Aspera

# Aspera APIs

## [Aspera on Cloud API](https://developer.ibm.com/apis/catalog/aspera--aspera-on-cloud-api/Introduction)

* 1. The Aspera on Cloud API (formerly the "Files API") supports a RESTful platform API. The AoC API allows developers to create web applications and automation tools for use with the Admin, Files, and Packages apps in AoC. It works closely with the Connect API and Node API to interwork between AoC and the underlying Aspera transfer node.
  2. We can use it to manage users (should be signed up on Aspera), clients, workspaces, packages, nodes, dropboxes,

## [Aspera Automation API](https://developer.ibm.com/apis/catalog/aspera--aspera-automation-api/Introduction)

* 1. A RESTful API that lets you create automated workflows that initiate file transfers, integrate with web services via API calls and notify your stakeholders. Automation can monitor your file storage system for event triggers; for example, a file uploaded to a shared folder can initiate a transfer, and the completed transfer can trigger an email notification.
  2. We can do an event trigger when a file is uploaded to a shared folder to send an email to notify the user (the user should be added) or call an API.

## [Aspera Activity API](https://developer.ibm.com/apis/catalog/aspera--aspera-activity-api/Introduction)

* 1. The Activity API returns data from Aspera objects related to performance, history, and usage of file transfers in Aspera on the Cloud.

## [Aspera Node API](https://developer.ibm.com/apis/catalog/aspera--aspera-node-api/Introduction)

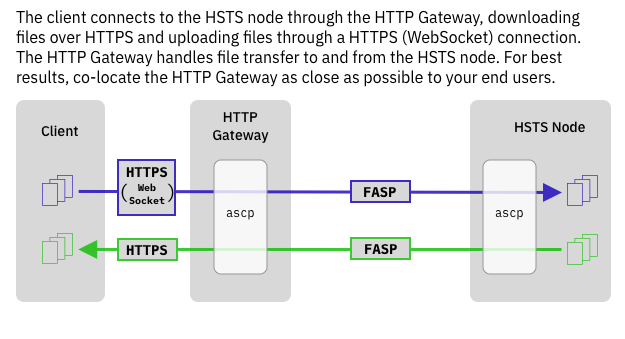
* 1. The Node API is a REST API web service that runs on an Aspera server (node) and enables you to do the following:
     1. Manage user access to the node’s file system and transfer capabilities.
     2. Upload and download files and directories from the node using the FASP protocol or HTTP/HTTPS (by HTTP fallback).
     3. Initiate and control transfers and manage files (start, stop, resume, and reconfigure transfers; create, delete, and rename files).
     4. Retrieve information about the node, including available space, transfer events, and transfer volume.

## [Aspera ATS API](https://developer.ibm.com/apis/catalog/aspera--aspera-ats-api/Introduction)

* 1. Is an authenticated, REST API that enables users to manage client access to their cloud storage. Once a client has a transfer service access key for the cloud storage. (Needs to High-Speed Transfer Client)

## [Aspera HTTP Gateway Javascript SDK](https://developer.ibm.com/apis/catalog/aspera--aspera-http-gateway-javascript-sdk/Introduction) (What we need)

* 1. The HTTP Gateway SDK is designed to provide third-party developers with tools to establish a connection with, upload files to, and download files from an HTTP Gateway service.
  2. Integrating the IBM Aspera HTTP Gateway Javascript SDK with your web application allows end users to upload and download files to a node – a server running IBM Aspera High-Speed Transfer Server (HSTS) – without IBM Aspera Connect.
  3. Instead of using Connect to make transfers with the web application, users now make transfers with the HTTP Gateway, which acts as a reverse proxy connecting two legs of a transfer:
     1. Client (such as a user through a web application) to HTTP Gateway, using HTTP, HTTPS, or WebSocket
     2. HTTP Gateway to node, through FASP, using the ascp binary
  4. The HTTP Gateway is more accessible to web application users because they don’t have first to download Connect. However, Connect is still faster and more powerful because it leverages FASP on both legs of the transfer.
  5. Repo has some examples for download and upload transfer operations: [here](https://github.com/laurent-martin/aspera-api-examples)
  6. To run transfers, we will need:
     1. bucket name
     2. storage endpoint
     3. API key
     4. crn
     5. auth endpoint
  7. Or it is also possible to use:
     1. bucket name
     2. region
     3. service credentials: follow: [get service credentials](https://www.rubydoc.info/gems/aspera-cli#using-service-credential-file)
  8. For authentication between the client and HTTP gateway, we could use JWT and implement it from our side.



## [Aspera Connect SDK](https://developer.ibm.com/apis/catalog/aspera--aspera-connect-sdk/Introduction) (What we need)

* 1. With the IBM Aspera Connect SDK, you can embed IBM Aspera capabilities in your web applications (It needs a web server).  
     NPM link: [from here](https://www.npmjs.com/package/@ibm-aspera/connect-sdk-js)  
     Documentation: [here](https://ibm.github.io/aspera-connect-sdk-js/)  
     Repo has some examples for download and upload transfer operations: [here](https://github.com/laurent-martin/aspera-api-examples)
  2. To run transfers, we will need:
     1. bucket name
     2. storage endpoint
     3. API key
     4. crn
     5. auth endpoint
  3. Or it is also possible to use:
     1. bucket name
     2. region
     3. service credentials: follow: [get service credentials](https://www.rubydoc.info/gems/aspera-cli#using-service-credential-file)
  4. For authentication between the client and the application, we could use JWT and implement it from our side or using.

# 

# High-Speed Transfer Server

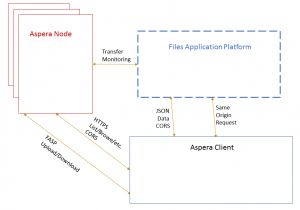
[where you can find information about how to install, maintain, and use the High-Speed](https://www.ibm.com/docs/en/ahts/4.4.x?topic=high-speed-transfer-server-admin-guide-linux)

# Node

A *node* is a server on which Aspera software is installed. The *Node API* allows you to do the following:

1. Manage user access to the node’s file system and transfer capabilities.
2. Create and delete files and folders.
3. Upload and download files and folders to and from the node using the FASP protocol or HTTP/HTTPS.
4. Query what files and directories are on the server.
5. Query what transfers have been made to and from the server.
6. Modify the attributes of a transfer while it is in progress.
7. Get transfer events that are occurring on the server.
8. Create access keys for permissions.

## Architecture



As shown in the drawing above, there are three main architectural components:

1. Application Servers, which consist of the Aspera on Cloud front-end and back-end servers. The AoC web application is retrieved from the AoC front-end servers. In the diagram, "Files Application Platform" represents the AoC servers.
2. Node Servers, (sometimes called “Nodes”) handle files, folders, access to files, file listings, file transfers… everything about files. There may be multiple clusters of Node Servers
3. Clients are everything that uses the APIs in the other servers. The AoC web application is a client, Aspera Drive is a client, etc.