
      <in>
        <value>token</value>
      </in>
      <out>
        <value>token</value>
      </out>
    </transfer>
    <token>
      <encryption_key>gj5o930t78m34ejme9dx</encryption_key>
    </token>
  </authorization>
  <file_system>
    ...
    ...
  </file_system>
  <name/>
</user>
```
Appendix

Restarting Aspera Services

Instructions on restarting Aspera services after configuration changes

If Aspera Central is stopped, or if you have modified the `<central_server>` or `<database>` sections in `aspera.conf`, then you need to restart the service by entering the following command in a Terminal window:

```
$ /etc/init.d/asperacentral restart
```

Optimizing Transfer Performance

Tips about testing and improving your computer's transfer performance.

To verify that your system's FASP transfer can fulfill the maximum bandwidth capacity, prepare a client machine to connect to this computer, and do the following tests:

1. Start a transfer with Fair transfer policy

   On the client machine, open the user interface and start a transfer. Go to the Details to open the Transfer Monitor.

   ![Transfer Monitor](image)

   To leave more network resources for other high-priority traffic, use the Fair policy and adjust the target rate and minimum rate by sliding the arrows or entering values.

2. Test the maximum bandwidth

   **Note:**

   This test will typically occupy a majority of the network's bandwidth. It is recommended that this test be performed on a dedicated file transfer line or during a time of very low network activity.

   Use Fixed policy for the maximum transfer speed. Start with a lower transfer rate and increase gradually toward the network bandwidth.
To improve the transfer speed, you may also upgrade the related hardware components:

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hard disk</td>
<td>The I/O throughput, the disk bus architecture (e.g. RAID, IDE, SCSI, ATA, and Fiber Channel).</td>
</tr>
<tr>
<td>Network I/O</td>
<td>The interface card, the internal bus of the computer.</td>
</tr>
<tr>
<td>CPU</td>
<td>Overall CPU performance affects the transfer, especially when encryption is enabled.</td>
</tr>
</tbody>
</table>

**Log Files**

Locate the log files related to your Aspera product.

**Viewing Logs and Setting Log Preferences**

The application log file includes detailed transfer information and can be useful for review and support requests.

To view the log, go to **Tools > View Log**.

Users can set the logging level for transfers from the **My Preferences** dialog. **My Preferences** can be opened from **Tools > Preferences** or from the **Preferences** button in the upper-right corner of the application window.
The five logging levels to select from are: **Off**, **Error**, **Warn**, **Info**, and **Debug**. The system default is **Info**.

**Redirecting Aspera Logging to a Different Location**

On Linux systems, the application transfer logs are recorded in the system log file. Instead of mixing Aspera logging with system logging, you may want to redirect Aspera logging to a separate log file of your choice.

On RedHat, CentOS, and Debian, the transfer logs are recorded in the following log file:

```
/var/log/messages
```

To redirect Aspera logging, modify `/etc/syslog.conf` (`/etc/rsyslog.conf` in the case of Red Hat or CentOS 6.XA) and add `local2.none` to the `/var/log/messages` line. For example, if you have the following line:

```
*.info;mail.none;authpriv.none;cron.none               /var/log/messages
```

Change it to:

```
*.info;mail.none;authpriv.none;cron.none;local2.none   /var/log/messages
```

Next, forward `local2.info` log messages to your new file. For example, to write to `/var/log/aspera.log`, add the following line just below the line you modified above:

```
local2.info        -/var/log/aspera.log
```

Note that the log file name should be separated from the log facility (`local2.info`) by tab characters, not spaces. Note the hyphen before the log file name; this allows for asynchronous logging.
Next, restart the syslog daemon to have it load the new configuration:

```
# service syslog restart
```

In the case of Red Hat or CentOS 6.X:

```
# service rsyslog restart
```

Now your Aspera log messages should appear in `/var/log/aspera.log` instead of `/var/log/messages`.

On SLES (Suse) systems, the transfer logs are recorded in the following system log file:

```
/var/log/localmessages
```

To redirect Aspera logging, locate the following section in `/etc/syslog-ng/syslog-ng.conf`:

```plaintext
filter f_local { facility(local0, local1, local2, local3, local4, local5, local6, local7); }
destination localmessages { file("/var/log/localmessages"); }
log { source(src); filter(f_local); destination(localmessages); }
```

Modify the section as follows:

```plaintext
filter f_local { facility(local0, local1, local3, local4, local5, local6, local7); };
    filter f_aspera { facility(local2); };

destination localmessages { file("/var/log/localmessages"); };
log { source(src); filter(f_local); destination(localmessages); };

    destination asperalog { file("/var/log/aspera.log"); };
    log { source(src); filter(f_aspera); destination(asperalog); };
```

Then run the following command:

```
# rcsyslog restart
```

Now your Aspera log messages should appear in `/var/log/aspera.log` instead of `/var/log/localmessages`.

To test this, run the following:

```
# logger -p local2.info aspera test
# cat /var/log/aspera.log
```

The `cat` command should display something similar to the following:

```
Jun 13 10:30:33 linux-kua5 root: aspera test
```

**Rotating Your Aspera Log File**

There are several ways to rotate Aspera logs in Linux:

1. You can add `/var/log/aspera.log` to `/etc/logrotate.d/syslog`.
2. You can create an entry for `aspera.log` in `/etc/logrotate.conf`.
3. You can create a separate config file for `aspera.log` in `/etc/logrotate.d/`. 
(1): If you add `/var/log/aspera.log` to the entries in `/etc/logrotate.d/syslog`, your logs will be rotated with the system logs (usually once a week, compressed, and saving the last 10 logs). Add the `aspera.log` file to the `syslog` file as follows:

```
{
    sharedscripts
    postrotate
    /bin/kill -HUP `cat /var/run/syslogd.pid 2> /dev/null` 2> /dev/null ||
    true
    /bin/kill -HUP `cat /var/run/rsyslogd.pid 2> /dev/null` 2> /dev/null ||
    true
    endscript
}
```

(2) and (3): On some servers, there is so much traffic that the logs need to be rotated more often than once a week. In these cases, you can either (2) add a log-rotate section to `/etc/logrotate.conf` (if it exists), or (3) create a separate `/etc/logrotate.d/aspera` configuration file containing the same information. The first example below will compress and rotate 10 logs whenever the size of `/var/log/aspera.log` reaches 100MB. After log rotation is complete, it will run whatever scripts are specified by `postrotate ... endscript`. If you use `/etc/logrotate.conf`, add the configuration after the line "# system-specific logs may also be configured here."

```
/var/log/aspera.log {
    rotate 10
    size 100M
    create 664 root
    postrotate
        /usr/bin/killall -HUP syslogd
    endscript
    compress
}
```

The simple example below will compress and rotate 10 logs once daily. Instead of moving the original log file and creating a new one, the `copytruncate` option tells logrotate to first copy the original log file, then truncate it to zero bytes.

```
/var/log/aspera.log {
    daily
    rotate 10
    copytruncate
    compress
}
```

### Accessing Shares from the GUI

1. To access Shares from the GUI, go to **Connections** and click the **+** button. Enter the following information:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td><a href="https://hostname">https://hostname</a></td>
<td><a href="https://10.0.0.1">https://10.0.0.1</a></td>
</tr>
<tr>
<td>User</td>
<td>Shares user</td>
<td>shares_admin</td>
</tr>
<tr>
<td>Authentication</td>
<td>Shares user password</td>
<td>X45ape34_1</td>
</tr>
</tbody>
</table>

2. Click **Test Connection** to confirm your client application has successfully connected to Shares.
3. Click **Browse** to specify the target directory.
4. Click **OK** to save the connection.

**Product Limitations**

Describes any limitations that currently exist for Aspera transfer server and client products.

- **Path Limit**: The maximum number of characters that can be included in *any* pathname is **4096 characters**.
Troubleshooting

Using the Troubleshooter

Troubleshoot a remote client's problem connecting to your server.

You can use the transfer application's troubleshooting tool to verify user account security configuration on your computer.

To use the troubleshooting tool, launch the application and select Help > Troubleshoot. The troubleshooter will identify potential problems with your Aspera software configuration.

Clients Can't Establish Connection

Troubleshoot the problem that your clients cannot connect to your IBM Aspera Enterprise Server.

The following diagram shows the troubleshooting procedure if clients can't establish a FASP transfer connection to your Enterprise Server. Follow the instructions to identify and resolve problems:
1. Test SSH ports

   To verify the SSH connection port, on the client machine, open a Terminal or a Command Prompt, and use the `telnet` command to test it. For example, to test connection to a computer (10.0.1.1) through a port (TCP/33001), use this command:

   ```
   $ telnet 10.0.1.1 33001
   ```

   If the client cannot establish connections to your Enterprise Server, verify the port number and the firewall configuration on your Enterprise Server machine.

2. Test UDP ports

   If you can establish an SSH connection but not a FASP file transfer, there might be a firewall blockage of FASP's UDP port. Please verify your UDP connection.

3. Verify SSH service status

   If there is no firewall blockage between the client and your Enterprise Server, on the client machine, try establishing a SSH connection in a Terminal or a Command Prompt: (Enterprise Server address: 10.0.1.1, TCP/33001)

   ```
   $ ssh aspera_user_1@10.0.1.1 -p 33001
   ```

   If the SSH service runs normally, the client should see a message prompting to continue the connection or for a password. However, if you see a "Connection Refused" message, which indicates that the SSH service isn't running, review your SSH service status. Ignore the "permission denied" message after entering the password, which is discussed in next steps.

4. Applied authentication method is enabled in SSH

   If you can establish a SSH connection, but it returns "permission denied" message, the SSH Server on your Enterprise Server may have password authentication disabled:

   ```
   Permission denied (publickey,keyboard-interactive).
   ```

   Open your SSH Server configuration file with a text editor:

   ```
   /etc/ssh/sshd_config
   ```
To allow public key authentication, add or uncomment the `PubkeyAuthentication yes`. To allow password authentication, add or uncomment `PasswordAuthentication yes`. Here is a configuration example:

```
... 
PubkeyAuthentication yes 
PasswordAuthentication yes 
...
```

To reload SSH service, execute the command:

<table>
<thead>
<tr>
<th>OS Version</th>
<th>Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>RedHat</td>
<td><code>$ sudo service sshd restart</code></td>
</tr>
<tr>
<td>Debian</td>
<td><code>$ sudo /etc/init.d/ssh restart</code></td>
</tr>
</tbody>
</table>

5. Verify that the user credentials are correct, and has sufficient access permissions to its docroot

To make sure that the client can establish a SSH connection to your Enterprise Server, and has correct system user credentials, use this command on the client machine: (Connect Server address: 10.0.1.1, login: aspera_user_1/aspera)

```bash
$ ssh aspera_user_1@10.0.1.1 -p 33001
aspera_user_1@10.0.1.1's password:
```

Enter the user's password when prompted. If you see "Permission denied" message, you may have a wrong user credentials, or the user has docroot configured and doesn't have sufficient access permissions to access it. Refer to Setting Up Users on page 55 or Setting Up Transfer Users (Terminal) on page 59 for instructions about setting up the user account, and review the user's docroot directory's permissions.

If you still encounter connection problems after going through these steps, contact Technical Support on page 187.
Technical Support

For further assistance, you may contact Aspera through the following methods:

<table>
<thead>
<tr>
<th>Method</th>
<th>Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Email</td>
<td><a href="mailto:support@asperasoft.com">support@asperasoft.com</a></td>
</tr>
<tr>
<td>Phone (U.S.)</td>
<td>+1 (510) 849-2386, option 2</td>
</tr>
<tr>
<td>Phone (Europe)</td>
<td>+44 (0) 207 993 6653</td>
</tr>
<tr>
<td>Request Form</td>
<td><a href="https://support.asperasoft.com/anonymous_requests/new/">https://support.asperasoft.com/anonymous_requests/new/</a></td>
</tr>
<tr>
<td></td>
<td>You can use this form to request help from Aspera Technical Support.</td>
</tr>
</tbody>
</table>

Support availability:

<table>
<thead>
<tr>
<th></th>
<th><strong>Standard Support</strong></th>
<th><strong>Premium Support</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Hours</strong></td>
<td>8:00am – 6:00pm, weekdays (Monday-Friday)</td>
<td>24 hours a day, 7 days a week</td>
</tr>
<tr>
<td>(Pacific Standard Time, GMT-8)</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Unavailable Dates</strong></td>
<td>Weekends (Saturday, Sunday)</td>
<td>Holidays: See support.asperasoft.com.</td>
</tr>
<tr>
<td></td>
<td>Holidays: See support.asperasoft.com.</td>
<td></td>
</tr>
</tbody>
</table>
Legal Notice

© 2007-2010-2011-2012-2013-2014-2016 Aspera, Inc., an IBM Company. All rights reserved.

Licensed Materials - Property of IBM
5725S58
US Government Users Restricted Rights- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Aspera, the Aspera logo, and FASP transfer technology are trademarks of Aspera, Inc., registered in the United States. Aspera Connect Server, Aspera Drive, Aspera Enterprise Server, Aspera Point-to-Point, Aspera Client, Aspera Connect, Aspera Cargo, Aspera Console, Aspera Orchestrator, Aspera Crypt, Aspera Shares, the Aspera Add-in for Microsoft Outlook, and Aspera Faspex are trademarks of Aspera, Inc. All other trademarks mentioned in this document are the property of their respective owners. Mention of third-party products in this document is for informational purposes only. All understandings, agreements, or warranties, if any, take place directly between the vendors and the prospective users.