Course Guide

IBM Aspera High-Speed Transfer Server Administration

Course code WT011 / ZT011 ERC 1.0

IBM Training

February 2020 edition

Notices

This information was developed for products and services offered in the US.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml.

Contents

Course description ix

Agenda x

Unit 1. Understanding IBM Aspera FASP 1-1
Unit objectives 1-2
Topics 1-3
1.1. TCP performance issues 1-4
TCP performance issues 1-5
Review: TCP and UDP protocols 1-6
Latency and RTT 1-7
TCP sliding window 1-8
TCP: Router queuing delay 1-9
TCP: LAN performance 1-10
TCP: Wide area network (WAN) performance 1-11
Bandwidth: Not the problem 1-12
1.2. FASP protocol. 1-13
FASP protocol 1-14
FASP: Overview 1-15
Transfer rate comparison 1-16
FASP: Transfer initiation 1-17
Optimal performance: Packet loss 1-18
Reliable and secure transfers 1-19
Adaptive rate control 1-21
Bandwidth control 1-22
1.3. Transfer rate factors 1-23
Transfer rate factors 1-24
Transfer rate factors 1-25
Transfer policies 1-27
Vlinks 1-29
FASP security and reliability 1-31
What you learned 1-33
Unit summary 1-34
Review questions (1 of 2) 1-35
Review questions (2 of 2) 1-36
Review answers (1 of 2) 1-37
Review answers (2 of 2) 1-38

Unit 2. Overview of IBM Aspera Software 2-1
Unit objectives 2-2
Topics 2-3
IBM Aspera products 2-4
Transfer servers 2-6
IBM Aspera High-Speed Transfer Server 2-7
IBM Aspera Sync 2-9
IBM Aspera Proxy Server 2-11
Client software 2-12
Web applications 2-13
IBM Aspera Faspex 2-14
IBM Aspera Shares 2-15

IBM Aspera Console . . . 2-16
Aspera on Cloud: Overview . . . 2-17
Aspera Product Integration: Overview . . . 2-18
The *ascp* utility . . . 2-20
The *ascp* utility: Overview . . . 2-21
Transfer servers and *ascp* . . . *2-22*
Basic *ascp* connections . . . 2-23
Web applications and *ascp* . . . *2-25*
Aspera and directory services . . . 2-26
Aspera and directory services . . . 2-27
Aspera and SAML . . . 2-29
What you learned . . . 2-31
Unit summary . . . 2-32
Review Questions (1 of 2) . . . 2-33
Review questions (2 of 2) . . . 2-34
Review answers (1 of 2) . . . 2-35
Review answers (2 of 2) . . . 2-36

Unit 3. Installing IBM Aspera High-Speed Transfer Server . . . 3-1
Unit objectives . . . 3-2
Topics . . . 3-3
Prepare operating system . . . 3-4
System prerequisites . . . 3-5
Configure SELinux (if “enforcing”) . . . 3-7
Installation process . . . 3-9
Firewall configuration . . . 3-10
Configure SSH . . . 3-12
Install Aspera software . . . 3-15
Install IBM Aspera High-Speed Transfer Server software . . . 3-16
Transfer server authentication . . . 3-18
Verify transfer connection . . . 3-20
Upgrading, downgrading, and reinstalling Aspera software . . . 3-22
Upgrading or Downgrading Aspera software: Overview . . . 3-23
Removing Aspera software . . . 3-25
Configure Aspera logging . . . 3-27
Redirect Aspera logging to *aspera.log* . . . *3-28*
Log file rotation . . . 3-30
What you learned . . . 3-31
Unit summary . . . 3-32
Review questions (1 of 2) . . . 3-33
Review questions (2 of 2) . . . 3-34
Review answers (1 of 2) . . . 3-35
Review questions (2 of 2) . . . 3-36
Lab Exercise 1 . . . 3-37

Unit 4. Configuring IBM Aspera High-Speed Transfer Server . . . 4-1
Unit objectives . . . 4-2
Aspera GUI transfers . . . 4-3
Server GUI menus and pages . . . 4-5
GUI menu options . . . 4-7
GUI access to Aspera server logs . . . 4-9
Creating GUI connections . . . 4-11
Connections GUI – transfer parameters . . . 4-14
Connections GUI: tracking parameters . . . 4-16
Connections GUI: filters . . . 4-17
Connections GUI: security . . . 4-18

Connections GUI: File Handling parameters 4-19
Importing and exporting connections 4-21
Server configuration – overview 4-22
Document root (docroot) parameters 4-24
Authorization parameters 4-26
Bandwidth parameters 4-28
Network parameters 4-30
File handling parameters 4-31
File handling parameters (2) 4-33
Database parameters 4-35
Transfer server parameters 4-36
HTTP fallback parameters 4-37
Vlinks 4-38
Creating Vlinks 4-39
What you learned 4-41
Unit summary 4-42
Review questions (1 of 2) 4-43
Review questions (2 of 2) 4-44
Review answers (1 of 2) 4-45
Review answers (2 of 2) 4-46
Lab Exercise 2 4-47

Unit 5. Managing Aspera users and groups 5-1
Unit objectives 5-2
System user versus transfer user 5-3
Creating transfer user accounts 5-4
Global versus user parameters 5-6
Transfer groups 5-8
Transfer groups precedence 5-9
What you learned 5-11
Unit summary 5-12
Review questions 5-13
Review answers 5-14
Lab Exercise 3 5-15

Unit 6. Using command-line operations 6-1
Unit objectives 6-2
Aspera Transfer Server command-line utilities 6-3
asuserdata: List parameters and *asconfigurator* syntax 6-4
The *asconfigurator* utility 6-5
asconfigurator command syntax 6-7
Examples of *asconfigurator* usage 6-9
ascp: Command-line transfers 6-10
Command-line transfer: general examples 6-11
Environment variables 6-12
What you learned 6-13
Unit summary 6-14
Review questions 6-15
Review answers 6-16
Lab Exercise 5 6-17

Unit 7. Configuring advanced features 7-1
Unit objectives 7-2
Authentication and authorization 7-3
Configure SSL/TLS certificate 7-5
Public keys: Transfer user on transfer server 7-7

Configuring new RSA key pair . . . 7-8
Enable HTTP Fallback . . . 7-10
Pre / Post file processing: Overview . . . 7-12
Implementing pre post processing . . . 7-13
The aspera-prepost file . . . 7-15
Prepost script examples . . . 7-16
Built-in pre/post email notification . . . 7-18
Aspera Node API: Overview . . . 7-20
Aspera Central & Node API . . . 7-21
Node API REST endpoints . . . 7-22
Typical Node API functions . . . 7-23
Node API requests and responses . . . 7-24
Enable Node API in aspera.conf . . . 7-26
Manage Node API (each server) . . . 7-27
Automating transfers: Overview . . . 7-29
Configuring Hot Folders . . . 7-31
Managing Hot Folders . . . 7-35
Hot Folders versus Watch Folders . . . 7-36
Aspera Watch Folder: Overview . . . 7-38
Watch Service & Watch Folder daemons . . . 7-40
Choosing user accounts for Watch Folder services . . . 7-42
Prepare system for Watch Folders . . . 7-44
Watch Folder restrictions . . . 7-46
Watch Folders from command line . . . 7-48
JSON configuration file . . . 7-50
The Watch Folder GUI . . . 7-51
Configuring Watch Folders from GUI . . . 7-53
Configuring Watch Folders from GUI (Cont.) . . . 7-55
Managing Watch Services from GUI . . . 7-56
What you learned . . . 7-57
Unit summary . . . 7-58
Review questions (1 of 2) . . . 7-59
Review questions (2 of 2) . . . 7-60
Review answers (1 of 2) . . . 7-61
Review questions (2 of 2) . . . 7-62
Lab Exercise 6 . . . 7-63

Unit 8. Routine maintenance tasks . . . 8-1
Unit objectives . . . 8-2
Potential transfer bottlenecks . . . 8-3
Transfer Server maintenance tasks . . . 8-4
Aspera logging: Overview . . . 8-5
Reading transfer logs . . . 8-7
Transfer Server log: Performance . . . 8-8
What you learned . . . 8-10
Unit summary . . . 8-11
Review questions . . . 8-12
Review answers . . . 8-13

Unit 9. Course summary . . . 9-1
Unit objectives . . . 9-2
Course objectives . . . 9-3
IBM badge . . . 9-4
IBM Professional Certifications . . . 9-5
Other learning resources (1 of 4) . . . 9-6
Other learning resources (2 of 4) . . . 9-7

Other learning resources (3 of 4) 9-8
Other learning resources (4 of 4) 9-9
Unit summary 9-10
Course completion 9-11

Trademarks

The reader should recognize that the following terms, which appear in the content of this training document, are official trademarks of IBM or other companies:

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide.

The following are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide:

AIX®	Aspera®	DB™
FASP®	IBM Cloud™	Initiate®
Notes®	Power®	System z®
Tivoli®		

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

VMware is a registered trademark or trademark of VMware, Inc. or its subsidiaries in the United States and/or other jurisdictions.

Other product and service names might be trademarks of IBM or other companies.

Course description

IBM Aspera High-Speed Transfer Server Administration

Duration: 3 days

Purpose

This course is intended to teach the necessary knowledge and skills to install, configure, and use the IBM Aspera High-Speed Transfer Server.

Audience

This course is intended for administrators the IBM Aspera High-Speed Transfer Server.

Prerequisites

- Fundamental knowledge of using Windows and Linux operating systems
- Basic understanding of networking

Objectives

- Describe the operation of the FASP protocol
- Outline the functions of various Aspera software products
- Explain Aspera configuration parameters and assign their values
- Create and manage Aspera users and groups
- Perform file transfers using the Aspera GUI and from the command line
- Implement support for Aspera Node API
- Configure Hot Folders and Aspera Watch Service
- Execute basic troubleshooting tasks for common problems

Agenda

Note

The following unit and exercise durations are estimates, and might not reflect every class experience.

Day 1

(00:15) Course introduction
(02:00) Unit 1. Understanding IBM Aspera FASP
(01:30) Unit 2. Overview of IBM Aspera Software
(02:30) Unit 3. Installing IBM Aspera High-Speed Transfer Server
(02:00) Exercise 1. Installing IBM Aspera High-Speed Transfer Server

Day 2

(02:00) Unit 4. Configuring IBM Aspera High-Speed Transfer Server
(02:00) Exercise 2. Configuring IBM Aspera High-Speed Transfer Server
(01:00) Unit 5. Managing Aspera users and groups
(01:00) Exercise 3. Managing Aspera users and groups
(02:00) Unit 6. Using command-line operations

Day 3

(02:00) Exercise 4. Using command-line operations
(02:00) Unit 7. Configuring advanced features
(02:30) Exercise 5. Configuring advanced features
(01:30) Unit 8. Routine maintenance tasks

Unit 1. Understanding IBM Aspera FASP

Estimated time

02:00

Overview

This unit describes the operation of the FASP protocol and how it compares with traditional file transfer protocols.

IBM Training

Unit objectives

- Explain the fundamental performance problem of TCP-based transfers
- Outline the process FASP uses to determine optimal packet size
- Highlight the difference between how FASP and TCP manage packet loss
- Describe the adaptive rate control process used by FASP
- Identify the factors that influence transfer rates
- Clarify the value of using Vlinks

Figure 1-1. Unit objectives

IBM Training

Topics

- TCP performance issues
- FASP protocol
- Transfer Rate Factors

Figure 1-2. Topics

1.1. TCP performance issues

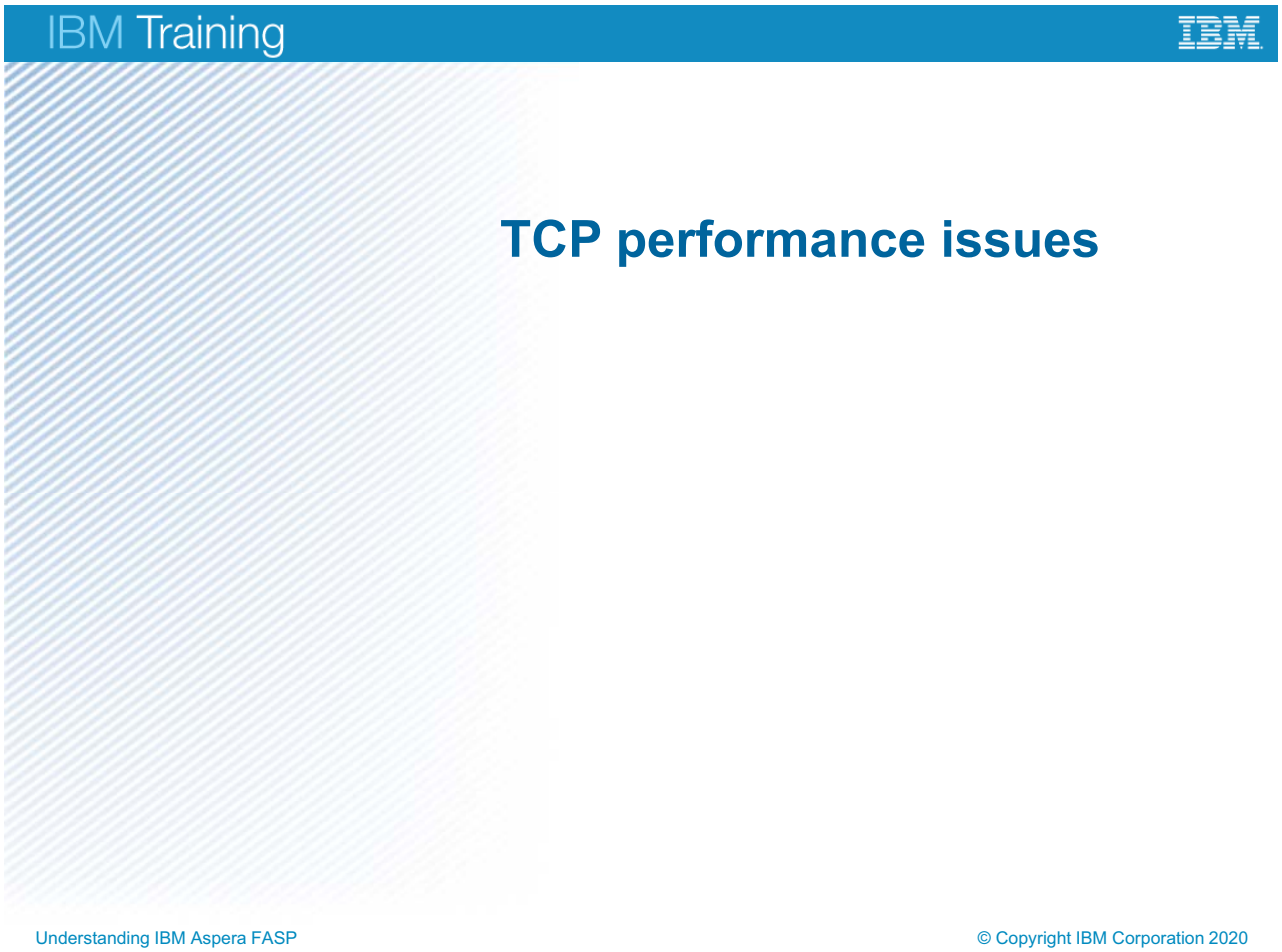
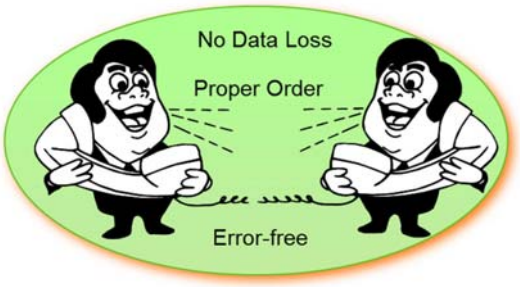


Figure 1-3. TCP performance issues

IBM Training

Review: TCP and UDP protocols

Transmission Control Protocol (TCP)



Good

- Establishes a connection (sessions)
- Reliable, guaranteed delivery
- Secure delivery
- Adaptive rate control

Bad

- Designed for low-bandwidth networks
- Performance degrades over distance and with packet loss
- Does not scale with increased bandwidth

User Datagram Protocol (UDP)

Bad

- Connectionless
- No guarantee of delivery or order

Good

- Performance does NOT degrade over distance or with packet loss

Figure 1-4. Review: TCP and UDP protocols

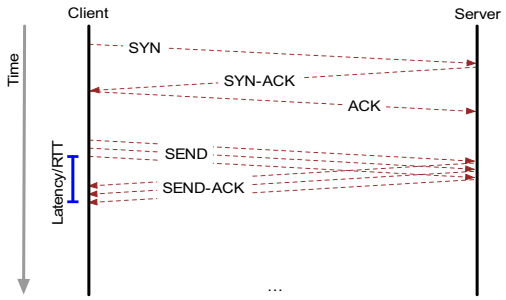
The Transmission Control Protocol (TCP) is the underlying protocol that is used for most file transfers programs, such as FTP, RCP, and SCP. TCP was designed to provide a virtual connection between two systems to ensure that data was received reliably, regardless of the network conditions. TCP uses several mechanisms to prevent data loss and deliver data in the order it should be to provide an error-free connection.

However, when TCP was developed, physical networks were low bandwidth and tended to have less than reliable connections, thus requiring the mechanisms provided by TCP. Unfortunately, these mechanisms for reliability come at a high cost on today's higher performance networks. Performance of TCP-based transfer programs degrades significantly over longer distances and with increased packet loss (commonly associated with increased network traffic). The result is that these TCP-based programs do not scale well, even when the network bandwidth is dramatically increased.

The User Datagram Protocol (UDP) was designed to provide a best effort to deliver, but without any mechanisms to ensure that the data is received in the same condition it was sent. In other words, UDP doesn't have the processes that are associated with connections and data delivery guarantees. Unfortunately, this lack of ensuring data is received properly renders it unacceptable as the sole basis for file transfer programs.

IBM Training

Latency and RTT



Round Trip Time (RTT) = time to receive acknowledgment

Latency is a problem on WAN, not a LAN

Latency increases with distance

TCP does NOT consider RTT

BUT

Higher latency—more TCP packets sent before acknowledgment received

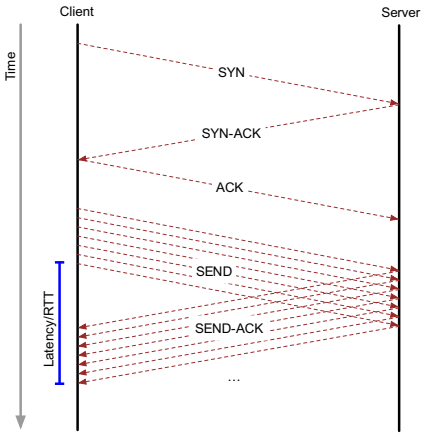


Figure 1-5. Latency and RTT

The slide illustrates the concept of Round Trip Time (RTT) and the role of latency in determining the RTT. Latency is a given on all networks, and is problematic for wide area networks (WANs) which frequently introduce more latency, depending upon the distance between connecting endpoints.

The rate control algorithm that is used by TCP does not directly consider RTT, but excessive RTT reduces TCP's transfer rates. The sending TCP program expects to receive an acknowledgment (ACK) of data transferred within a set timeframe. If the ACK is not received, the data is retransmitted, and the amount of data that is sent before receiving an ACK is reduced. Even if the receiving endpoint successfully receives the data, if the RTT is too great, the sending endpoint retransmits, thus increasing congestion and reducing the transfer rate.

IBM Training

TCP sliding window

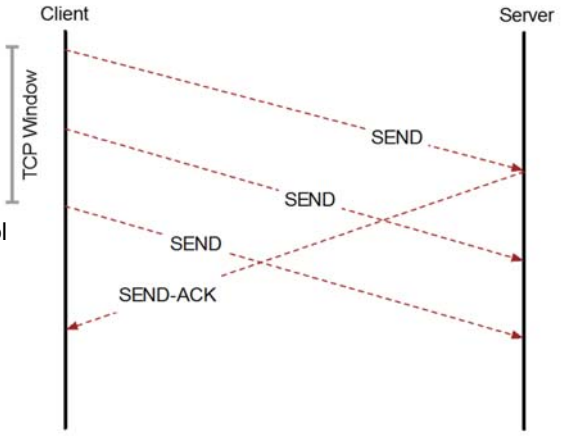
Sliding Window

Data sent but not acknowledged

Window size set by congestion and flow control algorithms

Window “slides” as data is sent

Unacknowledged data resent when timer expires



Flow control

No data loss
Proper order
Error-free

Congestion control

Packet or “ACK” loss = sender reduces window 50% or 100%

Flow and congestion control managed by SENDER

Figure 1-6. TCP sliding window

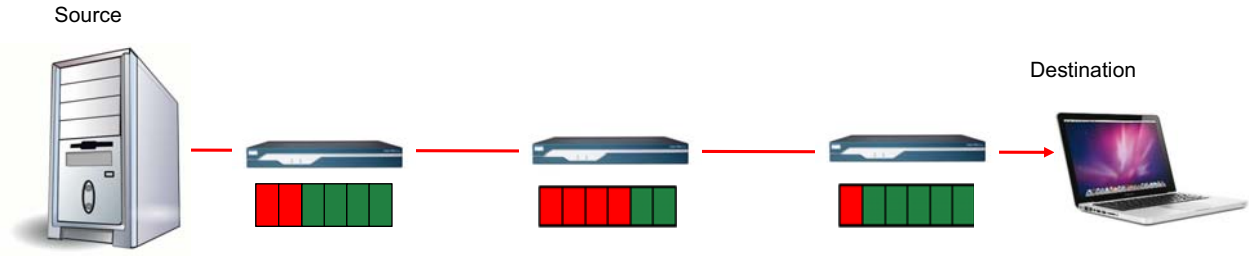
TCP implements the concept of a sliding window to control how much data can be sent without receiving an acknowledgment (ACK) packet. The size of the sliding window is a function of TCP’s Congestion and Flow Control algorithms. When ACKs are received, the window moves to include data that was not transmitted.

If an ACK is not received before the ACK timer expires, data is retransmitted. Additionally, if packet loss is detected or the ACK is not received, the size of window is reduced, initially by 50% and up to 100% in an attempt to account for network congestion.

This window reduction leads to the TCP spiked performance characteristic, which can seriously diminish the realized transfer rate.

IBM Training

TCP: Router queuing delay



Busy/slow routers take longer to process packets = increased latency

Single slow router can impact end-to-end throughput

Full queue = dropped packets = decreased “sliding window” = reduced transfer rate

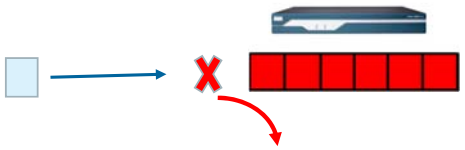


Figure 1-7. TCP: Router queuing delay

Routers all use buffers to receive and process data packets. This behavior introduces another issue that can negatively impact TCP transfers.

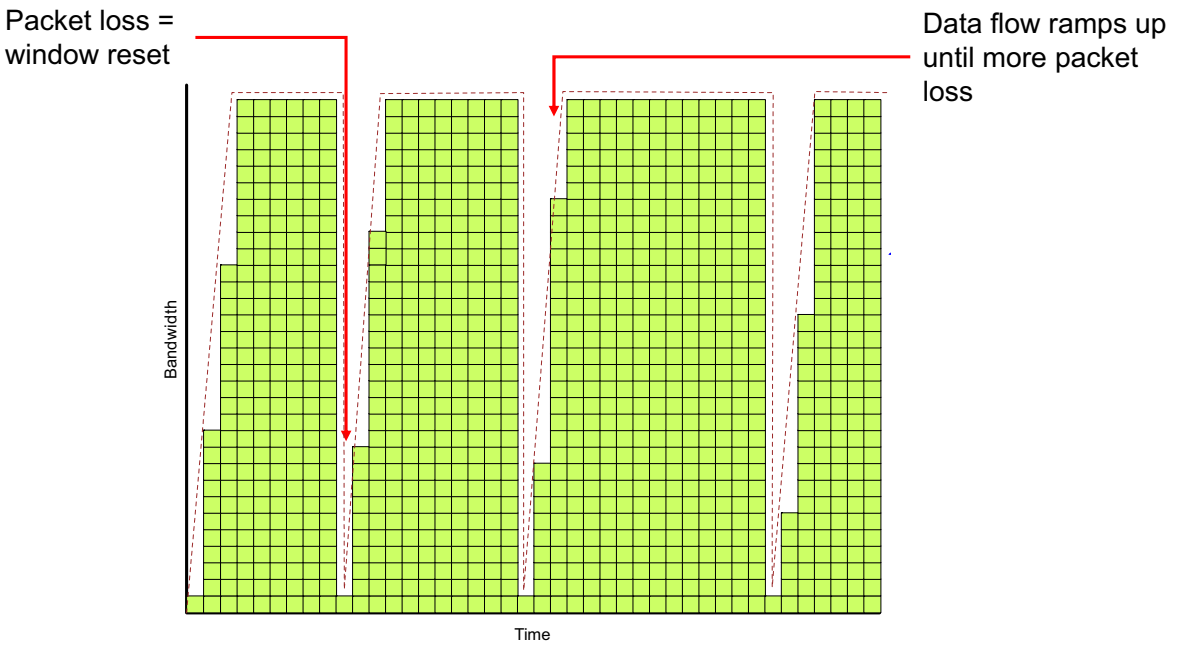
As routers get busier, their buffer queues get larger, resulting in an increase in observed latency.

Routers with slower computing capability also take longer to process packets through their queue, resulting in increased latency.

When a router’s buffer is full, it drops inbound packets, resulting in more issues for TCP transmitted data.

IBM Training

TCP: LAN performance



Low packet loss & latency = good performance
TCP performance is good on local area network (LAN)

Figure 1-8. TCP: LAN performance

TCP transfers usually do well when with low latency and minimal packet loss, as found in LAN environments. Even though rate is reduced periodically, generally the performance is acceptable on a LAN.

IBM Training

TCP: Wide area network (WAN) performance

WAN environment

Greater distance and hops = more packet loss and latency

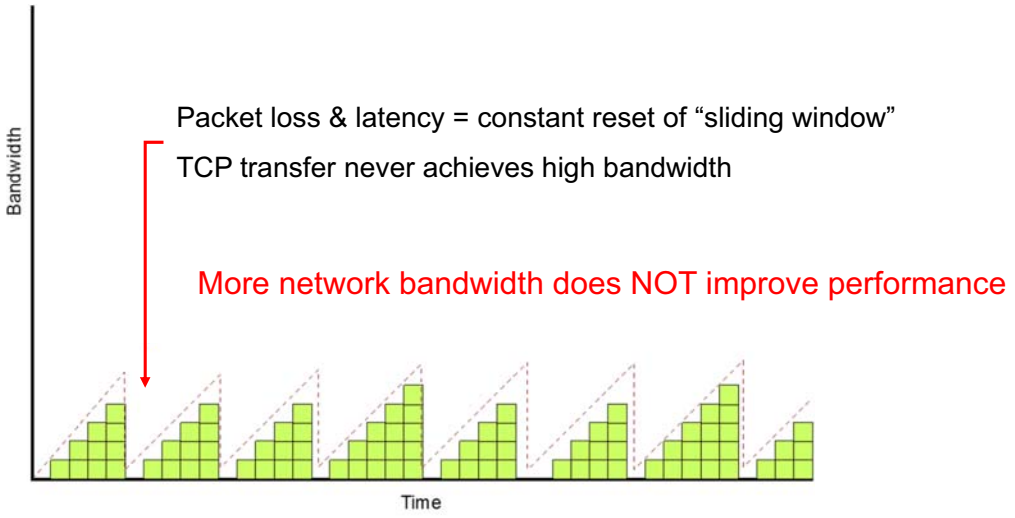


Figure 1-9. TCP: Wide area network (WAN) performance

While LANs typically provide low latency and low packet loss, WANs are frequently the exact opposite. As distance increases, more latency is introduced. And the latency that is introduced with distance is highly correlated with increase packet loss. These conditions have an adverse effect on TCP-based file transfers. Because TCP’s sliding window is constantly being reset, the actual transfer rate never achieves a high rate.

This negative impact is the result of latency and packet loss, and even increasing bandwidth of the network does not really increase the realized transfer rate.

IBM Training

Bandwidth: Not the problem

More bandwidth doesn’t solve the problem!

TCP is the problem!

Sliding Window

- Flow Control
- Congestion Control
- Queuing Delays

What about UDP?

Good performance, BUT

- No reliability
- Lack of control

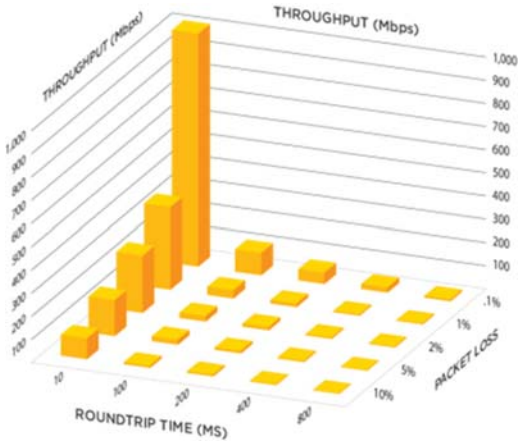


Figure 1-10. Bandwidth: Not the problem

The previous materials discussed the problems that are associated with TCP-base file transfers, and determined that TCP’s sliding window is the major factor in TCP’s poor WAN performance.

The materials also identified that UDP can provide good performance, but lacks the ability to ensure reliable data delivery and does not provide any type of congestion or flow control.

1.2. FASP protocol

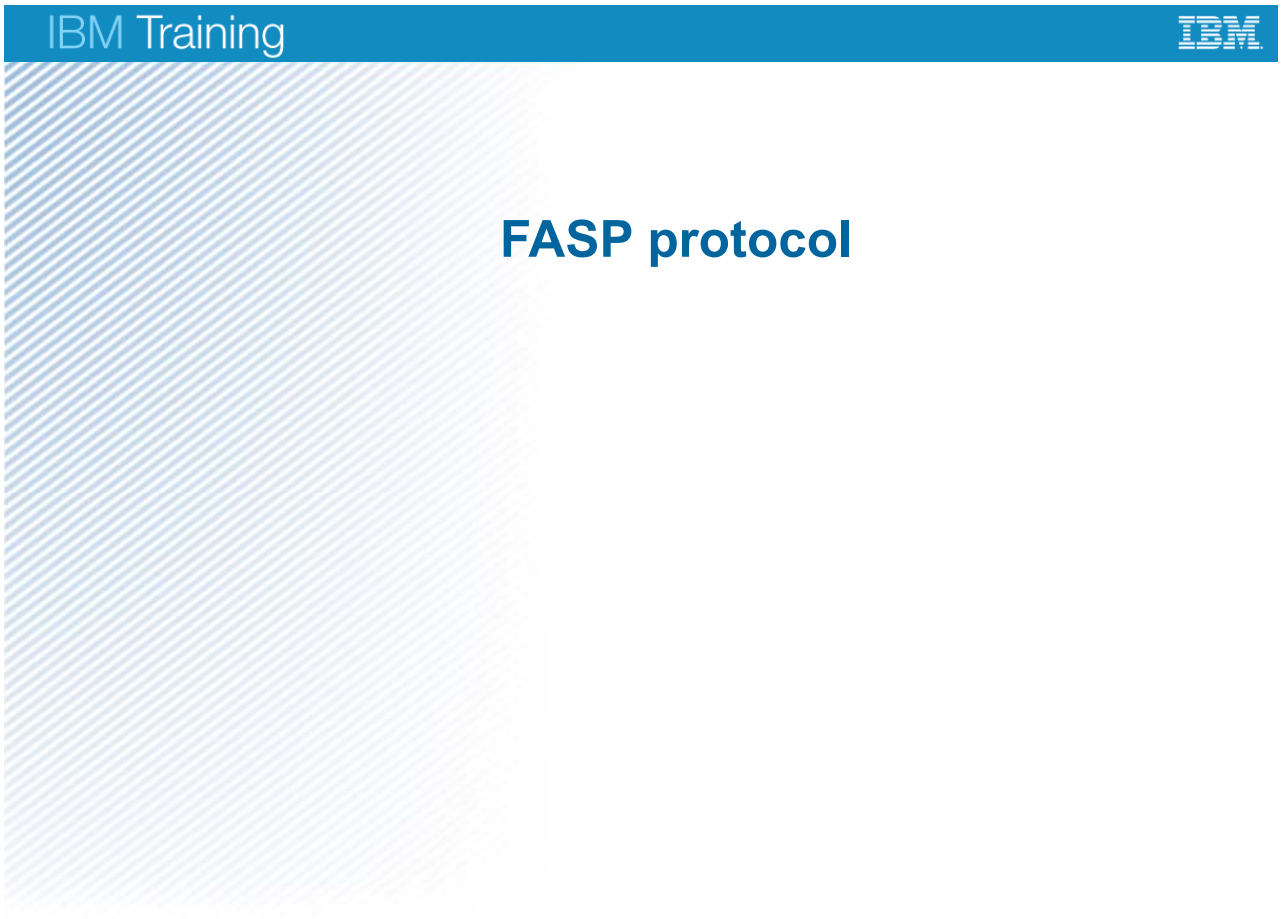
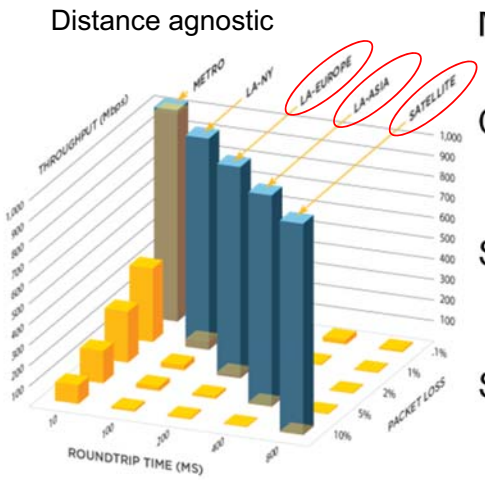


Figure 1-11. FASP protocol

IBM Training

FASP: Overview



Maximum transfer speeds
- Independent of distance and packet loss

Congestion avoidance and policy control
- Full utilization of available bandwidth
- Dynamic prioritization and bandwidth allocation

Secure and reliable
- User and endpoint authentication
- AES-128/256 cryptography in transit and at-rest

Scalable management
- Real-time progress, performance, and bandwidth utilization
- Detailed transfer history, logging, and manifest

Low overhead
- Less than 0.1% overhead at 30% packet loss
- High performance with both large and small files

Figure 1-12. FASP: Overview

All Aspera products are based on the FASP protocol, which facilitates the high speed transfers seen with Aspera solutions. The FASP protocol uses the User Datagram Protocol (UDP) for transport and avoids the problems that are associated with data transfers over the standard Transmission Control Protocol (TCP). However, while UDP provides packet delivery, the FASP protocol implements all congestion and flow control, avoiding the problems normally associated with UDP.

The FASP protocol implements many features that optimize data transfers. These features include:

- The separation of flow and congestion control functions
- Using the Return Transfer Time (RTT) for real-time adaptive rate control
- Depending upon negative acknowledgments to minimize the number of resent packets

Your instructor might ask some questions to help reviewing the critical concepts of FASP.

IBM Training

Transfer rate comparison

Fasp-based transfer speeds – location agnostic

		Across US		US to Europe		US to Asia	
		10 GB	100 GB	10 GB	100 GB	10 GB	100 GB
FTP	10 Mbps	10 to 20 Hours	Impractical	15 to 20 Hours	Impractical	Impractical	Impractical
	100 Mbps						
	1 Gbps						
	10 Gbps						
		10 GB	100 GB	10 GB	100 GB	10 GB	100 GB
Aspera FASP™	10 Mbps	140 Min	23.3 Hrs	140 Min	23.3 Hrs	140 Min	23.3 Hrs
	100 Mbps	14 Min	2.3 Hrs	14 Min	2.3 Hrs	14 Min	2.3 Hrs
	1 Gbps	1.4 Min	14 Min	1.4 Min	14 Min	1.4 Min	14 Min
	10 Gbps	8.4 Sec	1.4 Min	8.4 Sec	1.4 Min	8.4 Sec	1.4 Min

https://www.ibm.com/aspera/file-transfer-calculator/

The IBM Aspera FASP protocol

Figure 1-13. Transfer rate comparison

As the comparison shows, implementing Aspera FASP protocol results in significant improvements in data transfer rates over the rates realized by TCP-based file transfer programs.

NOTE: The URI shown on the slide links to the Aspera performance calculator that can provide a comparison of anticipated transfer times for TCP and FASP.

So how does FASP achieve such dramatic improvements over more traditional transfer programs?

The following pages present some details about the operations of FASP to realize such high transfer rates.

FASP: Transfer initiation

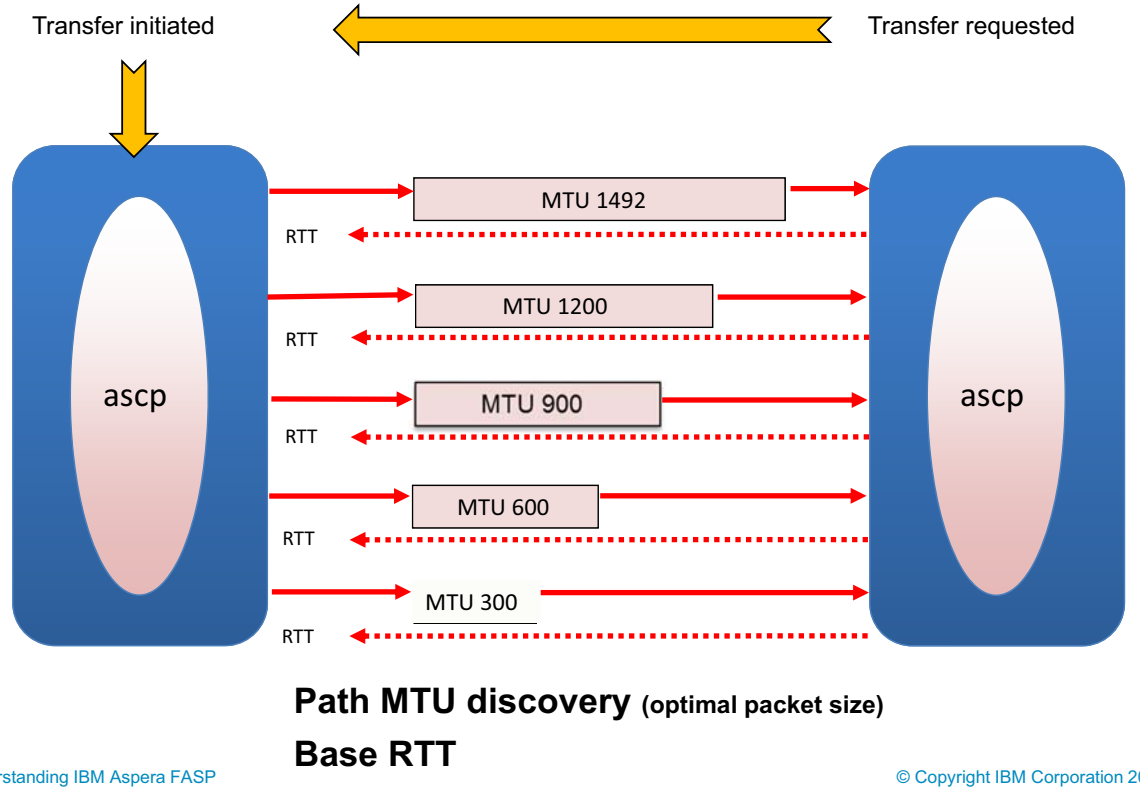


Figure 1-14. FASP: Transfer initiation

The FASP protocol is implemented within the ascp program, which is how all Aspera transfers are handled (discussed later). When ascp first starts, it sends a series of small packets to the destination address with various Mean Transmission Unit (MTU) sizes to determine the optimal packet size to be used for the upcoming transfer. The Internet Protocol defines the Path MTU of an internet transmission path as the smallest Mean Transfer Unit (MTU) of any of the IP hops of the path between a source and destination. Put another way, the path MTU is the largest packet size that can traverse this path without suffering fragmentation. In an IP network, the path from the source address to the destination address often gets modified dynamically. This behavior is in response to various events, for example, load-balancing, congestion, and outages. Changing the deliver path can result in different path MTU sizes (sometimes repeatedly) during a transmission. Resizing packet requires extra processing by routers, which can introduce further packet drops before the host finds a new reliable MTU.

The RTT for each packet is calculated to determine the optimal MTU size based on the amount of time it took from packet transmission to packet receipt. The RTT for each MTU is compared to determine which packet size yields the best network performance based on the lowest RTT.

The best RTT is then set as the Base Return Time (RTT) value. The Base RTT is continuously compared with the current RTT as the standard for assessing network congestion (details presented later in this module).

IBM Training

Optimal performance: Packet loss

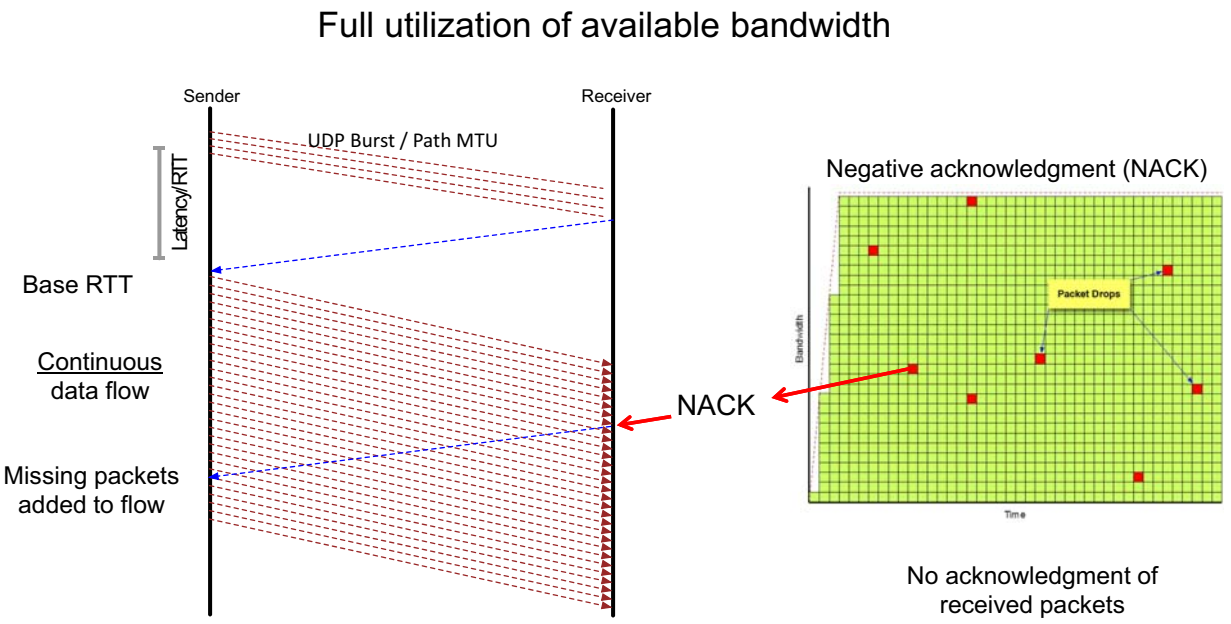


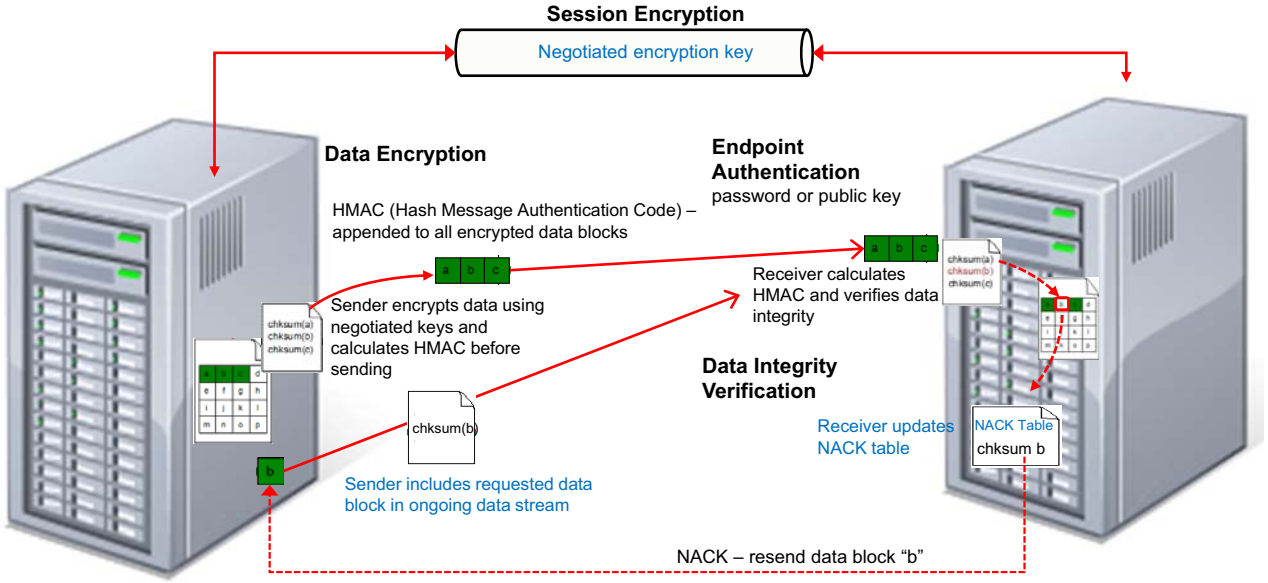
Figure 1-15. Optimal performance: Packet loss

The FASP protocol does not depend upon ACKs from the sender to determine what data was successfully delivered, so consequently does not waste bandwidth for data that was previously received. Instead, FASP uses Negative Acknowledgments (NACKs) to identify specific blocks of data that were not received in an acceptable state. Whereas TCP uses the sliding window and ACKs to determine which data needs to be retransmitted, FASP sends NACKs only for the specific blocks of data that is needed. The use of NACKs reduces the number of unneeded retransmissions, thus reducing network traffic.

A related feature of FASP is that the determination of transfer rate specified by the receiver, not the sender. So, the receiver sets the Target Rate for the sender, and the sender sends a steady stream of data to the receiver at the specified rate. If the sender receives a NACK, it adds the requested data to its ongoing stream, thus minimizing performance spikes like those of TCP-based transfers.

IBM Training

Reliable and secure transfers



- Encrypted session using SSH2 shell to establish secure channel to exchange encryption keys
- End-points authenticate using password or public key
- Datagrams encrypted using per session key and random 128-bit key for HMAC
- Receiver recalculates and verifies HMAC and decrypts data
- NACK table updated with bad data blocks and sent to sender

Understanding IBM Aspera FASP © Copyright IBM Corporation 2020

Figure 1-16. Reliable and secure transfers

Aspera transfers have complete, built-in security for data transfers that use the standard open source OpenSSL toolkit. The OpenSSL cryptographic libraries and the standard Secure Shell (SSH2) are used unmodified to take full advantage of the standard.

The Aspera security model consists of the following features:

- Session encryption to establish a secure channel for exchanging a random per-session key for encryption.
- Secure authentication of transfer endpoints.
- Flexible data encryption.
- Verification of each transmitted data block.

Session Encryption

Each transfer job begins by establishing a secure encrypted session between the endpoints that use the SSHv2 Secure Shell. The session encryption key is negotiated between the client and server by using a host-specific RSA key. A new key is generated each time the SSH daemon starts up, is regenerated each hour, and is never stored on disk.

Endpoint Authentication

After the secure session channel is established, the transfer endpoints authenticate by using one of the secure authentication mechanisms in SSH interactive password or public-key.

Data Encryption

Following SSH authentication, the FASP transfer session performs a three-way handshake. The remote endpoint generates a random AES-128/192/256-bit per-session key for data encryption, a random 128-bit key for computing an SHA2 checksum, and sends these keys to the initiator over the secure ssh channel. A new encryption and HMAC-SHA2 key is generated on each FASP transfer session, and the keys are never stored on disk.

Data Integrity Verification

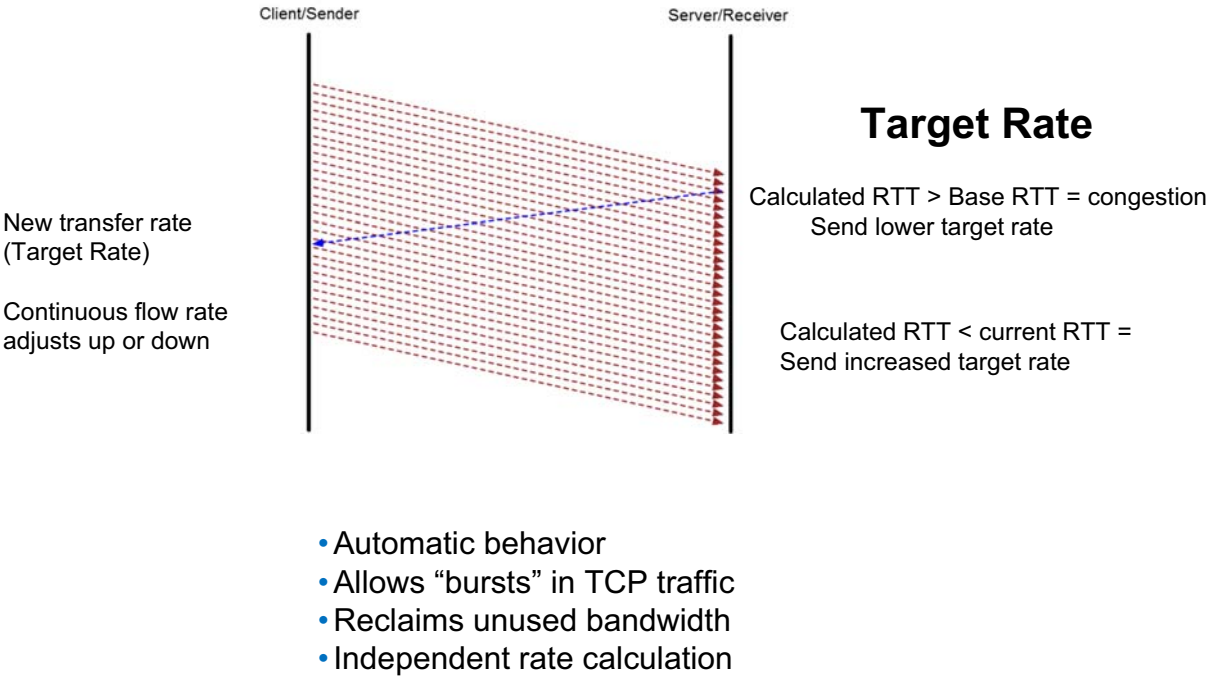
An SHA2 cryptographic hash function is applied to each encrypted datagram before transmission on the network. The resulting message digest or checksum is appended to the secure datagram and verified at the receiver for data integrity (to prevent man-in-the-middle, replay, and UDP denial-of-service attacks).

Additionally, the checksum value is also used for detecting errors, introduced during data transfers. Even the slightest differences between data blocks that are sent and data blocks that are received result in different checksum values.

If the checksum values match, the data is good and no additional processing is required. If the checksum values do not match, the receiving system updates the NACK table and notifies the sender to resend the data block. The sending system adds the requested data block to the stream of data it is already sending.

IBM Training

Adaptive rate control



- Automatic behavior
- Allows “bursts” in TCP traffic
- Reclaims unused bandwidth
- Independent rate calculation

Figure 1-17. Adaptive rate control

The IBM Aspera FASP protocol implements an Adaptive Rate Control function that provides the means for constant monitoring of the network conditions and adjusting the Target Rate value to reflect the observed network performance RTT. This Adaptive Rate Control function is a part of the FASP protocol, and does not require any specific configuration for proper operation.

The Base RTT is determined when a transfer is first initiated, but the receiver constantly evaluates the RTT, calculates a new Target Rate, and sends the new Target Rate to the sending system. By adjusting the Target Rate value down when the Observed RTT is greater than the Base RTT, Aspera transfers accommodate increased traffic on the network. When the Observed RTT is determined to be greater than the Base RTT value (reduced network traffic), the Target Rate is increased, yielding optimal performance.

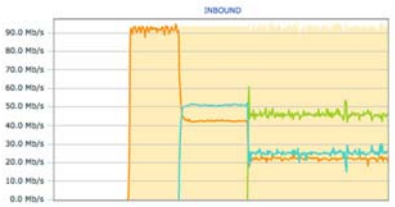
It is important to realize that Aspera transfer rates are determined for each transfer, independent of all other transfers in process. In other words, the transfer processes do not communicate with each other, but each makes its own calculation to determine a transfer rate based on the conditions and values of that particular transfer. This concept means that no central controlling function exists, and consequently, no single location for determining the actual transfer rates. Additionally, as each transfer is determined independently, the transfer rate can be increased or decreased dynamically throughout the duration of the transfer.

IBM Training

Bandwidth control

Default Behavior

- Adaptive rate control
  Concurrent transfers adjust bandwidth consumption



Configuration Parameters

- Sender and receiver parameters
- Bandwidth settings
- Bandwidth “policy”
- Vlinks

User Control

- Real-time job prioritization
  Per flow, per user, and per job

Understanding IBM Aspera FASP

Figure 1-18. Bandwidth control

Several bandwidth controls are implemented within the Aspera environment.

The Aspera FASP protocol has a built-in Adaptive Rate Control function that dynamically adjusts individual transfer rates, based on network congestion (details are provided later in this module).

Numerous parameters can be configured on both the sending and receiving systems that manage and control bandwidth during transfers, including bandwidth settings, transfer policies, and the use of Vlinks. A discussion of these parameters is presented later in this module.

Users can also manage their preferences for transfers when they initiate a transfer, including job priority, and the ability to stop, pause, resume, and cancel current transfers.

1.3. Transfer rate factors

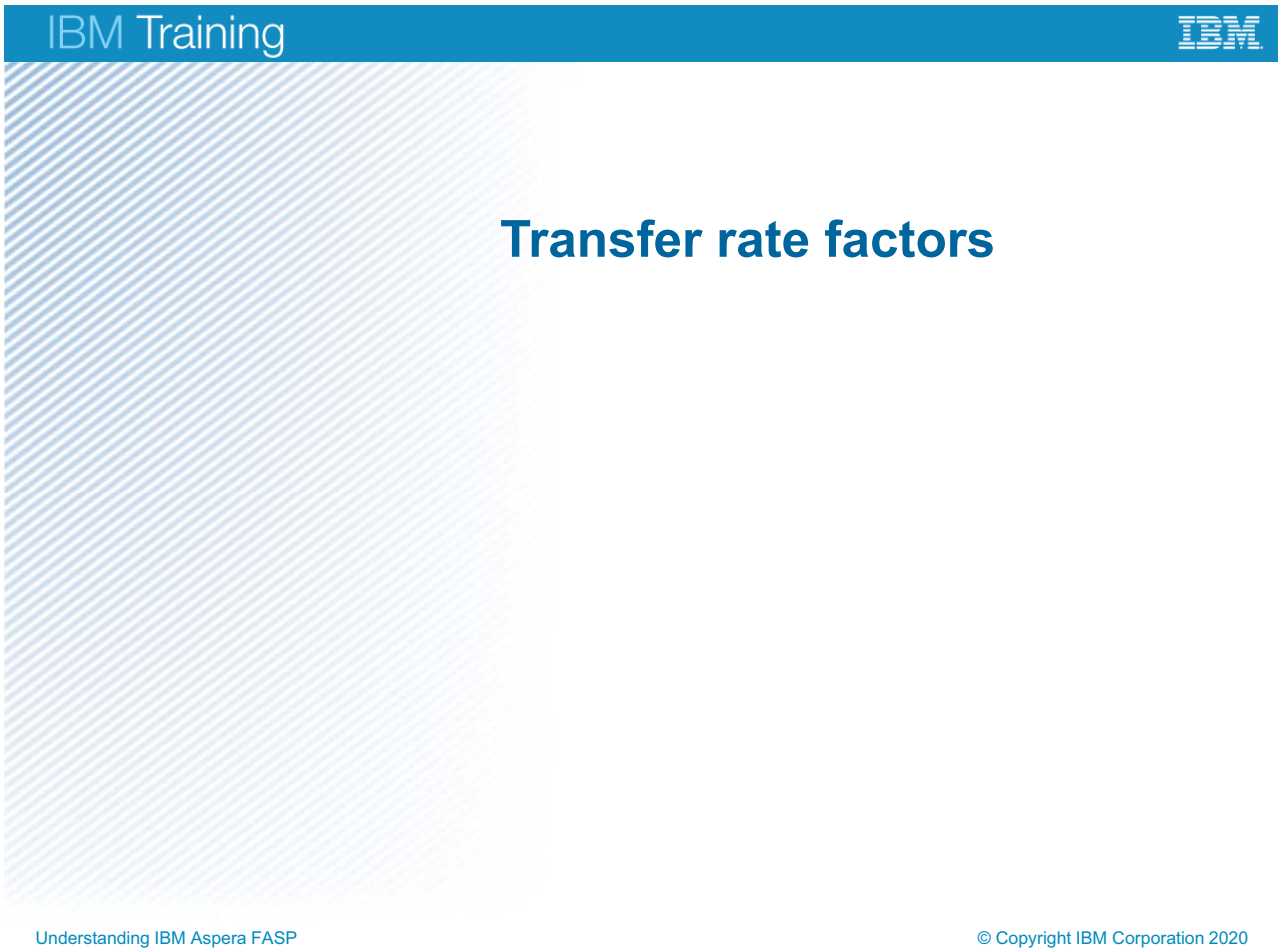


Figure 1-19. Transfer rate factors

IBM Training

Transfer rate factors

Absolute Limits

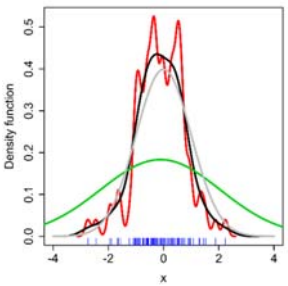
- Max network capacity (bandwidth)
- Max licensed bandwidth (Max global bandwidth)

Configured Limits

- Target Rate
- Target Rate Cap
- Vlink Rate

Storage I/O

- Sender
- Receiver



Available Bandwidth

- Bandwidth used by other traffic
- Transfer Policy

Understanding IBM Aspera FASP

Figure 1-20. Transfer rate factors

The actual bandwidth that is realized for any transfer is variable, dependent upon multiple settings and factors within the transfer path. Regardless of what values set in the Transfer Server configuration (or the client system's configuration), a transfer might not be as fast as you would like

A couple of configuration values cannot be exceeded, regardless of what values you configure for your system:

- The maximum capacity of the network the transfer is using.
- The maximum bandwidth rate that is embedded in the license key you purchased.

You can define the Default Target Rate value on both the Transfer Server and the remote client system at the same level as either of these limits, but you cannot exceed them. It is a relatively simple task to increase the Maximum Global Bandwidth (license key rate) value by purchasing a higher-capacity license and adding it to your system.

Target Rate: The Transfer Server's Default Target Rate and Target Rate Cap values can be configured for both incoming and outgoing transfers. While a transfer initiated from a client system can request a rate to use for their transfer, when the Transfer Server compares the client's requested rate against its own Target Rate Cap value. If the requested rate exceeds the server's Target Rate Cap, the request is denied. However, if the client's request is at or below the server's Target Rate Cap value, the server attempts meet the requested rate. So, the Target Rate values defined on the Transfer Server system takes precedence over a requested transfer rate. Both the

server and client's Target Rate values must be at or below the license key's maximum bandwidth value.

In addition to the client and server Target Rate values, if a Vlink is assigned to the user, the assigned Vlink capacity on the server also limits the transfer rate between the client and server. If more connections associated with the same Vlink ID are established, the resulting transfer rate for each Vlink-related transfer is restricted to maintain consistency with the maximum rate supported by the Vlink itself.

The performance of the I/O system on the sending and receiving systems can become a limitation to transfer performance. While the network capacity, the license key rate, and the server's Target Rate value might be high, reduced I/O capabilities on either the sending or receiving system impacts the transfer rate.

Another major factor in determining the actual transfer rate of any FASP transfer, is the number of bandwidth other applications on the same network use. Obviously, a busy network can result in lower transfer rates. Depending upon the Transfer Policy value setting (Fixed, High, Fair, or Low), the FASP transfer rate might be below the allowed maximum values. A Low or Fair transfer policy setting on the Transfer Server or the client system can yield slower transfer rates, regardless of the defined Target Rate or Vlink values. If the transfer policy has a value of Fixed or High, transfer rates might be optimized, but at the expense of other applications that use the network at the same time as the FASP transfer.

Ultimately, when considering the entire path between the FASP-based sending and receiving systems, the actual transfer rate is limited to the lowest value defined for any of the bandwidth parameters discussed.

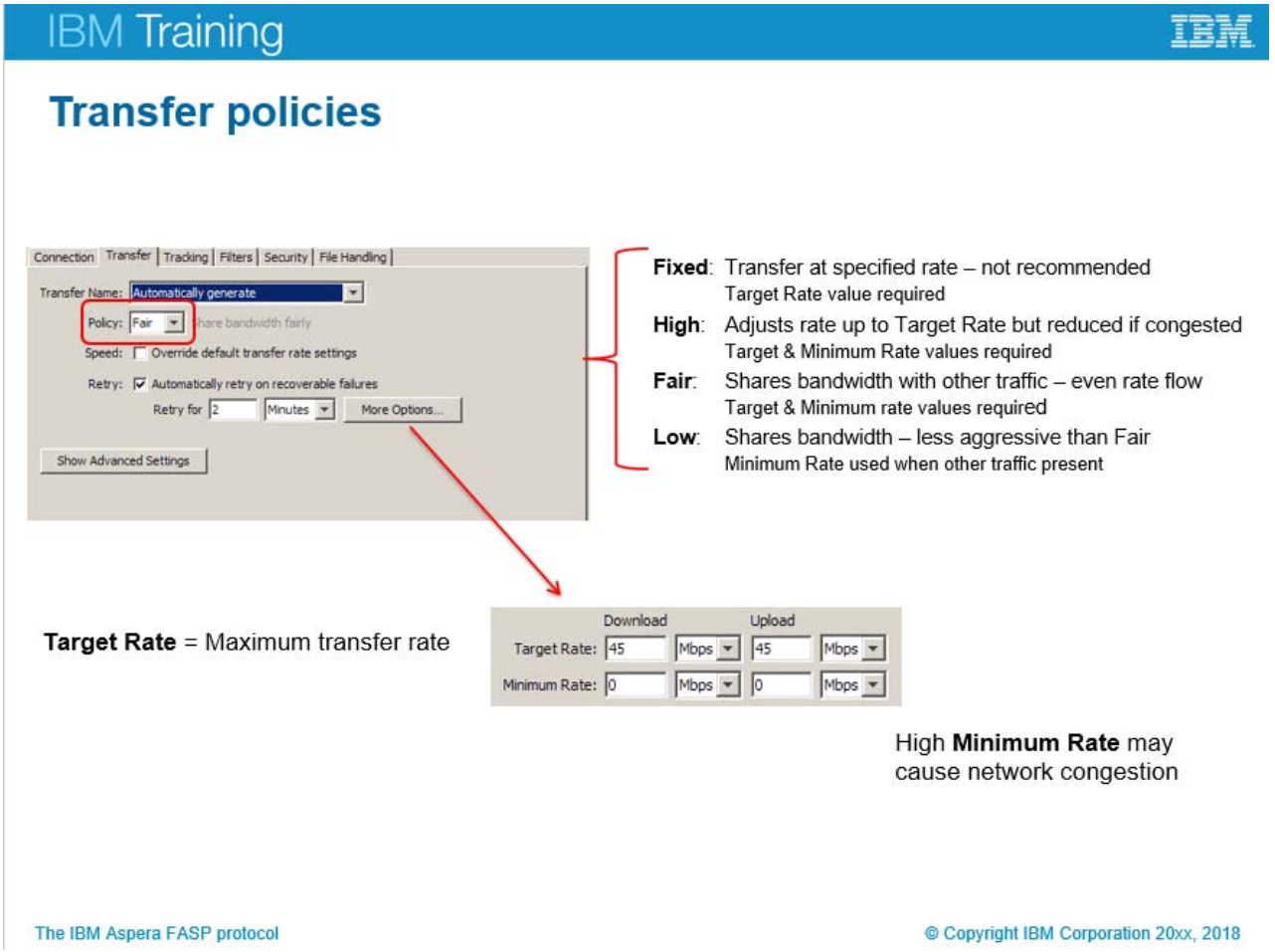


Figure 1-21. Transfer policies

Parameters that define transfer behaviors between Aspera client software and a remote Aspera server are defined in the aspera.conf file, found on each IBM Aspera High-Speed Transfer Server. The following parameters can be set:

Transfer Name

This field indicates how the system should name files that are transferred to or from this server. Files can be automatically named, or can include a specified prefix value.

Policy: The transfer policy and speed determine how the network resources are used for FASP file transfers:

Fixed: FASP 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 specified time. This policy occupies most of the network's bandwidth, and is not recommended in most file transfer scenarios. In this mode, a maximum (target) rate value is required.

High: FASP monitors the network and adjusts the transfer rate to fully use the available bandwidth up to the maximum rate. When congestion occurs, a *FASP* session with high policy 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.

Fair: FASP monitors the network and adjusts the transfer rate to fully use the available bandwidth up to the maximum rate. When other types of traffic buildup and congestion occur, FASP shares bandwidth with other traffic fairly by transferring at an even rate. In this mode, both the maximum (target) and the minimum transfer rates are required.

Low: Similar to Fair mode, the Low (or Trickle) 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 decreased all the way down to the minimum rate until other traffic is reduced.

Speed: This checkbox provides a means of setting specific Target and Minimum Rate values for downloads from and uploads to this remote server, rather than using the default Target and Minimum Rates defined for your system.

The optional rate fields are displayed when the Override default transfer rate settings checkbox is selected. The Target Rate value indicates the maximum allowed transfer rate when transferring files to or from the server. The target rate value also defines the initial rate that is attempted for transfers. The target rate can be dynamically reduced throughout the life of a transfer, depending upon the Transfer Policy value, is not exceeded at any time during a transfer. The Target Rate value should not exceed the maximum physical capacity of the network, nor should it exceed the maximum rate that is designated by the Aspera license key. When determining the Target Rate value, make sure to consider the receiving system’s capacity to write data to it’s storage device.

The Minimum Rate value references the absolute minimum rate that is allowed for transfers. The default Minimum Rate value is set to 0 Mbps, but can be set higher if wanted. However, setting the Minimum Rate too high can cause network congestion for other traffic. Aspera suggests leaving the Minimum Rate value set to 0 unless a specific reason to do otherwise exists.

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 can set the initial and maximum retry intervals by clicking the More Options button.

Initial interval - The first retry waits for the initial interval. You can use the menu to specify the value 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. You can also specify this time in seconds, minutes, or hours.

NOTE: Unrecoverable transfer failures include significant events such as server or network failure, not failures due to network congestion.

IBM Training

Vlinks

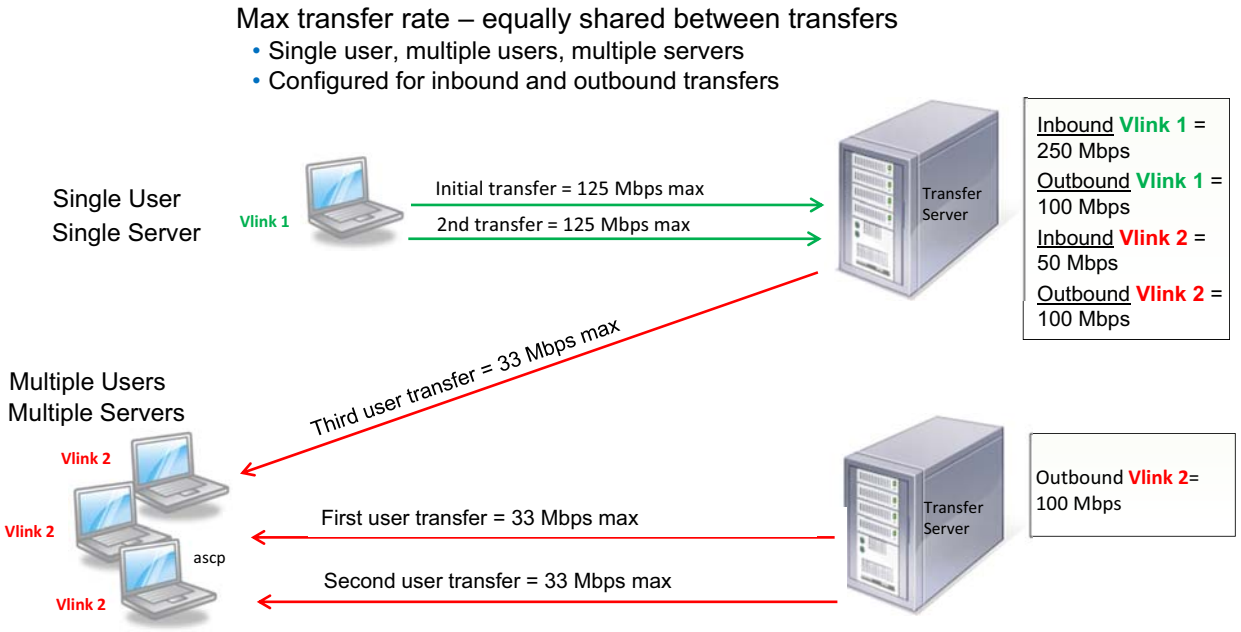


Figure 1-22. Vlinks

Vlinks provide a means for limiting the amount of aggregate bandwidth available to one or more users. The bandwidth assigned to a Vlink is an aggregate maximum, regardless of how many users can be assigned to the Vlink. Vlinks can be associated with both inbound and outbound transfers (relative to the transfer server where the Vlink is defined), and can apply across transfer servers. Vlinks offer a powerful tool for controlling network bandwidth usage and ensuring adequate bandwidth for other network traffic.

The examples shown highlight how a single user is restricted to the Vlink-defined bandwidth for inbound transfers, regardless of how many simultaneous transfers can be active. The second example that is shown indicates how multiple users are restricted to the aggregate Vlink value (100 Mbps), as well how Vlinks can control transfers that involve multiple transfer servers.

Vlinks are policies that allow aggregate bandwidth control. Vlinks can be thought of as virtual pipes that act like an overall limiter to the bandwidth that is shared among transfers that are assigned the same Vlink.

These shared transfers can be between a user and the transfer server, but can also be applied to transfers between servers.

The use of Vlinks provides a reliable means of managing Aspera transfers, regardless of how many simultaneous transfers occur.

Vlinks can be configured for both inbound and outbound transfers, and across multiple transfer servers.

Single User Example

A maximum bandwidth value is assigned to each Vlink, and the Vlinks can be assigned to a single user. For example, a user can be assigned a Vlink with a maximum bandwidth of 100 Mbps. When that user starts a single transfer, they are not allowed to exceed a rate of 100 Mbps. This initial behavior is not different than setting a maximum rate for that user. However, because a Vlink is involved, when a second transfer is started while the first is still active), the maximum rate for first transfer is reduced to 50 Mbps. And, the maximum rate for second transfer is also 50 Mbps. If a third transfer is started while the others are still active, the 100 Mbps Vlink maximum would be divided between the three current transfers.

Multiple-User Example

The same Vlink can be assigned to multiple users, allowing you to control the maximum bandwidth available for transfers by the aggregate of all users who are assigned with the same Vlink. If the Vlink rate is 100 Mbps, and that same Vlink is assigned to three transfer users, then the first user who starts a transfer would have a maximum rate of 100 Mbps. If a second user who is assigned to the same Vlink starts a transfer, then both users would be limited to 50 Mbps. If a third user who is assigned with the same Vlink starts another transfer, the maximum bandwidth for all three users is limited to approximately 33 Mbps. If any of the three users started a second transfer while the others were still active, the maximum rate for all transfers would be reduced to equal parts for each session.

Multiple-Server Example

The same Vlink ID and capacity can be configured on multiple servers. When a user starts a second transfer with a different server, both transfers are adjusted to be consistent with the Vlink capacity.

The use of Vlinks allows an administrator to manage bandwidth for the entire community of transfer users. Setting limits with Vlinks, allows more capacity for users that require it, while still managing the overall bandwidth for all transfers.

Vlinks can be assigned as a global value, or assigned to specific users, or to user groups (any user who is a member of the group). Additionally, maximum rates can be assigned to both incoming and outgoing Vlink values on both the client and server systems, providing maximum flexibility.

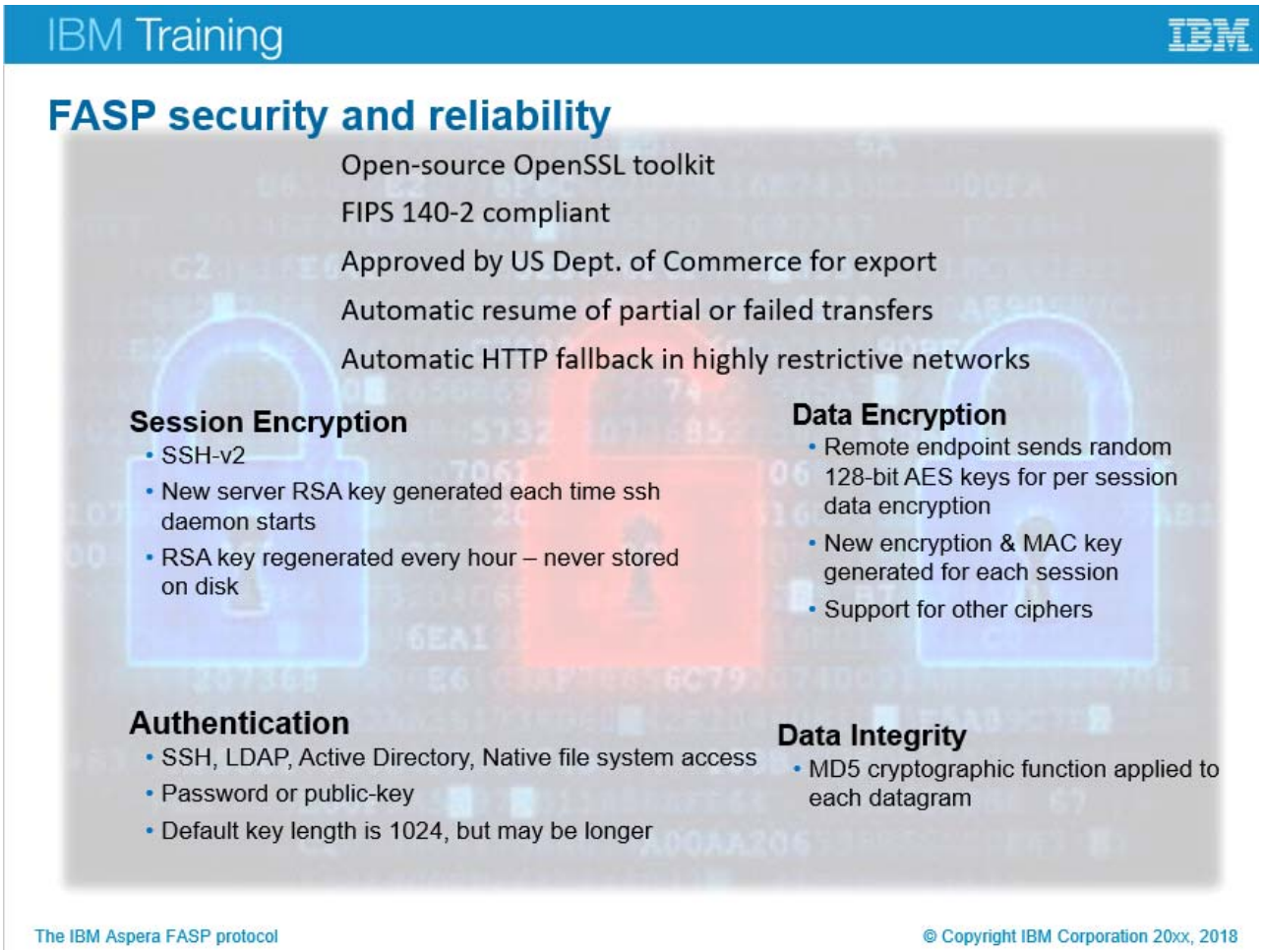


Figure 1-23. FASP security and reliability

The FASP protocol provides a comprehensive built-in security model that does not compromise transfer speed. Based on open standards cryptography, FASP provides SSH end-point authentication, ad hoc data encryption, and data integrity verification that protects against man-in-the-middle, replay, and UDP denial-of-service attacks. Aspera products use the standard open source OpenSSL toolkit. The OpenSSL cryptographic libraries and the standard Secure Shell (SSH) are used unmodified to take full advantage of the standard.

Session Encryption

Before a transfer is begun, a secure encrypted session is established between the sender and receiver. By default, SSHv2 is used for the encrypted session that uses a Diffie-Hellman key agreement to negotiate the session encryption key. Host-specific RSA keys are generated on each host and regenerated every hour, which is provided to the client and is integrated into the negotiated session encryption key.

Authentication

After session encryption is finalized, the client endpoint is authenticated by using SSH, either with a password or with a public key. Private keys are stored in encrypted format.

Data Encryption

Following successful authentication, data is encrypted using a random 128-bit per-session key for computing an MD5 checksum, which is used to encrypt each data block transferred. New encryption and MAC keys are generated for each FASP transfer session, and the keys are never stored on disk.

Data Integrity

As described previously, an MD5 cryptographic hash function is applied to each encrypted datagram sent, and verified for integrity at the receiving endpoint.

IBM Training

What you learned

- FASP begins each session by sending multiple MTU discovery packets to determine the Base RTT
- FASP uses NACK rather than ACK (used by TCP) which reduces the amount of unneeded retransmits of already received data
- FASP implements an adaptive rate control, whereby RTT is continuously monitored, resulting in new Target Rate values to reflect optimal transfer rate
- Adaptive Rate Control allows FASP to automatically reduce transfer rates when needed, but also increase rates up to maximum performance
- Transfer policy settings (Fixed, High, Fair, and Low) that are used for each transfer influence rate control
- Vlinks are policy settings that allow aggregate bandwidth control
- Vlinks can be implemented between a client and a single server, or can span multiple servers
- FASP has a built-in security model that is based on open cryptography standards
- FASP implements SSHv2-based session encryption, SSH authentication (password or public key), data encryption, & MD5-based data integrity

Figure 1-24. What you learned

IBM Training

Unit summary

- Explain the fundamental performance problem of TCP-based transfers
- Outline the process FASP uses to determine optimal packet size
- Highlight the difference between how FASP and TCP manage packet loss
- Describe the adaptive rate control process used by FASP
- Identify the factors that influence transfer rates
- Clarify the value of using Vlinks

Figure 1-25. Unit summary

IBM Training

Review questions (1 of 2)

1. **True or False:**
   The sending Aspera system tracks the packets that it sends and uses NACK to inform the receiver that it resending a corrupted packet.

2. **True or False:**
   HTTP Fallback can be configured for an individual transfer user.

Figure 1-26. Review questions (1 of 2)

1. False

It is the receiving system that manages the transfer process, not the sender. The receiving system identifies corrupted or missing packets and enters the packet information into the NACK table, which is periodically sent to the sending system to include the problem packet in the transfer stream.

2. True

IBM Training

Review questions (2 of 2)

3. Which of the following features should not be used in normal Aspera server configurations? Select all that apply.
   A. Fixed bandwidth transfer policy
   B. Vlinks
   C. Encryption at rest
   D. Minimum rate setting of zero

4. Which of the following represent performance limitations of TCP-based transfer routines? Select all that apply:
   A. Dependence upon the UDP protocol
   B. The “sliding window”
   C. The use of positive acknowledgments
   D. Using RTT to resize the sliding window

Figure 1-27. Review questions (2 of 2)

3. A and D
4. B and C

IBM Training

Review answers (1 of 2)

1. True or False: The sending Aspera system tracks the packets that it sends and uses NACK to inform the receiver that it resending a corrupted packet.
   The answer is False. It is the receiving system that manages the transfer process, not the sender. The receiving system identifies corrupted or missing packets and enters the packet information into the NACK table, which is periodically sent to the sending system to include the problem packet in the transfer stream.

2. True or False: The FASP protocol's adaptive rate control is dynamically managed by the receiving system, which uses RTT to determine when to notify the sending system to use a new target rate for the transfer.
   The answer is True.

Figure 1-28. Review answers (1 of 2)

IBM Training

Review answers (2 of 2)

3. Which of the following features should not be used in normal Aspera server configurations? Select all that apply.
   A. Fixed bandwidth transfer policy
   B. Vlinks
   B. Encryption at rest
   C. Minimum rate setting of zero

   The answer is A and D.

4. Which of the following represent performance limitations of TCP-based transfer routines? Select all that apply:
   A. Dependence upon the UDP protocol
   B. The “sliding window” feature
   C. The use of positive acknowledgments
   D. Using RTT to resize the sliding window

   The answer is B and C.

Figure 1-29. Review answers (2 of 2)

Unit 2. Overview of IBM Aspera Software

Estimated time

01:30

Overview

This unit a brief overview of IBM Aspera software and how they might be integrated as an Aspera environment

IBM Training

Unit objectives

- Identify the function of common IBM Aspera software
- Outline how IBM Aspera software products interact with each other
- Describe the ascp process and how it communicates with Aspera products

Figure 2-1. Unit objectives

IBM Training

Topics

- IBM Aspera products
- The **ascp** utility
- Aspera and directory services

Figure 2-2. Topics

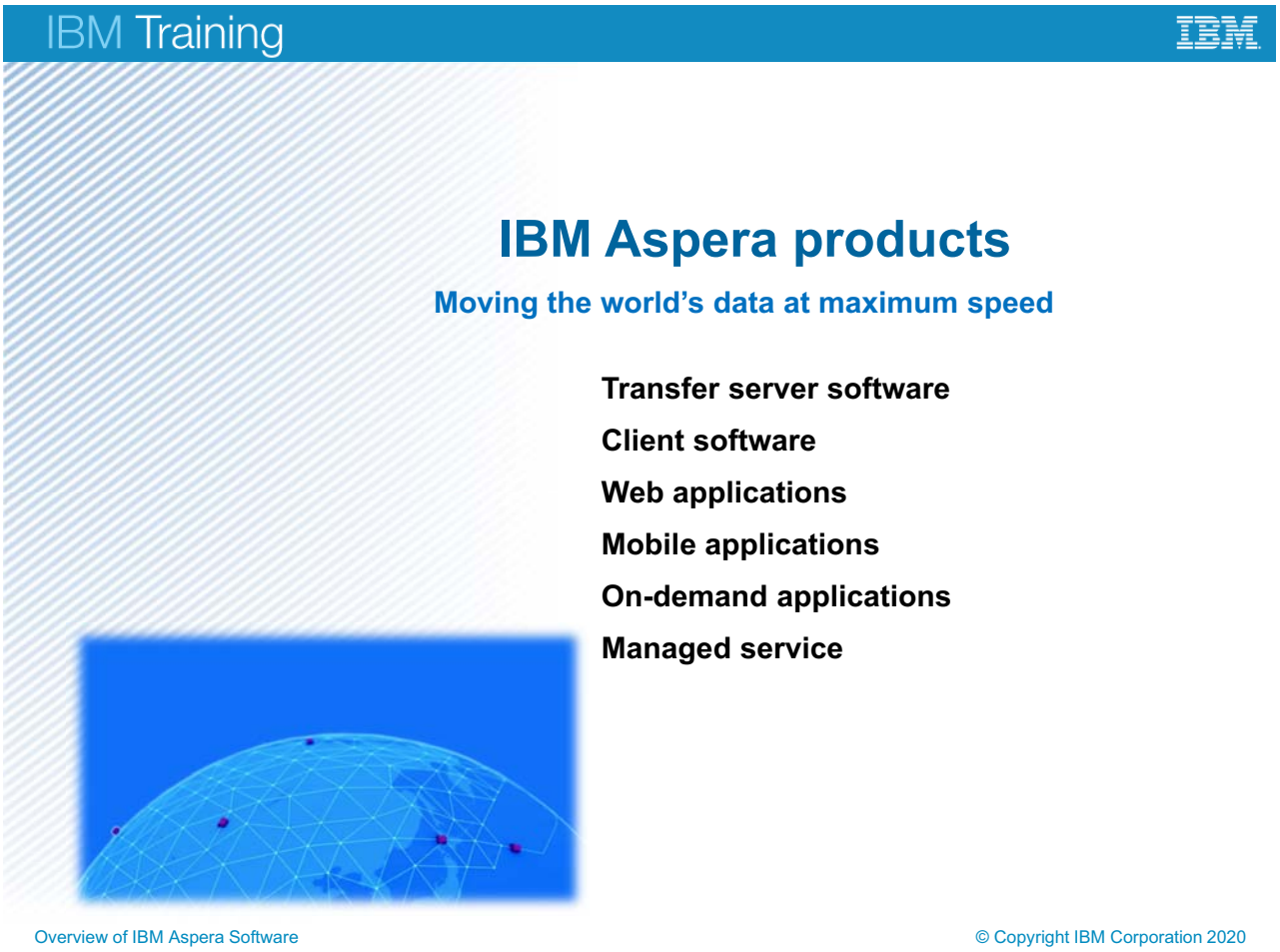


Figure 2-3. IBM Aspera products

IBM offers various products that are designed to meet the needs of organizations who need high-speed file transfers. These products can be grouped into the categories shown.

Transfer Servers

A transfer server is the foundation of the Aspera environment. Transfer server function is provided by IBM Aspera High-Speed Transfer Server and IBM Aspera High-Speed Endpoint software.

NOTE: IBM Aspera High-Speed Endpoint is marketed as a client application due to its support for a limited number of simultaneous users.

Client software

The transfer server is the engine behind all FASP-based transfers, but an application that supports FASP must be installed on client systems that initiate transfers.

Web applications

IBM offers several web-based applications that use the FASP protocol for file transfers, but minimize the knowledge that is required to perform transfers.

Mobile applications

IBM offers several mobile applications that enable mobile devices to interact with an Aspera environment. IBM Aspera Uploader Mobile, IBM Aspera Faspex Mobile, IBM Aspera Drive Mobile,

and IBM Aspera on Cloud Mobile are all available for both iOS and Android devices. Details about these products are not included in this course. You can download these applications from the /www.ibm.com/aspera/downloads website.

On-demand applications

Customers who want Aspera services in a cloud environment, but want to maintain the systems themselves can subscribe to IBM Aspera services installed in a cloud environment. IBM Aspera Server On Demand provides the IBM Aspera High-Speed Transfer Server software that is installed on one or more virtual servers in the customer's selected cloud environment. IBM Aspera Faspex on-Demand Server and IBM Aspera Shares On Demand provide the same functions of the on-premises versions of the products, but run on images that are provided by IBM in a cloud environment.

Managed service

IBM Aspera on Cloud is a hosted service that allows users to move files across on-premises and multicloud environments with an easy-to-use interface that simplifies file uploads, downloads, sharing, and distribution.

IBM Training

Transfer servers

IBM Aspera High-Speed Transfer Server (IBM Aspera HSTS)
- Previously sold as Enterprise Server or Connect Server
- Network speed defined by license
- Unlimited users
- IBM Aspera Sync software embedded (requires separate license)

IBM Aspera High-Speed Transfer Endpoint (point-to-point)
- Entry-level transfer solution
- Sold as client software (limited server function)
- Limited to two users accounts – 1 for transfers and 1 for administration
- Same features and functions as IBM Aspera HSTS

Figure 2-4. Transfer servers

In the past, the Aspera server software was sold as two separate products, Aspera Enterprise Server and Aspera Connect Server. The source code of these products was the same, but their functions were different based on the license applied to the installation. The Aspera Connect Server license implemented extra web capabilities to allow HTTP connections to the server. The Aspera Enterprise Server did not support web-based connections, but required specific client software to connect with the server. This distinction is no longer implemented, and the IBM Aspera HSTS software is available, which does support web-based connections.

Aspera High-Speed Transfer Endpoint (previously Aspera Point-to-Point Client) is the entry level transfer software can directly connect to other endpoints and transfer servers. The software comes complete with an interface that allows users to easily navigate file systems, initiate drag-and-drop transfers, track transfer progress, and reorder jobs or adjust bandwidth where needed.

The function of this product is the same as the IBM Aspera HSTS systems, but it has a two user limit to the number of simultaneous user connections. One account is used for FASP-based transfers and the other for application administration. The single transfer user account can authenticate user requests, making the system useful when files need to be pushed to a location, but only a single transfer user account is needed.

IBM Training

IBM Aspera High-Speed Transfer Server

- Engine for FASP-based transfers
- Linux/Windows/Mac operating systems
- Unlimited number of concurrent users (except Endpoint)
- Transfer initiation from GUI, Connect, Desktop client, Endpoint client, Command Line Interface, Drive, Cargo, 3rd-party embedded client, or Mobile Uploader app
- Administrative and user GUI
- Parallel and multi-host transfer feature enables transfers to multiple computers in both cloud and on-premises
- Automated transfers
- Supports Node API
  - Pre/Post file processing
  - Enterprise-grade security
    - ✓ Thorough SSH authentication
    - ✓ Encryption in-transit and at-rest
    - ✓ Server-side encryption at rest with secret key
    - ✓ Encryption settings per user
    - ✓ FIPS 140-2 compliant cryptographic module
    - ✓ Allow or deny transfer requests by client IP address and by cookie patterns

Overview of IBM Aspera Software

Figure 2-5. IBM Aspera High-Speed Transfer Server

The foundation for all Aspera transfers is provided with an IBM Aspera High-Speed Transfer Server. Both IBM Aspera High-Speed Transfer Server and IBM Aspera High-Speed Endpoint can fulfill the role of the transfer engine, all of which are supported on multiple operating systems.

NOTE: IBM Aspera Endpoint is classified as a client system rather than a server, but it supports an administrative account and a single user (as opposed to unlimited concurrent users). The IBM Aspera High-Speed Endpoint application allows a FASP-based client system to initiate a transfer to the IBM Aspera Endpoint system. Other Aspera clients do NOT support authentication, thus can be used only to initiate a transfer.

An IBM Aspera HSTS supports transfer initiations from a number of different clients. These clients include:

- IBM Connect browser plug-in.
- IBM Aspera GUI on another IBM Aspera HSTS server.
- IBM Aspera Desktop Client
- IBM Aspera Command-line Interface
- IBM Aspera Endpoint
- IBM Aspera Mobile Uploader application.

The IBM Aspera HSTS system can also respond to HTTP requests made via Node API.

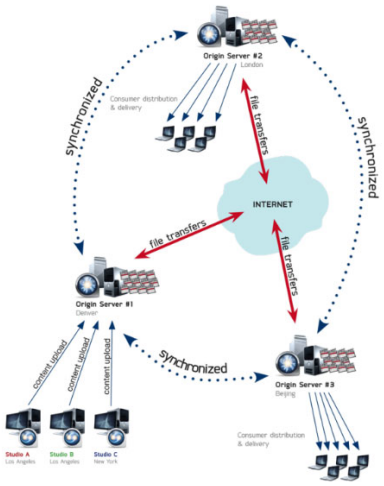
Transfer servers can also support parallel transfer feature, which enables a single transfer to multiple transfer servers, whether they are on-premises or in the cloud.

Transfer servers provide robust management of users and transfers, including automating transfers between transfer servers and pre or post file transfer processing of individual files.

All Aspera products, are designed to provide business-critical digital assets safe with enterprise-grade security features.

IBM Training

IBM Aspera Sync



- High-speed file-based replication and synchronization
- Uses FASP transport
- Compatible ***rsync*** command-line interface
- Locally detects changes and compares to file system snapshot without having to check with remote systems
- Replicates file moves and file renames on the source as a file move or rename on the target, avoiding unnecessary data transfers
- Supports bidirectional and multi-directional synchronization topologies
- Uses file system notifications for change notification, when available.
- Waits for the system to become stable (detects growing files) then performs synchronization
- Licensed as add on to IBM Aspera HSTS or IBM Aspera HST Endpoint

Figure 2-6. IBM Aspera Sync

IBM Aspera Sync is a software application that provides high-speed and highly scalable multi-directional, file-based replication, and synchronization. Aspera Sync is designed to fill the performance gap of uni-directional file synchronization tools like rsync, which are often slow for synchronizing large files and large sets of files over the WAN. Additionally, Aspera Sync extends the capability of uni-directional synchronization tools with full support for synchronization that is bidirectional and multi-directional.

IBM Aspera Sync offers the following key capabilities:

- Uses high-speed FASP transport for moving data at maximum speed over the WAN, whereas traditional synchronization tools are built on TCP. Aspera Sync transfers new data between remote hosts at full bandwidth capacity, regardless of round-trip delay and packet loss, and does not degrade in performance for large file sizes.
- Compares against a local snapshot, thus avoiding the process of making a comparison against the remote file system over the WAN, which is used by most traditional tools and can be slow.
- Recognizes file system changes (such as moves and renames) and propagates these changes to peers. Traditional tools treat these operations as deletion of old data and then re-create or retransfer the new data, which can lead to costly data copy over the WAN.
- Supports bidirectional and multi-directional synchronization topologies, where files are changing on multiple nodes. For a bidirectional synchronization, Aspera Sync runs with a

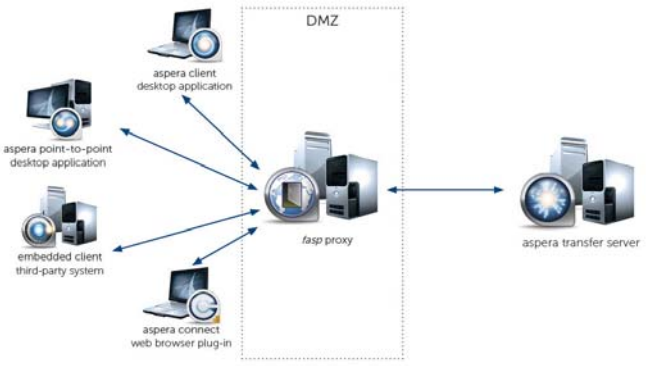
bidirectional option. For a multi-directional synchronization, one session is run for each peer to remain synced. Any topology that has an acyclic graph topology between peers is supported.

- Uses file system notifications for change notification, when available.
- Monitors file contents and waits for files to be stable (no longer changing in md5sum) before transferring. The wait period is configurable and is designed to avoid transferring only partially complete files.
- Licensed as an add-on to Aspera HSTS or Aspera HST Endpoint.
- One license per installed host, with tiers based on file system size (number of files).

IBM Aspera Sync is a command-line program called async that, like rsync, uses an SSH connection to establish connectivity with its remote peers and is created as an SSH subsystem binary on the remote system. The program can run one time or periodically on file systems that do not provide asynchronous change notification, or in a continuous mode on file systems that do support asynchronous change notification. IBM Aspera Sync is designed to process files and transfer new data in a continuous pipeline for maximum speed, even when running in scan-only mode (when no file system change notification is available).

IBM Training

IBM Aspera Proxy Server



- Secure FASP-based transfers to or from internal or external clients
- Enables external access to transfer servers without placing in DMZ
- Forward or reverse proxy services
- Uses DNAT to hide internal IP addresses

- Optional user authentication controls which users can initiate transfers
- Can be run on server cluster behind load balancer for HA solution
- Supports chained proxies for 2-tier DMZ configurations
- Forwarding rules define access to specific transfer servers
- Easy-to-use interface for configuration

Figure 2-7. IBM Aspera Proxy Server

IBM Aspera FASP Proxy Server software provides a secure method to allow designated external users access to internal Aspera transfer servers without requiring the transfer server be placed in the DMZ. **IBM Aspera FASP Proxy Server** can also allow internal users access to Aspera transfer servers in highly secure network configurations.

IBM Aspera FASP Proxy Server can be configured to provide both forward and reverse proxy services, depending upon specific requirements

IBM Aspera FASP Proxy Server uses Destination Network Address Translation (DNAT) to hide the IP address of the internal transfer server seen by external initiators.

IBM Aspera FASP Proxy Server offers granular authentication controls to allow or disallow transfers by specific users, and access to specific transfer servers.

IBM Training

Client software

IBM Aspera Connect

IBM Aspera Command Line Interface

IBM Aspera Desktop Client

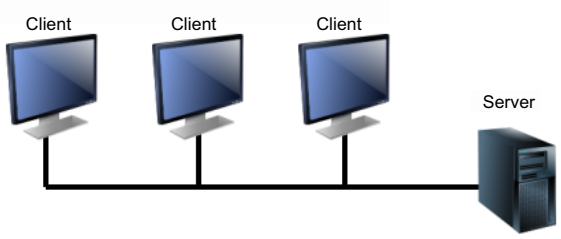


Figure 2-8. Client software

IBM Aspera Connect is an install-on-demand browser plug-in that is compatible with most standard web browsers. IBM Aspera Command Line Interface implements the ascp (Aspera secure copy) program on client systems. The ascp routine is a command-line FASP transfer program that is required for all FASP-based transfers. IBM Aspera Desktop Client is a desktop version of the Aspera GUI that is included with an Aspera Transfer Server environment.

Other client applications are available that further simplify FASP-based transfers. These applications are not covered in this training course, but can be viewed on the IBM Aspera downloads website: www.ibm.com/aspera/downloads.

IBM Training

Web applications

IBM Aspera Faspex

IBM Aspera Shares

IBM Aspera Console

IBM Aspera Orchestrator

Figure 2-9. Web applications

IBM Aspera Shares and **IBM Aspera Faspex** are web-based applications that use the FASP protocol for file transfers, but minimize the knowledge users need to conduct and manage transfers.

IBM Aspera Console is a web-based application that allows users to centrally manage, monitor, and control Aspera servers (nodes) and transfers.

IBM Aspera Orchestrator is another web-based application that automates the collection, processing, and distribution of large volumes of file-based digital assets. The Orchestrator engine supports conditional decision-making, manual user inputs, automated high-speed file movement, and third-party system integrations. Orchestrator uses FASP transport technology for all high-speed file transfers and is fully integrated with all Aspera transfer mechanisms.

IBM Training

IBM Aspera Faspex

- Person-to-person and project-based file-exchange
- Simple interface for exchanging files and directories
- Web-based interface, email, mobile and desktop client interfaces
- Customizable email notifications of Faspex events
- Create and manage workgroups for file-based collaboration
- Enterprise-scale user management and access control
- Seamless integration with organization's directory service users and groups
- Multi-server relay (transfers from sender are relayed to recipient's home server)
- HA configuration available

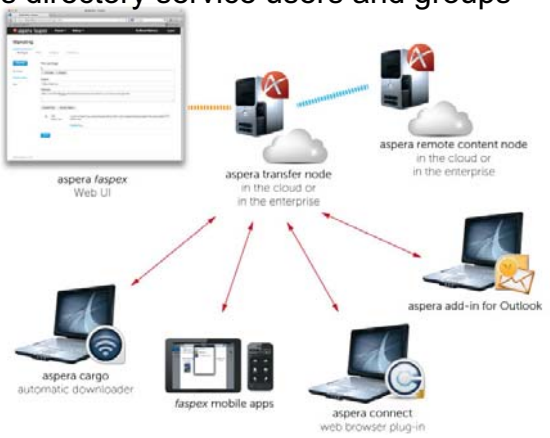
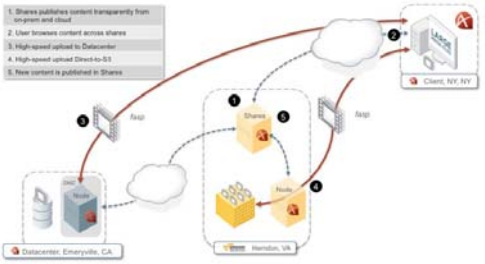


Figure 2-10. IBM Aspera Faspex

IBM Aspera Faspex presents a

IBM Training

IBM Aspera Shares



- Web-based interface to remote resources
- Content published from on-premises or cloud
- Single view of shared content across nodes
- Powered by FASP protocol for high-speed transfers
- Windows & Linux servers
- Direct drag-and-drop transfers between Shares
- Complete control of user permissions – browsing, uploading, downloading, etc.
- Integrates with Aspera Console for configuring, controlling, and monitoring all FASP transfers
- Requires Aspera Connect Browser Plug-Inplug-In

Figure 2-11. IBM Aspera Shares

IBM Aspera Shares is designed for companies that need to import or share content as large files and directories in multiple locations within their organization or with external customers and business partners. A single web interface consolidates browsing across all shared content and a powerful and flexible security model that provides a single management point with authorization, user management, and access control.

The IBM Aspera Shares web application can be accessed from most standard web browsers, and provides secure access to a consolidated view of all available content.

IBM Aspera Shares can run in the cloud and with the ability to connect to on-premises or in-cloud Aspera Transfer Server instances. You can use IBM Aspera Shares to use the unlimited storage and computing capacity of the cloud and to seamlessly tie together public and private storage.

IBM Training

IBM Aspera Console

Centrally manage and monitor all Aspera server transfer activity

Centralized administration and control
- Consolidated dashboard and detailed drill-down views
- Control over individual transfers' speed and priority
- Remote browsing of servers and transfer initiation

Transfer management and automation
- Automated one-time or recurring transfers, including multi-point "Smart Transfers" used for multi-point distribution
- Custom pre/post processing on managed servers
- Automatically forward any file to or from any managed transfer server
- Monitor status and performance of Aspera Sync sessions

Node management and user access control
- Remotely configure all node properties (bandwidth, encryption, email notifications)
- Define transfer settings and authorization policies by user or group

Auditing and reporting
- Centralized transfer history database with scheduled backups, age-based purging and ability to handle large transfer histories
- Built-in report templates and support for custom report fields
- Automated email notifications

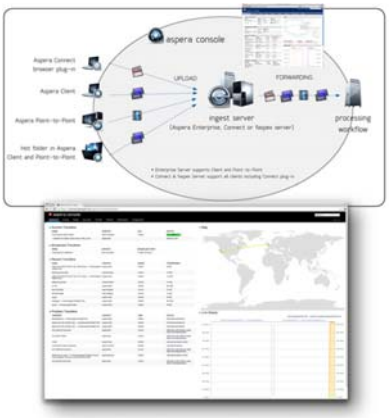


Figure 2-12. IBM Aspera Console

Aspera Console is a web-based management application that provides complete visibility over your Aspera high-speed transfer environment.

The dashboard shows all transfer activity and allows full control over file transfers, priority, and bandwidth control. New transfers can be initiated ad hoc or automated for specific scheduling. Administrators can remotely configure any connected Aspera server and use comprehensive transfer logs to create customized activity, usage and billing reports, and notifications.

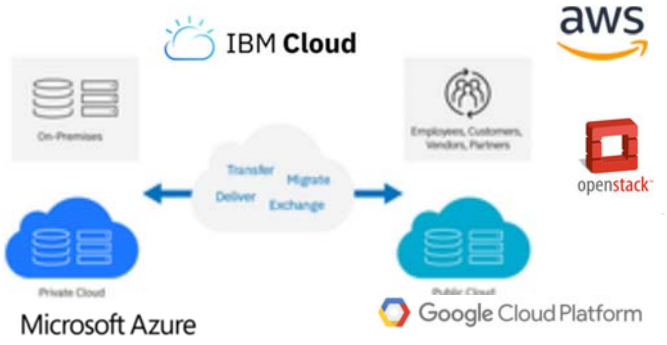
Aspera Console can be used to configure all node properties, such as bandwidth controls, encryption settings, directory creation masks, and email notifications. They can also define transfer and authorization settings by user or group, including bandwidth caps, transfer priorities, encryption, and security settings.

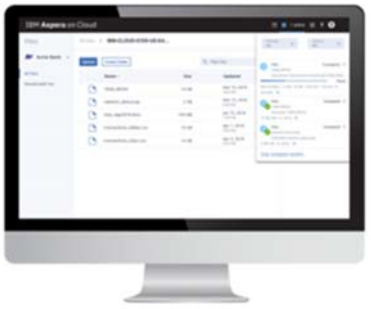
IBM Aspera Console provides a centralized transfer history database with automated backup and purging, and a customizable reporting engine for user-defined reports and advanced custom query development.

IBM Aspera Console users can also automate files transfers, including multi-site synchronization with IBM Aspera Sync, full business workflow processing with IBM Aspera Orchestrator, and integrate with third-party applications.

IBM Training

Aspera on Cloud: Overview





File sharing and delivery across hybrid-cloud environment
- Drag files and folders to transfer to or from any location
- Organize files and users in collaboration workspaces
- Distribute digital packages to recipients with email-like interface
- Integrates with on-premises IBM Transfer Servers

Central administration of hybrid environments
- Connect all cloud and on-premises storage
- Remotely manage transfer nodes with single interface

Direct to cloud technology and built-in clustering
- Bypassing two-phase transfers
- More computing resources automatically used when needed

Supported storage
- On-premises block storage (SAN, NAS, etc.)
- Object-based storage
- IBM Cloud Platform
- Open Stack Swift (IBM Cloud)
- Amazon web Services S3
- Google Cloud Platform
- Microsoft Azure BLOB
- Microsoft Azure files

Overview of IBM Aspera Software

Figure 2-13. Aspera on Cloud: Overview

IBM Aspera on Cloud is a hosted service that can quickly, securely and reliably move files and data sets of any size and type across a hybrid cloud environment. It unifies Aspera cloud solutions into a single offering with integrated reporting, advanced administration, and a new easy-to-use interface.

Using IBM Aspera on Cloud, organizations can seamlessly access and share data that is stored across multiple clouds and on-premises data centers. Internal and external users collaborate over the data in a secure environment that tightly controls access to content and application functions. Large files and data sets are transferred across the storage environment by using the Aspera FASP protocol, which overcomes the limitations of other file-transfer technologies to move data at maximum speed.

IBM Training

Aspera Product Integration: Overview

- Transfers between user and transfer server or transfer server and transfer server
- Transfer servers must have system, transfer, and Node API user accounts for web applications
- Web application can be installed on transfer server system or on dedicated system
- Web applications maintain separate databases for application user authentication
- Transfer servers do not need accounts for every web application user

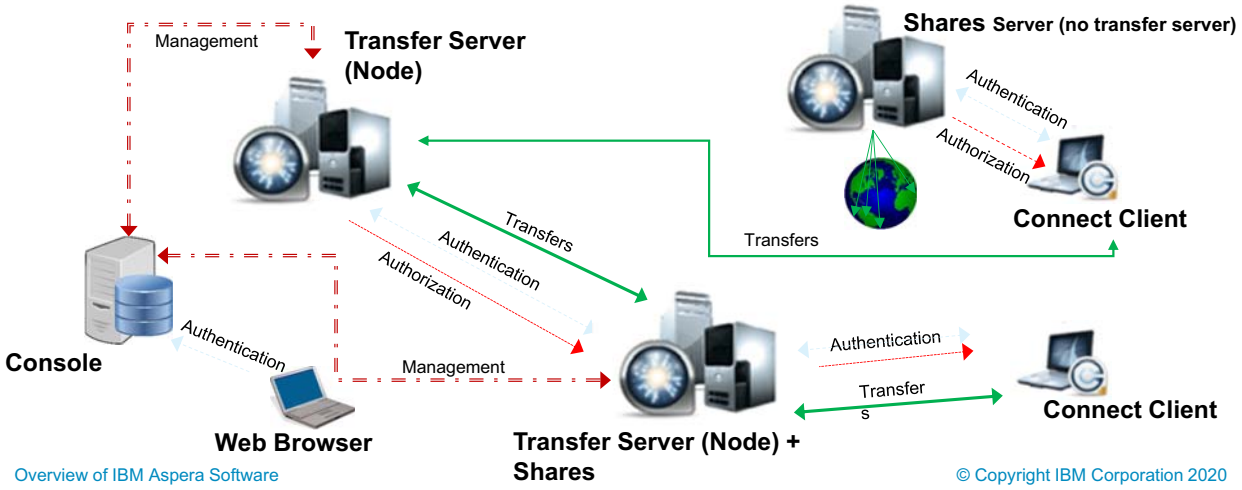


Figure 2-14. Aspera Product Integration: Overview

Aspera transfer servers (also referred to as a “node”) act as the engine for all data transfers. Transfers always occur between either an Aspera client system and an Aspera Transfer Server, or between Aspera transfer servers (connections shown in green).

Each transfer server can validate authentication requests (via SSH). Therefore, a transfer requires that the transfer server has a system user account that can be authenticated via SSH. But an Aspera user account (transfer user) that the transfer server uses to apply transfer control parameters is also required. And each transfer server must also have a Node API user that authenticates with the transfer server on behalf of an application user. Thus, individual system user accounts are NOT required for each application user.

IBM Aspera Shares and IBM Aspera Faspex server software can be installed on the same server as the transfer server software, or can be installed on a stand-alone system, depending upon customer needs. However, the Shares and Faspex software cannot be installed on the same system. Both applications maintain a MySQL database that conflict with each other when they are installed on the same system.

Aspera Console also has its own MySQL database, and is usually installed on a stand-alone system.

These MySQL databases store user account data necessary for the application to authenticate application users, along with the Node API account data that is used by the application to

authenticate with the transfer servers. Notice that the application is not part of the actual transfer path, but provides the authentication function for users who are configured for the application.

After the Aspera application server authenticates the user account, the user can initiate a request for a transfer. This request causes the Aspera application server to authenticate with a transfer server (that uses the Node API user credentials). After the transfer server authenticates the Aspera application server, the application server requests a transfer on behalf of the application user. The transfer server generates a token that provides authorization data. This token is sent to the application server, which then sends it to the client that requested the transfer. The token provides the details that the client system needs to transfer the file. Details of communications between the client, the application server, and transfer server are provided later in this course.

For now, it is sufficient to realize that the application servers themselves do not participate in the transfer. However, they do eliminate the need to configure each transfer server with all application user accounts, thus significantly reducing administrative effort.

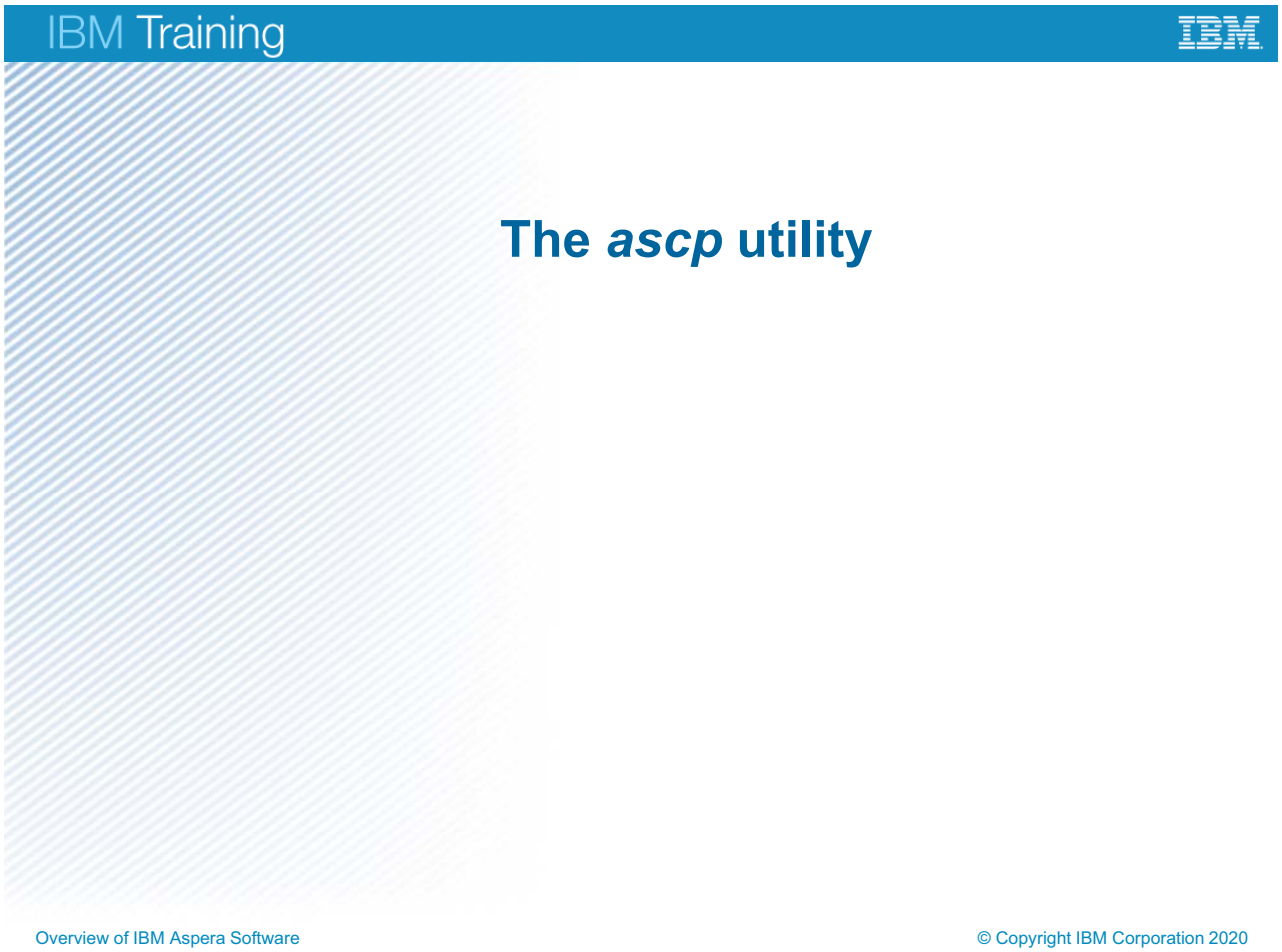


Figure 2-15. The ascp utility

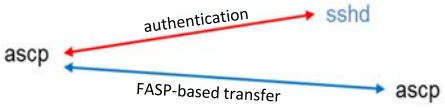
IBM Training

The *ascp* utility: Overview

Command-line implementation of FASP
Many options available

Transfers require *ascp* at each end (session)
ascp can be sender or receiver
Web interface ultimately uses ***ascp*** for transfers

Uses UDP port 33001
Other port values can be configured



Uses SSH for authentication
ascp contacts sshd on server

Verifies authorization token for web-based apps
Denies transfer when verification fails

Not implemented in Aspera web app software

Figure 2-16. The ascp utility: Overview

IBM Training

Transfer servers and *ascp*

- ***ascp*** process for each Aspera session
- Each ***ascp*** process operates independently
- Management port is common port for all ***ascp*** processes
- ***ascp*** reports metrics to management port for logging and other API purposes
- Node API used to access ***ascp*** metrics for web applications

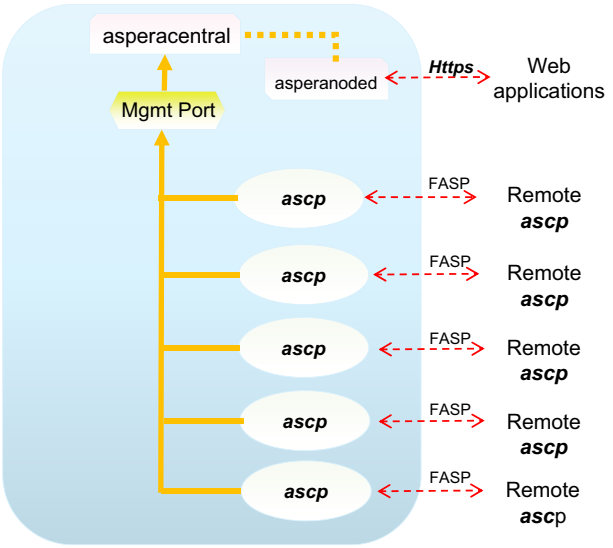


Figure 2-17. Transfer servers and ascp

Every Aspera session requires an ascp process on the transfer server, and each ascp process is independent of other ascp processes. An Aspera Transfer Server can have many active ascp processes at any time, and needs to be able to provide the transfer status and statistical data to other applications when requested.

Another process on the transfer server is a process that is called asperacentral. The asperacentral process is responsible for monitoring the status of each ascp process, collecting statistics from each ascp process about the transfer, and making that data available to applications that request it.

To facilitate the collection of data from all running ascp processes, Aspera Transfer Server software implements a management port for collecting transfer statistics. Each ascp process on the server connects to this management port, as does the asperacentral routine. The asperacentral process monitors the management port to collect this data and provides it to other applications when it is requested.

When an Aspera application like IBM Aspera Console needs to query the transfer server for those statistics, they connect to the asperanoded process, which in turn, interacts with asperacentral to extract the data.

IBM Training

Basic *ascp* connections

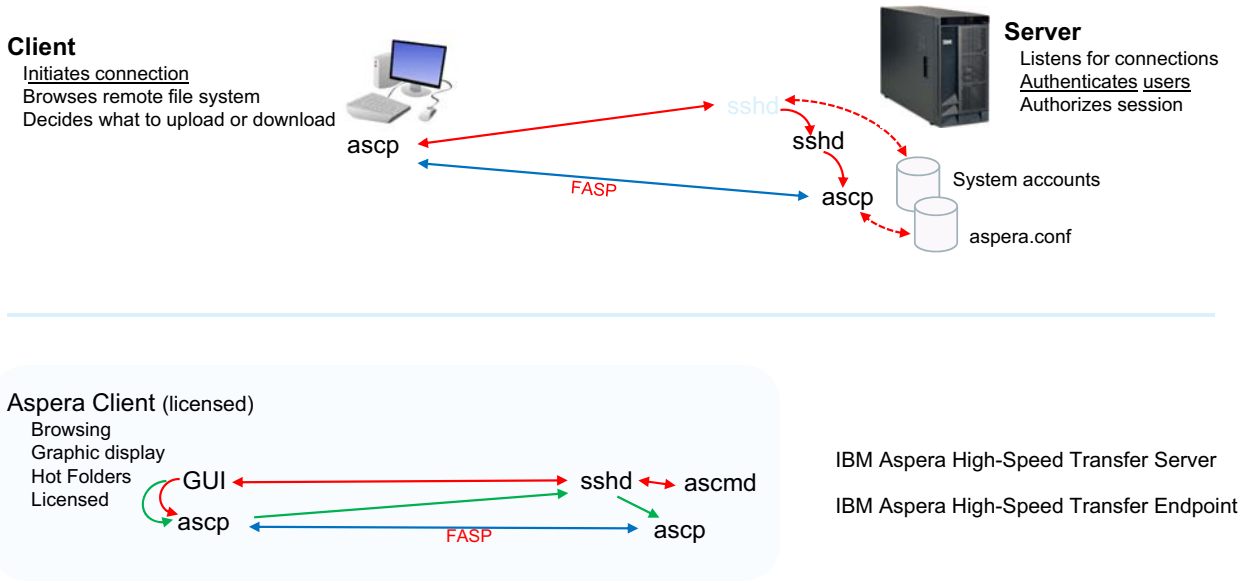


Figure 2-18. Basic ascp connections

As presented previously, all FASP-based transfers require an ascp process at each end of the connection. The ascp process can be run from the command line or through a GUI.

When Aspera transfers are run from the command line, a user must identify the files or directory they want to transfer. The ascp command (with appropriate options) is run on their local system to initiate the upload or download. The ascp binary first contacts the sshd daemon process on the server, which n turn creates a separate sshd process to handle the authentication request from the client system. If the client's credentials are successfully authenticated, an ascp process is started to support the requested transfer.

The Aspera Client software can be installed on a user's system to provide a GUI interface to the ascp routine. The GUI allows the user to view the contents of the assigned transfer server directory, navigate to other directories on the server, and initiate a transfer. Users highlight the files that they want to transfer and click the upload or download links provided in the GUI.

Underlying the Aspera Client GUI is the same procedure as the procedure used for command-line operations, but the connection and selection process is (or can) be automated. For example, connections to transfer servers can be configured with the appropriate credentials, thus eliminating the need for the user to enter that information each time they transfer files to or from that server. The transfer server software uses the ascmd process to provide the client software with a listing of files on the server. The user can view and navigate the directories on the server, then select the

files or directories to transfer. Ultimately, an ascp process on the client system connects with an ascp process on the server to support the FASP-based transfer.

IBM Training

Web applications and *ascp*

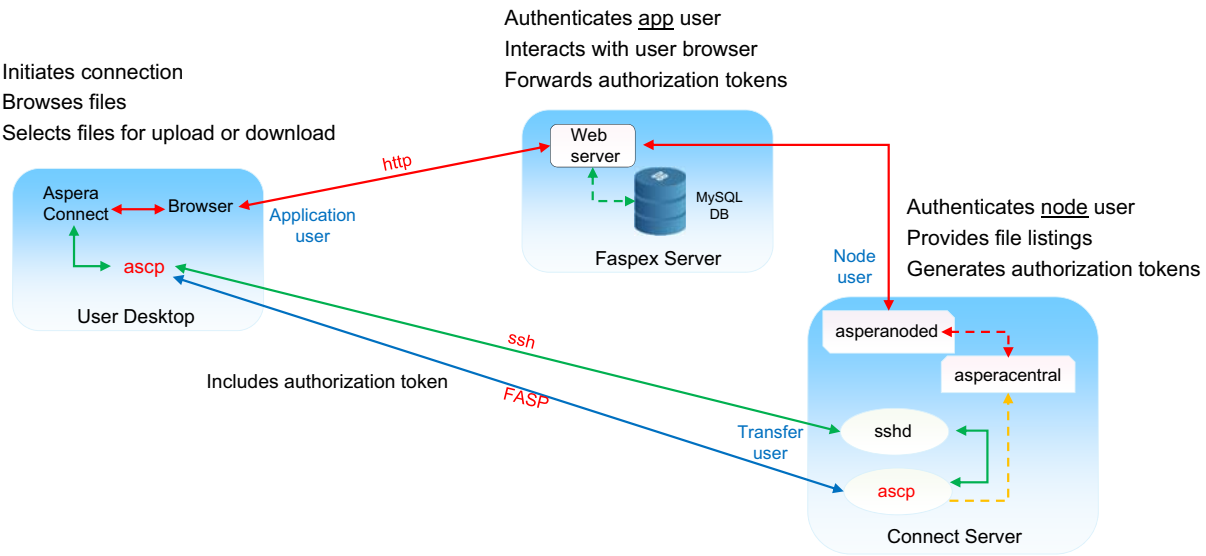


Figure 2-19. Web applications and ascp

A user can interact with Aspera web-based applications without the need for the Aspera client software. Users running a web browser contact the Aspera web application over a standard HTTP connection. The Aspera application authenticates the user's credentials (stored in application's MySQL database) and contacts the asperanoded process. The application uses the Node API credentials that are configured on the transfer server to authenticate, then requests a listing of files available to the user. After a user requests an upload or download, the web application contacts the transfer server, which in turn, generates a token that is passed back to the user's IBM Aspera Connect software. The IBM Aspera Connect software starts an ascp process, which authenticates with the identified transfer server. The ascp process then connects to the ascp process created on the server to complete the transfer.

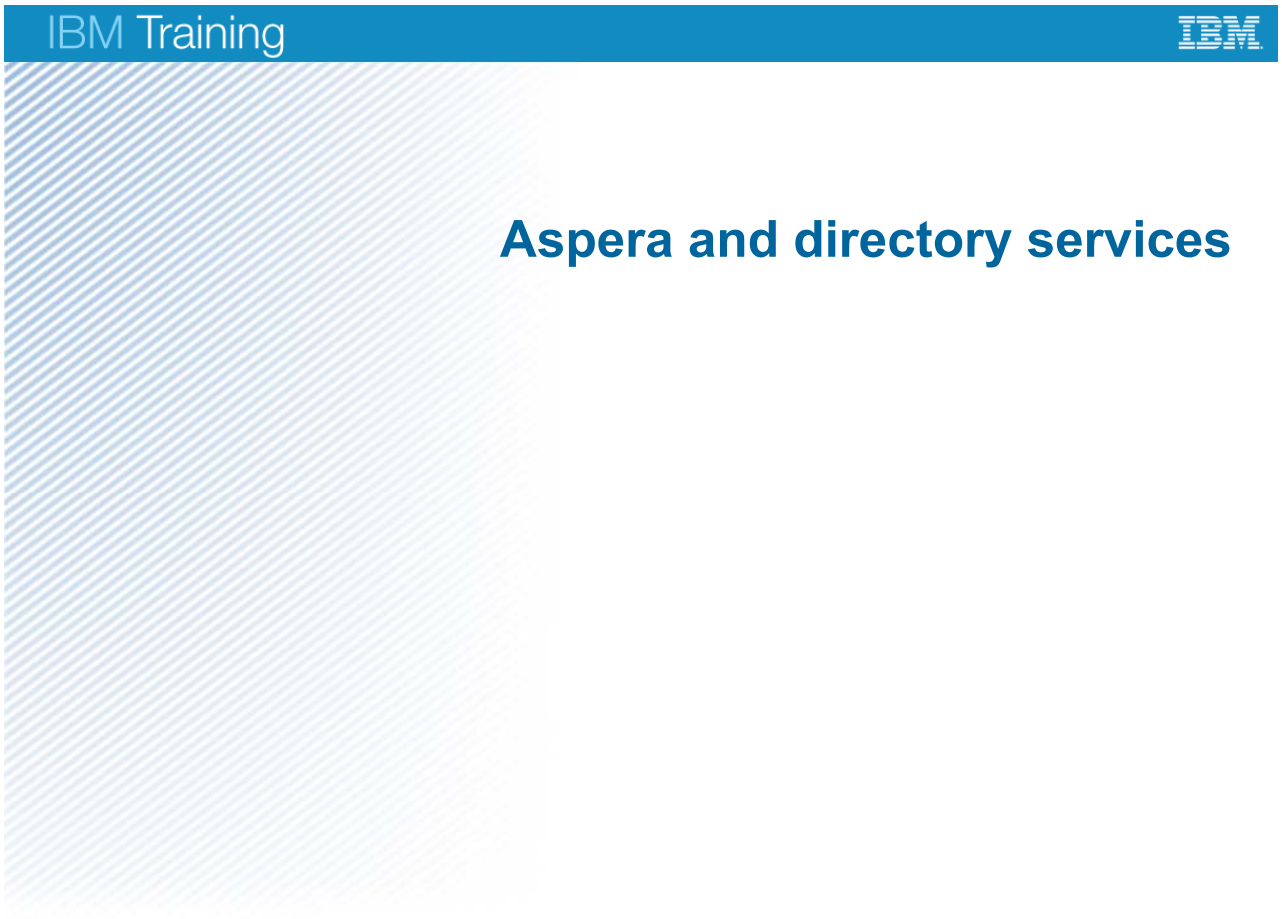


Figure 2-20. Aspera and directory services

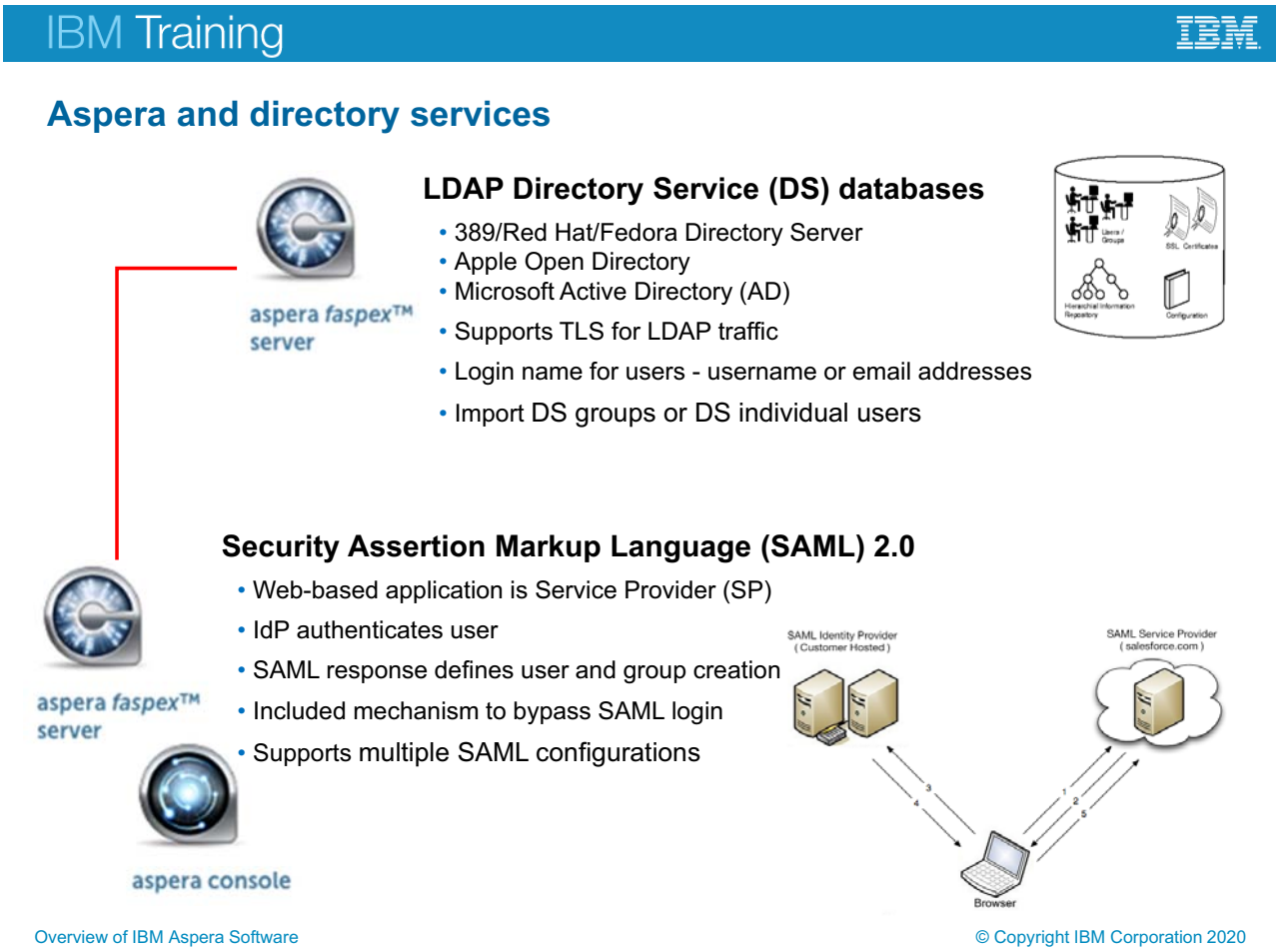


Figure 2-21. Aspera and directory services

Directory Services (DS)

Aspera web applications can be configured to work with Lightweight Directory Access Protocol (LDAP)-based directory servers like 389/Red Hat/Fedora, Apple's Open Directory, and Microsoft's Active Directory (AD).

IBM Aspera Faspex can be configured to use TLS to provide secure LDAP traffic. If enabled, the director server's port number is automatically changed (in the Faspex application) to port 636.

You can identify the type of login name for users of the directory servicer as either am email address or the user's name (or uid, depending upon the DS selected).

Faspex can import a DS group or DS individual users. When importing a DS group, all users who are listed under the group are added into Faspex. When adding a DS group, Faspex searches for groups recursively to import users. For example, if Group A contains Group 1, importing Group A also imports all of Group 1 members.

NOTE: When Faspex imports AD groups, it is bounded by the AD server parameter MaxValRange. If you need a larger AD group, change the MaxValRange parameter on the AD server.

Individual DS user accounts can also be imported, but only one user at a time. User's can be identified as an "admin", a manager, or a regular user (discussed in the Faspex module). After DS

users (or groups) are imported, the corresponding users can authenticate and log in to the IBM Aspera Faspex server.

Security Assertion Markup Language (SAML)

IBM Aspera Faspex, IBM Aspera Shares, and IBM Aspera Console can all be configured to work with SAML 2.0, an XML-based standard that allows secure web domains to exchange user authentication and authorization data. The IBM Aspera Faspex application acts as the SAML Service Provider (SP), which contacts a SAML Identity Provider (IdP) to authenticate IBM Aspera Faspex users.

When SAML is enabled, IBM Aspera Faspex redirects login requests to the IdP sign-on URL. The user signs in to the IdP and, if successful, then the SAML IdP sends a SAML assertion back to the client system.

When a SAML user successfully logs in to IBM Aspera Faspex the first time, Faspex automatically creates a new user account based on the information that is provided by the SAML assertion. If the SAML assertion also contains group information, and that group does not yet exist in the IBM Aspera Faspex database, the application automatically creates a new SAML group.

Faspex supports multiple SAML configurations on the same server. Faspex redirects users to the default SAML IdP. But if no default is specified, Faspex directs users to the local login page where users can choose to log in to publicly visible SAML configurations or log in locally.

Faspex users can bypass the SAML redirect by adding the text login?local=true to the end of the URL used to access the Faspex server:

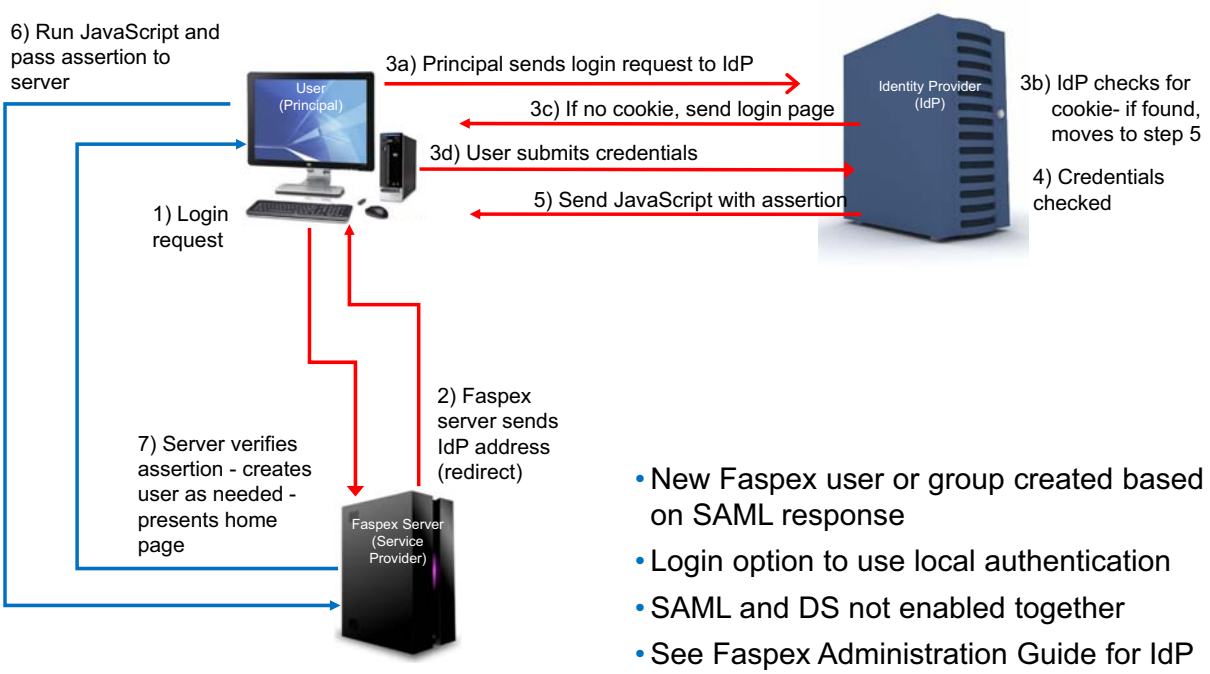
`198.26.212.24/aspera/Faspex/login?local=true`

This feature allows administrators to correct server settings, including mis-configured SAML setup, without logging in through SAML.

Do not enable both SAML and Directory Services at the same time. Select one or the other.!

IBM Training

Aspera and SAML



- New Faspex user or group created based on SAML response
- Login option to use local authentication
- SAML and DS not enabled together
- See Faspex Administration Guide for IdP requirements

Figure 2-22. Aspera and SAML

Security Assertion Markup Language (SAML), is an XML-based data format for exchanging authentication and authorization data between secure web domains. The SAML specification defines three roles: The principal (typically a user), the identity provider (IdP), and the service provider (SP).

Principal

The Principal is the user who is requesting access to system resources.

Identity Provider

An Identity Provider (IdP) is a service that creates, maintains, and manages identity information for principals and provides principal authentication to other service providers within a federation, such as with web browser profiles. The IdP asserts the identity of a user.

Service Provider

A Service Provider (SP) is a role that is provided by a system entity or application, which provides services to principals or other system entities or applications. An SP receives the assertion from the IdP and passes it on to relevant applications.

Assertions

SAML uses security tokens that contain assertions to pass information about a principal (usually a user) between an identity provider and a web service. Each assertion contains data that is used by

service providers to make access decisions. SAML supports three types of assertions: authentication, attribute, and authorization. Authentication assertions validate a user's identity, attribute assertions contain specific information about the user, and authorization assertions identify what a user is authorized to do.

SAML Operation with Aspera Faspex & Console

IBM Aspera web applications support SAML 2.0, and can be configured as a SAML SP that contacts a separate IdP to authenticate users for IBM Aspera Faspex or Console to access secure content.

If SAML is configured on the Faspex or Console server and a user connects to that server, the client's browser is sent a redirect to the IdP's URL. The redirect from the application server provides the IdP's URL and the data that the IdP needs to create a JavaScript. The JavaScript is sent to the client system, where it is run to connect to the Aspera application server.

If the IdP previously authenticated the user, the user's browser login request includes any previous SAML assertions in the form of a cookie. The IdP examines the cookie, and if f the cookie is valid and still active, the IdP creates an updated assertion and sends it and the JavaScript back to the user's browser. If no cookie is found in the browser's login request, the IdP sends the user browser a login page.

The user provides their credentials and the IdP authenticates the user against the identity store.

After the user's browser receives the assertion and JavaScript from the IdP, it automatically runs the JavaScript to connect to the IBM Aspera Shares server and sends the assertion it received from the IdP.

The Aspera application server evaluates the assertion that is sent with the user's browser connection to determine whether it is valid and verifies that the roles identified are appropriate. It also determines whether the user is configured in the Aspera application's MySQL database. If the user is not defined in the MySQL database, the applications server creates an account for that user. After the account is created, or verified, the server sends the user their home browser page.

All IBM Aspera web-based transactions occur in the normal manner, without further input from the IdP.

NOTE: When SAML is enabled, IBM Aspera Faspex creates a user account based on the information that is provided by a SAML response. Therefore, the Aspera application user account is not created manually.

IBM Training

What you learned

- IBM Aspera High-Speed Transfer Server – primary transfer server engine
- IBM Aspera High-Speed Endpoint - differs only in # of simultaneous connections
- IBM Aspera Proxy Server – secure transfers to or from external clients
- IBM Aspera Sync – file-based replication and synchronization
- IBM Aspera Connect – browser plug-in provides ***ascp*** to client
- IBM Aspera Command Line Interface – implements ***ascp*** on client system
- IBM Aspera Desktop Client – graphic interface for file transfer on client system
- IBM Aspera Shares – enhanced access to remote directories
- IBM Aspera Faspex – mail notification for FASP transfers
- IBM Aspera Console – management application for Aspera products
- IBM Aspera Orchestrator – workflow management
- Transfer servers provide primary engine for all FASP transfers – implemented in IBM Aspera Transfer Server and IBM Aspera Endpoint client
- ***ascp*** is the routine that implements the FASP protocol
- Aspera web applications provide a graphic and command-line interface for FASP transfers
- ***ascp*** processes communicate transfer statistics to **asperacentral**, which is accessed through Node API
- Aspera web applications integrate with LDAP Directory Services and SAML
- DS users and groups can be imported to web applications

Figure 2-23. What you learned

IBM Training

Unit summary

- Identify the function of common IBM Aspera software
- Outline how IBM Aspera software products interact with each other
- Describe the ascp process and how it communicates with Aspera products

Figure 2-24. Unit summary

IBM Training

Review Questions (1 of 2)

1. True or False: All IBM Aspera software products require a connection to an IBM Aspera transfer server

2. Which of the following statements is true regarding IBM Aspera Console? Select all that apply:

   A. IBM Aspera Console can be configured to be able to modify the configuration of any IBM Aspera High-Speed Transfer Server

   B. Transfer between 2 IBM Aspera High-Speed Transfer Servers can be initiated from IBM Aspera Console

   C. IBM Aspera Console cannot be used to modify default bandwidth parameters

   D. A single IBM Aspera Console server can manage all IBM Aspera High-Speed Transfer Servers in an organization, regardless of where located

Figure 2-25. Review Questions (1 of 2)

IBM Training

Review questions (2 of 2)

3. True or False: IBM Aspera Faspex uses an email server to transfer files to both internal and external users

4. Which of the following statements is correct regarding the ***ascp*** program? Select all that apply:

   A. The ***ascp*** program uses both SSH and FASP when it initiates a transfer

   B. The ***ascp*** program is implemented only on the IBM Aspera High-Speed Transfer Server and does not require any additional software on the client system

   C. Each ***ascp*** process on an IBM Aspera High-Speed Transfer Server uses a management port on the server to report its transfer statistics

   D. The ***ascp*** program is not a part of the IBM Aspera web applications,

Figure 2-26. Review questions (2 of 2)

IBM Training

Review answers (1 of 2)

1. True or False: All IBM Aspera software products require a connection to an IBM Aspera transfer server
   The answer is True

2. Which of the following statements is true regarding IBM Aspera Console? Select all that apply:
   A. IBM Aspera Console can be configured to be able to modify the configuration of any IBM Aspera High-Speed Transfer Server
   B. Transfer between two IBM Aspera High-Speed Transfer Servers can be initiated from IBM Aspera Console
   C. IBM Aspera Console cannot be used to modify default bandwidth parameters
   D. A single IBM Aspera Console server can manage all IBM Aspera High-Speed Transfer Servers in an organization, regardless of where located

   The answer is A, B, and D

Figure 2-27. Review answers (1 of 2)

IBM Training

Review answers (2 of 2)

3. True or False: IBM Aspera Faspex uses an email server to transfer files to both internal and external users
   Faspex uses email to notify users that files are available for download. The actual file transfer is completed by using the FASP protocol

4. Which of the following statements is correct regarding the ***ascp*** program? Select all that apply:
   A. The ***ascp*** program uses both SSH and FASP when it initiates a transfer
   B. The ***ascp*** program is implemented only on the IBM Aspera High-Speed Transfer Server- and does not require any additional software on the client system
   C. Each ***ascp*** process on an IBM Aspera High-Speed Transfer Server uses a management port on the server to report its transfer statistics
   D. The ***ascp*** program is not a part of the IBM Aspera web applications
   The answer is A, C, and D

Figure 2-28. Review answers (2 of 2)

Unit 3. Installing IBM Aspera High-Speed Transfer Server

Estimated time

02:30

Overview

This unit presents the necessary tasks to prepare a system for and installation of IBM Aspera High-Speed Transfer Server software

How you will check your progress

- Exercise

IBM Training

Unit objectives

- Identify the prerequisites for a successful deployment of IBM Aspera High-Speed Transfer Server
- Configure the system firewall to support Aspera transfers
- Secure access to Aspera services by modifying the SSH configuration
- Locate and install the appropriate IBM Aspera High-Speed Transfer Server software (Windows and Linux)
- Explain the purpose of the Aspera service account on Windows systems running IBM Aspera High-Speed Transfer Server software
- Verify installation success by transferring files to and from the Aspera Demo Server
- Configure Aspera log redirection

Figure 3-1. Unit objectives

The Aspera FASP protocol is implemented within the ascp binary routine, which is included in Aspera Transfer Server software and other Aspera client software (for example, Aspera Desktop Client and Connect browser plug-in). The ascp binary can be accessed from the command line and has an extensive set of options for managing a transfer.

Because ascp is the binary that implements the FASP protocol, all FASP-based transfers require an ascp process at each end. Aspera transfers are always between ascp processes.

By default, the FASP protocol (which is implemented in the ascp binary) uses UDP port 33001, unless the server and client are configured to use a different port for FASPFASP-based transfers.

The ascp binary is responsible for several tasks as part of handling transfers. Besides managing the transfer itself, the ascp program is responsible for authenticating with the transfer server as part of the initiation of a transfer (details presented later in this module). Aspera web applications pass authorization tokens to a client's Connect software, which then hands the token to the ascp routine. The ascp program presents the token to the server as part of the file transfer initiation after successful authentication via SSH.

It is important to remember that the ascp binary is NOT included in Aspera web-based application software. However, the transfers that are performed through the web-based applications use the ascp program on the transfer server and the ascp program on the client system.

IBM Training

Topics

- Prepare the operating system
- Install Aspera software
- Upgrading, downgrading, and reinstalling Aspera software
- Configure Aspera logging

Figure 3-2. Topics

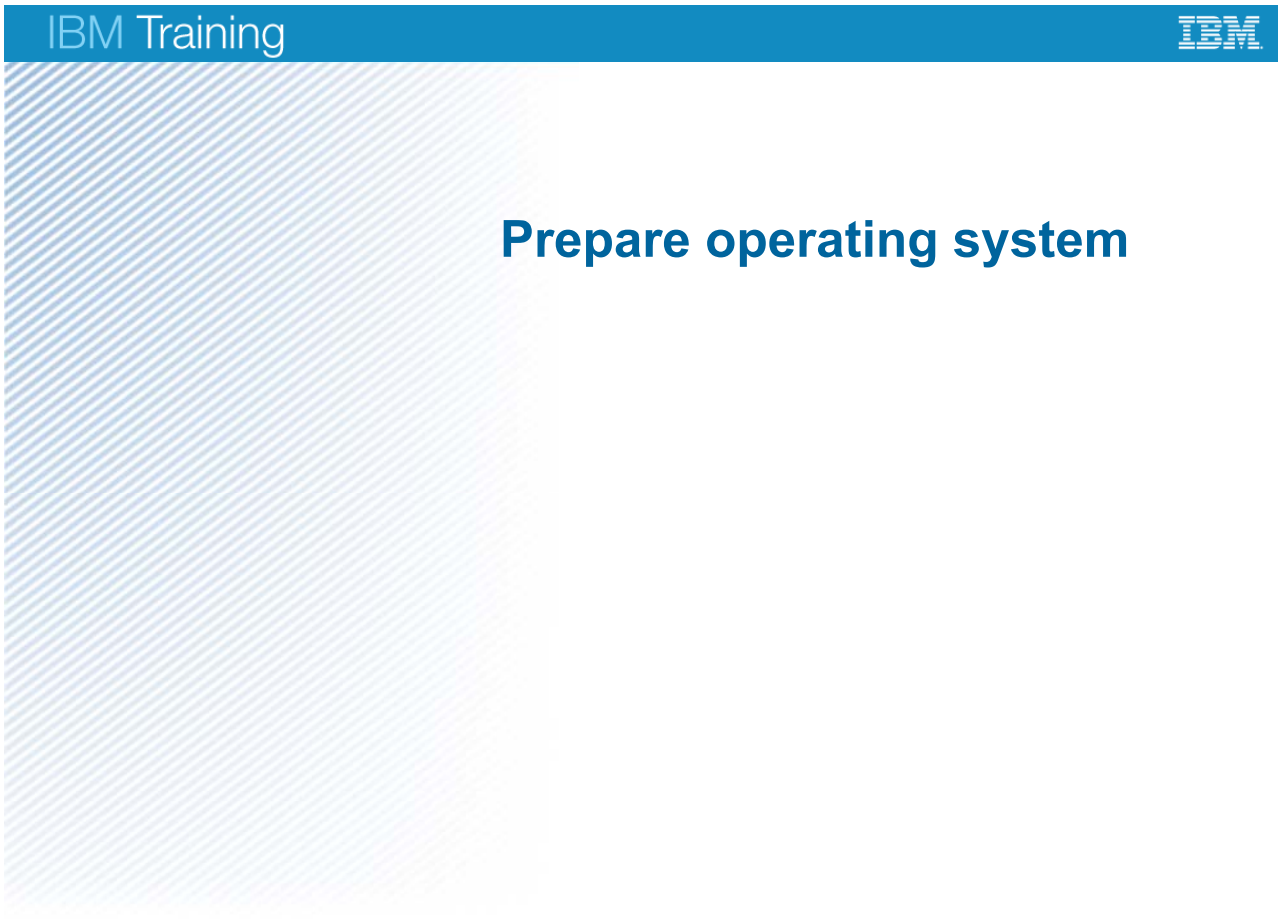


Figure 3-3. Prepare operating system

IBM Training

System prerequisites

Windows Platform

- Windows 7, 8, 10; Windows Server 2008 R2, 2012 R2, 2016
- Aspera license file
- Access to domain administrator account (for Active Directory environments)
- Access to run Windows Management Instrumentation (MWI)
- Screen resolution 1024 X 768 or higher
- Active Perl to enable Perl scripts (if Pre/Post processing is needed)

Aspera High-Speed Transfer Server software https://www.ibm.com/aspera/downloads/

Download requires credentials provided with customer purchase

LINUX platform
(Redhat or Debian)

- Aspera license file
- Linux kernel 2.4 or higher
- libc version GLIB 2.5 or higher
- SSH server (version 5.2 or higher recommended)
- Configure SELinux - “Permissive” or “Disabled” advised, but “Enforcing” OK with some configuration (/etc/selinux/config)

Installing IBM Aspera High-Speed Transfer Server © Copyright IBM Corporation 2020

Figure 3-4. System prerequisites

Some basic requirements must be met before installing the Aspera IBM Aspera HSTS software, depending upon which operating system is in use. The graphic indicates requirements for some, but not all, supported operating systems. For details about specific operating systems, check the Requirements section of the platform-specific documentation.

Platform-specific documentation can be downloaded from the IBM Aspera download page. You can download both the appropriate software and documentation.

NOTE: You need a valid login and password (provided to your organization by IBM) to download the software. Downloading documentation does NOT require these credentials

IBM Aspera HSTS Software

Installer routines for all Aspera software can be downloaded from the IBM Aspera download website www.ibm.com/aspera/downloads. The page presents links for downloading various Aspera software, including the IBM Aspera High-Speed Transfer Server.

You select the product that you want to download (in this case, IBM Aspera HSTS, which opens a page where you select Transfer server software. Another page opens where you can select which Aspera server product you want to download. The IBM Aspera High-Speed Transfer Server listing provides options to download the software or view the documentation.

NOTE: After you select the version of software you want to download, you are prompted for a valid login and password

SELinux Configuration

Security-Enhanced Linux (SELinux) is a Linux kernel security module that provides a mechanism for supporting access control security policies. By default, SELinux defines the access and transition rights of every user, application, process, and file on the system. SELinux then governs the interactions of these entities with a security policy that specifies how strict or lenient an installation should be. The default behavior for SELinux is to run in enforcing mode, which is the most restrictive, and causes issues for the operation of the Aspera HSTS software, unless specific changes are implemented.

Aspera recommends modifying the `/etc/selinux/config` file to set SElinux's mode to be either permissive or disabled by changing the SELINUX= parameter as follows:

```
SELINUX=disabled
 or
SELINUX=permissive
```

IBM Training

Configure SELinux (if “enforcing”)

```
# sestatus
SELinux status:                 enabled
SELinux mount                   /selinux
Current mode:                   enforcing
Mode from config file:          enforcing
Policy Version:                 24
Policy from config file:        targeted
```

SELinux is **enforcing**

Allow the use of **aspshell**

```
# echo /bin/aspshell-r >> /etc/shells
# echo /bin/aspshell >> /etc/shells
# semanage fccontext –a –t shell_exec_t “/bin/aspshell”
# restorecon –v /bin/aspshell
```

Allow access to **authorized_keys** file (Faspex & Shares)

```
# semanage fcontext –a –t ssh_home_t “/path/to/user_homedirectory/.ssh(/.*
# restorecon –Rv /path/to/user_homedirectory/.ssh
```


Figure 3-5. Configure SELinux (if “enforcing”)

If SELinux is required on your organization’s systems, you need to modify some SELinux configuration settings to allow Aspera software to work as expected.

Configure SELinx to allow *aspshell*

Aspera recommends modification of all system user accounts that are used for Aspera transfers. Replace the user’s login shell with the aspshell routine that is provided by Aspera as part of the transfer server software installation. The aspshell routine restricts users from performing any tasks other than those tasks required when transferring files, thus minimizing potential security breaches.

If SELinux is configured to be enforcing, you need to perform the following tasks to allow the use of aspshell:

```
echo /bin/aspshell-r >> /etc/shells
echo /bin/aspshell >> /etc/shells
semanage fccontext –a –t shell_exec_t “/bin/aspshell”
restorecon –v /bin/aspshell
```

Configure SELinux to allow the authorized_keys file to be accessed (Faspex & Shares)

If your transfer server is used with Aspera Faspex or Aspera Shares web applications, you need to set an SELinux policy for the transfer user account used in your application. For Faspex, the

transfer user account name is usually faspex. In Shares, the transfer user account can be shares, but can be something different. You need to confirm the actual transfer user account names on your system and substitute those values for the `/path/to/user_homedirectory` entry in the following commands:

```
semanage fcontext –a –t ssh_home_t “/path/to/user_homedirectory/.ssh(/.*
restorecon –Rv /path/to/user_homedirectory/.ssh
```

IBM Training

Installation process

- Configure firewall
- Secure SSH
  - Modify SSH port
  - Restrict user access
  - Update SSH authentication methods
- Run Aspera installer software
- Add license key
- Configure transfer server authentication
- Configure connections
- Test locally initiated transfer

Figure 3-6. Installation process

Some modifications to the operating system configuration need to be made to support the Aspera HSTS software. Details about these modifications, and the initial configuration of the Aspera software, are discussed in the next few pages of this module.

IBM Training

Firewall configuration

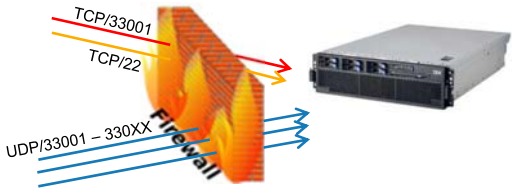
Corporate & Transfer Server System Considerations

Allow inbound SSH on TCP/33001 (TCP/22 optional)

Allow inbound FASP transfers on UDP/33001

Multiple concurrent clients - range of inbound UDP ports (on Windows/Mac OS X/FreeBSD/Isilon)
For example, 10 concurrent users: UDP/33001 – UDP/33010

If Vlinks are used across multiple servers, open UDP/55001 (default) for multicast traffic



Client System Considerations

Allow outbound SSH on TCP/33001

Allow outbound FASP transfers on UDP/33001

Multiple concurrent clients - range of outbound UDP ports
For example, 10 concurrent users: UDP/33001 – UDP/33010

Installing IBM Aspera High-Speed Transfer Server

Figure 3-7. Firewall configuration

An Aspera server runs one SSH (Secure Shell) server on a configurable TCP port (22 by default).

NOTE: Aspera strongly recommends running the SSH server on a non-default port (for example, 33001) to ensure that your server remains secure from SSH port scan attacks

Corporate and Server Firewall Considerations

The corporate firewall must allow the open TCP port to reach the Aspera server.

NOTE: It is not uncommon to completely disable the firewall on the server where Aspera HSTS software is installed, depending upon the deployment environment.

When an Aspera client initiates a transfer, the client opens an SSH session to the server on the designated TCP port and negotiates the UDP port over used for the transfer.

Your corporate firewall should be configured as follows:

Inbound UDP/33001: The port for FASP transfers, which use UDP/33001 by default. However, the server can also choose to run FASP transfers on another port.

The local firewall (on system where Aspera HSTS software is installed) must not block the SSH and FASP transfer ports. If your server is located completely within the LAN and is not located within a DMZ, you can disable the server firewall.

If you use Vlinks, you need to allow the Vlink UDP port (55001 is the default port for Vlinks) access in order to support multicast traffic.

NOTE: Aspera strongly recommends allowing inbound connections for SSH on TCP/33001, and disallowing inbound connections on TCP/22. If you have a legacy customer base that uses TCP/22, then you can allow inbound connections on both ports

For Aspera servers that have multiple concurrent clients, the Windows, Mac OS X, FreeBSD, and Isilon operating systems do not allow the Aspera FASP protocol to reuse the same UDP port for multiple connections. Thus, if you have multiple concurrent clients and your Aspera Transfer Server is running on one of these platforms then you must allow inbound connections on a range of UDP ports. The range of ports is equal to the maximum number of concurrent FASP transfers expected. These UDP ports are opened incrementally from the base port, which is UDP/33001, by default. For example, to allow 10 concurrent FASP transfers, allow inbound traffic from UDP/33001 to UDP/33010.

Client System Firewall Considerations

Typically, consumer and business firewalls allow direct outbound connections from client computers for both TCP and UDP. No configuration is required for Aspera transfers in this case. However, in the special case of firewalls disallowing direct outbound connections, such as 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), or, if required, a range.

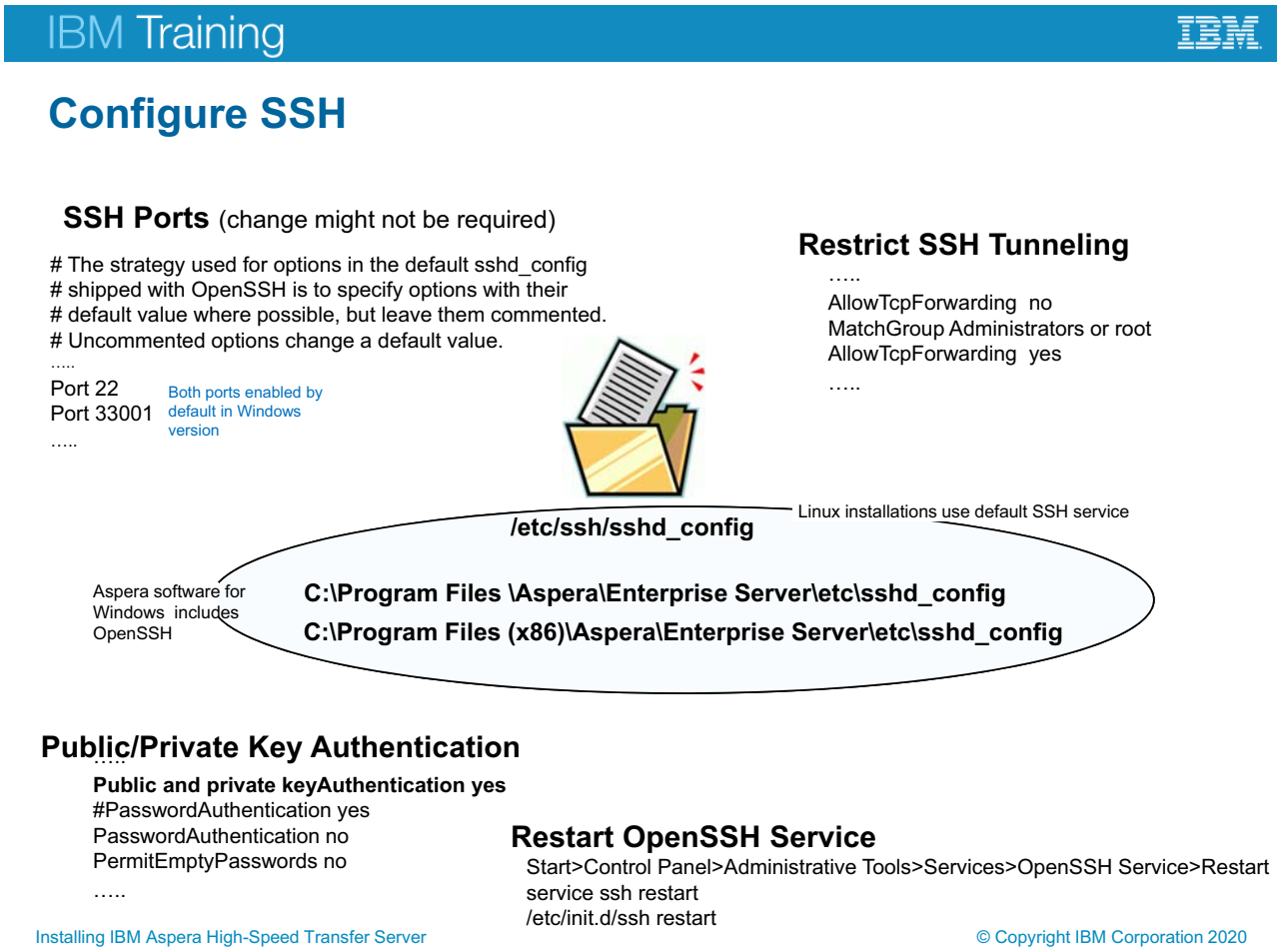


Figure 3-8. Configure SSH

The IBM Aspera HSTS uses SSH, which is configured with the sshd_config file to identify numerous parameter values that define the behavior of the SSH server.

The values in the `sshd_config` file identify which ports SSH are monitored for connection requests. This configuration file also specifies which other SSH functions are allowed:

- Enabling key authentication.
- Locating files containing SSH key values.
- Controlling what user accounts are allowed access via SSH.

NOTE: OpenSSH is normally installed as part of the Aspera HSTS installation for Windows platforms. You need to install the Aspera HSTS software before you can configure SSH, unless and existing SSH service (such as Cygwin) is installed. If an existing SSH service installed, you can customize the Aspera HSTS installation to avoid conflicts with you existing implementation.

The Linux version of Aspera HSTS uses the default SSH service that is provided by the Linux operating system, which can be configured before installing the Aspera HSTS software.

SSH Ports

SSH servers normally 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 (up to 65535). To standardize the port for use in Aspera transfers, Aspera recommends using TCP/33001. Remote Aspera application connections attempt to establish an SSH connection using the default port TCP/33001. However, if the connection fails, the application attempts the connection using port 22.

NOTE: Aspera HSTS for Windows ships with OpenSSH listening on both TCP/22 and TCP/33001 ports. Aspera recommends exposing TCP/33001 through your organization's firewall and disabling TCP/22.

Disable Non-admin SSH Tunneling

If you are using OpenSSH version 4.4 or newer, disable SSH tunneling to avoid potential attacks; only allow tunneling from users in the Administrator group on Windows platforms, or root on Linux platforms. To disable non-administrative SSH tunneling, add the following lines to the end of the `sshd_config` file (or modify them if they exist):

- `AllowTcpForwarding  no`
- `Match Group Administrators or root (depending upon the operating system in use)`
- `AllowTcpForwarding  yes`

NOTE: 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 or address) are met. If you run an OpenSSH version older than 4.4, the Match directive is available and Aspera recommends updating to the latest version.

NOTE (Linux Platforms): 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 discussion of modifying shell access later in this module.

Public Key Authentication

Public key authentication is an alternative means of identifying yourself to a login server, instead of typing a password. If all password-based authentication methods are disabled, public key authentication can prevent brute force SSH attacks. 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 the PubkeyAuthentication yes entry and comment out PasswordAuthentication yes. Details about configuring SSH public and private keys for users are presented in the Configuring and Managing Aspera Users module of this course.

Note: It is possible to use password authentication along with public key authentication by leaving the PasswordAuthenticaion parameter at yes. However, if you choose to leave password authentication enabled, be sure the PermitEmptyPasswords parameter is set to no.

NOTE (Linux Platforms): OpenSSH defaults to allowing root logins; however, disabling root access helps to maintain a more secure server. Aspera suggests commenting out the PermitRootLogin yes parameter in the sshd_config file and adding PermitRootLogin No. Administrators can use the *su* command if root privileges are needed.

Restart SSH Server

After you modify the `sshd_config` file, you need to restart the SSH Server to recognize your changes:

On Windows systems, use `Start >Control Panel >Administrative Tools > Services > OpenSSH > Restart`

Linux Platforms: sudo `service ssh restart` or `etc/init.d/ssh restart` (depending upon the version of Linux).

NOTE: Restarting the SSH server does not impact currently connected users.

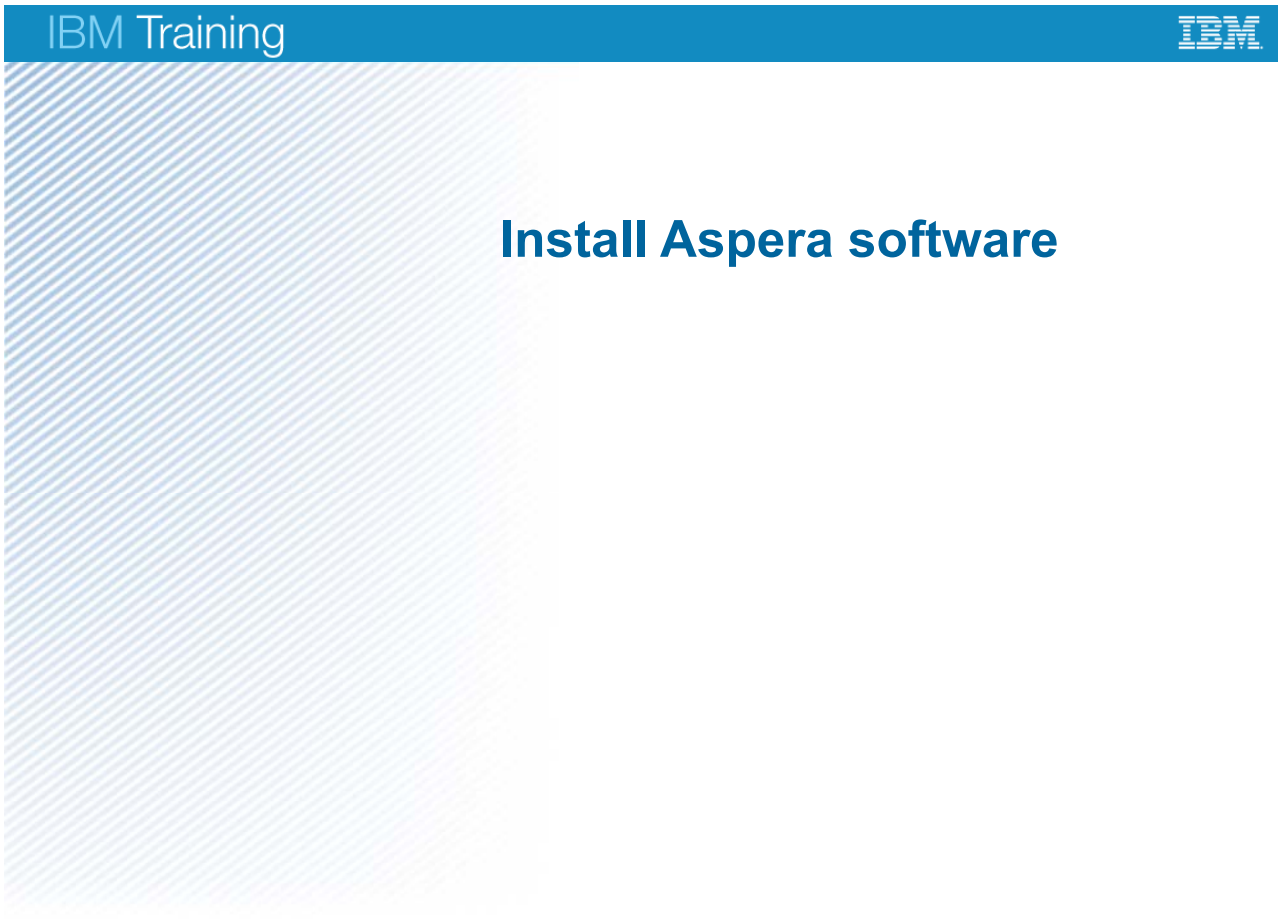
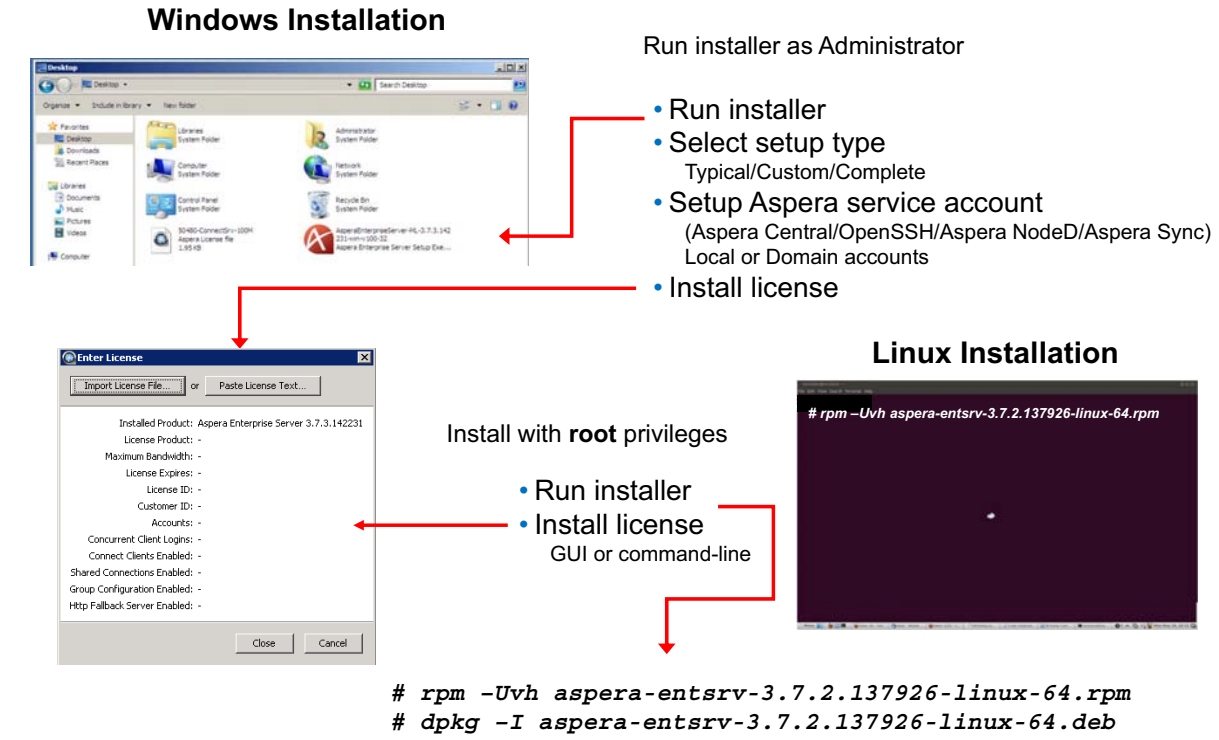


Figure 3-9. Install Aspera software

IBM Training

Install IBM Aspera High-Speed Transfer Server software



Installing IBM Aspera High-Speed Transfer Server

Figure 3-10. Install IBM Aspera High-Speed Transfer Server software

Run the Aspera installer routine that you downloaded from the IBM website. The Windows version of the installer is an executable (.exe) file, and can be run by clicking the appropriate icon. The Linux version of the installer varies, depending upon which Linux is running.

NOTE: On Windows 7 or Windows 2008 with UAC (User Account Control) enabled, you must run the installer as an Administrator.

Windows Installations

Type of setup

The installer routine prompts you to select the type of setup you want to run:

- **Typical**: Install the standard Aspera HSTS, including an SSH Server (OpenSSH)
- **Custom**: Select the features and the path to install (follow on-screen instructions)
- **Complete**: Install all features, including OpenSSH and Connect Server Web User Interface (Web UI cannot be used without a Connect Server license)

*Aspera Service Accoun*t

The Aspera service account runs services for Aspera products, including asperacentral, OpenSSH, asperanoded, and Aspera Sync.

The default user name for the Aspera service account is *svcAspera*.

If your system is not joined to a Windows domain, a local account (for example, *svcAspera*) is all that is required to run Aspera services. If the server is part of a Windows domain, or if you need to provision Active Directory accounts, you need to follow the instructions provided in the Aspera HSTS Admin Guide for Windows.

If you use the local account *svcAspera*, enter a password for the account.

If you use domain accounts, run the services with a domain account that is in the local administrator's group. You must create this domain account in the Domain Controller first, and the user name must be in the form: *username@fully.qualified.domain.name*, for example, *svcAspera@acme.com*.

NOTE: If you use the Aspera Transfer Server system with Aspera Console, the Aspera service account must be set up as a transfer user within the Aspera software. If you are performing a clean installation (no previous version of Aspera software), only the service account is created, NOT the corresponding transfer user. In this case, you create the transfer user account manually (details are provided in another module of this training).

Launch the Aspera GUI

Start the Aspera GUI from the Start Menu > All Programs >Aspera > Enterprise Server > Enterprise Server menu selection.

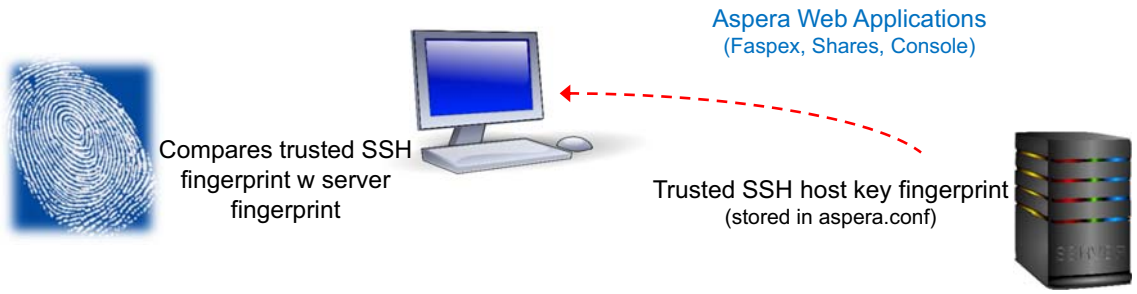
On a new installation, you are prompted for a valid license key.

Linux Installations

Invoke the Aspera installer routine with root privileges with the appropriate software installer tool for your version of Linux, for example, rpm or dpkg.

IBM Training

Transfer server authentication



Protects against server impersonation (man-in-the-middle) attacks
Not required – recommended

Add host fingerprint to aspera.conf

 # ssh-keygen –E sha1 –l –f /etc/ssn/ssh_host_key.pub

key strength fingerprint key file key type

2048 SHA1:31:40:42:d1:08:23:26:e9:0a:dd:13:57:0a:91:82:c2 /etc/ssh/ssh_host_key.pub (RSA1)

 # asconfigurator –x "set_server_data:ssh_host_key_fingerprint,31:40:42:d1:08:23:26:e9:0a:dd:13:57:0a:91:82:c2"

Figure 3-11. Transfer server authentication

For transfers initiated by a web application (such as Faspex, Shares, or Console), the client browser requests the transfer with the server using an HTTPS connection. The Aspera FASP engine on the client machine connects to the transfer server via UDP. In so doing, the client machine 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. The client confirms the server's authenticity by comparing the server's fingerprint with the trusted fingerprint.

Note: If a fingerprint is specified in `aspera.conf` and HTTP fallback is enabled, when the client falls back to HTTP, server SSL certificate validation (HTTPS) is enforced. If the server has a self-signed certificate, the validation fails; a properly signed certificate is required.

Configure Host Fingerprint

You can view the host's SHA1 fingerprint value by executing the following command (the response is shown below each command string):

Linux: `ssh-keygen –E sha1 –l –f /etc/ssh/ssh_host_key.pub` (or your own key)

2048 SHA1:31:40:42:d1:08:23:26:e9:0a:dd:13:57:0a:91:82:c2 /etc/ssh/ssh_host_key.pub (RSA1)

Windows: `ssh-keygen –E sha1 –l –f "C:\Program Files (x86)]\Aspera\Enterprise Server\etc\ssh_host_rsa_key.pub"`

2048 SHA1:ffhr0khgwGLyQb4ktmAKKPodgsA System@windows-12 (RSA)

After the host key value is identified, you can place it into the `aspera.conf` file in order to enforce the host key validation between the Aspera server and client systems. Use the following command-line routine to update the aspera.conf file:

```
asconfigurator –x "set_server_data;ssh_host_key_fingerprint,fingerprint_value"
```

Replace the `fingerprint_value` string with the actual fingerprint value returned from the *ssh-keygen* command.

NOTE: The asconfigurator command string is the same for both Windows and Linux systems

The results of executing this asconfigurator command update the aspera.conf file to include a line similar to the following entry in the <server> section of the file:

```
<ssh_host_key_fingerprint>31:40:42:d1:08:23:26:e9:0a:dd:13:57:0a:91:82:c2
</ssh_host_key_fingerprint>
```

Restart the Aspera NodeD Service

You must restart the Aspera node service for your changes to take effect:

Linux: `/etc/init.d/asperanoded restart`

Windows: `Control Panel > Administrative Tools > Computer Management > Services and Applications > Services >Aspera > NodeD >Restart`

IBM Training

Verify transfer connection

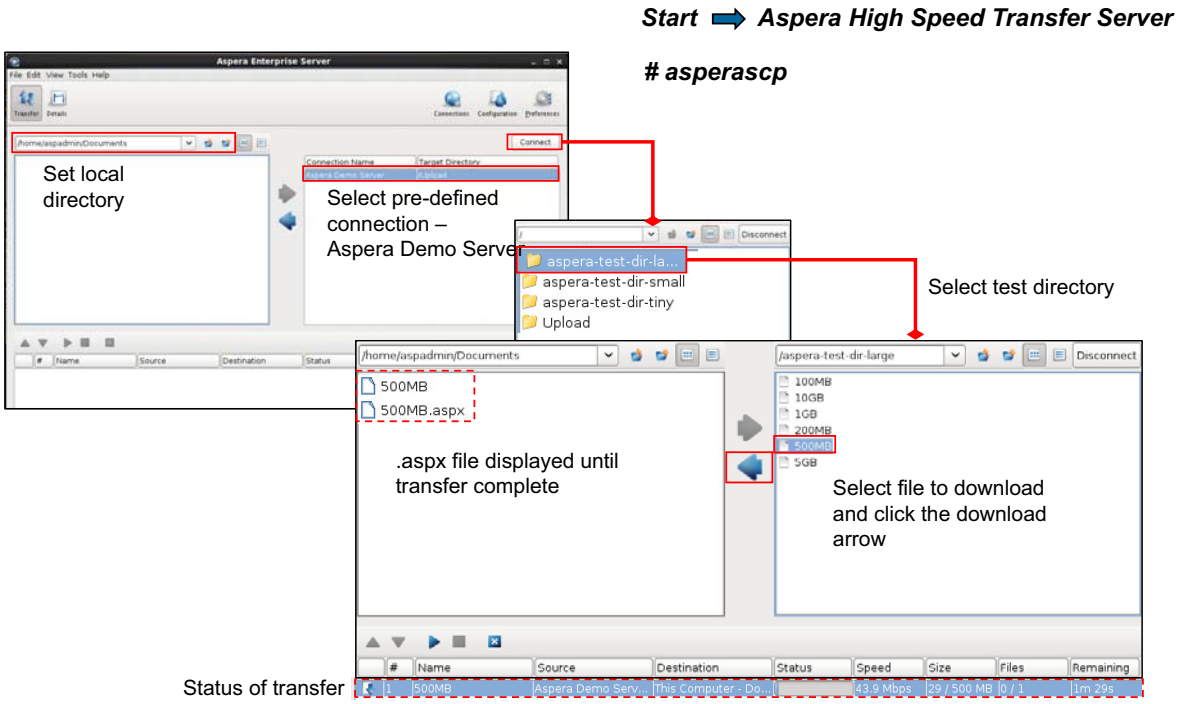


Figure 3-12. Verify transfer connection

After the initial installation and configuration is finished, you can test the system's ability to upload or download files via FASP.

Aspera GUI Transfer

The Aspera GUI provides a graphic interface for transferring files to and from the server. Start the GUI as follows:

Windows: `Start > All Programs > Aspera >Enterprise Server > Enterprise Server`

Linux: asperascp

The GUI opens a window that shows the contents of the local system directory on the left side. You can change the local directory with the menu box, the Up One Level icon, or create a new directory with the Create New Folder icon. The right side shows a pre-defined connection to the Aspera Demo Server.

Click the Connect button to open a connection.

Once the connection is established, the right window will show a list of directories on the Aspera demo server. Clicking one of the directories shows the contents of the directory.

Select a file on the Aspera demo server, then click the Download arrow in the middle of the page.

The example shown selected the file named 500MB from the aspera-test-dir-large directory. Notice the transfer status information provided at the bottom of the screen. This Transfer report indicates the name of the file, the name of the remote and local systems, and the status of the transfer.

The name of the download file appears in the local system window. Additionally, you might see the same file name, but with the `.aspx` extension. This file contains data that FASP needs to resume the transfer if it is interrupted. Once the file transfer is finished, this file is removed.

NOTE: If you selected a small file to download, the transfer might complete before the .aspx file is displayed.

Command Line Transfer

You can also test your system's ability to perform transfers from the command line by using the ascp command:

```
ascp  -P 33001 aspera@demo.asperasoft.com:aspera-test-dir-large/500MB /tmp
```

When prompted for a password, enter demoaspera

NOTE: The Aspera Demo Server is configured to accept SSH connections on TCP/33001 and not on TCP/22. By default, SSH ascp sends its SSH request on TCP/22. So, you must include the –P 33001 option as an argument to ascp.

Transfer Does Not Work

If you cannot download or upload files with the connection to the Aspera Demo Server, check the status of the firewall and network configuration. The *IBM Aspera High-Speed Transfer Server Administration Guide* also includes a section on troubleshooting connection issues.

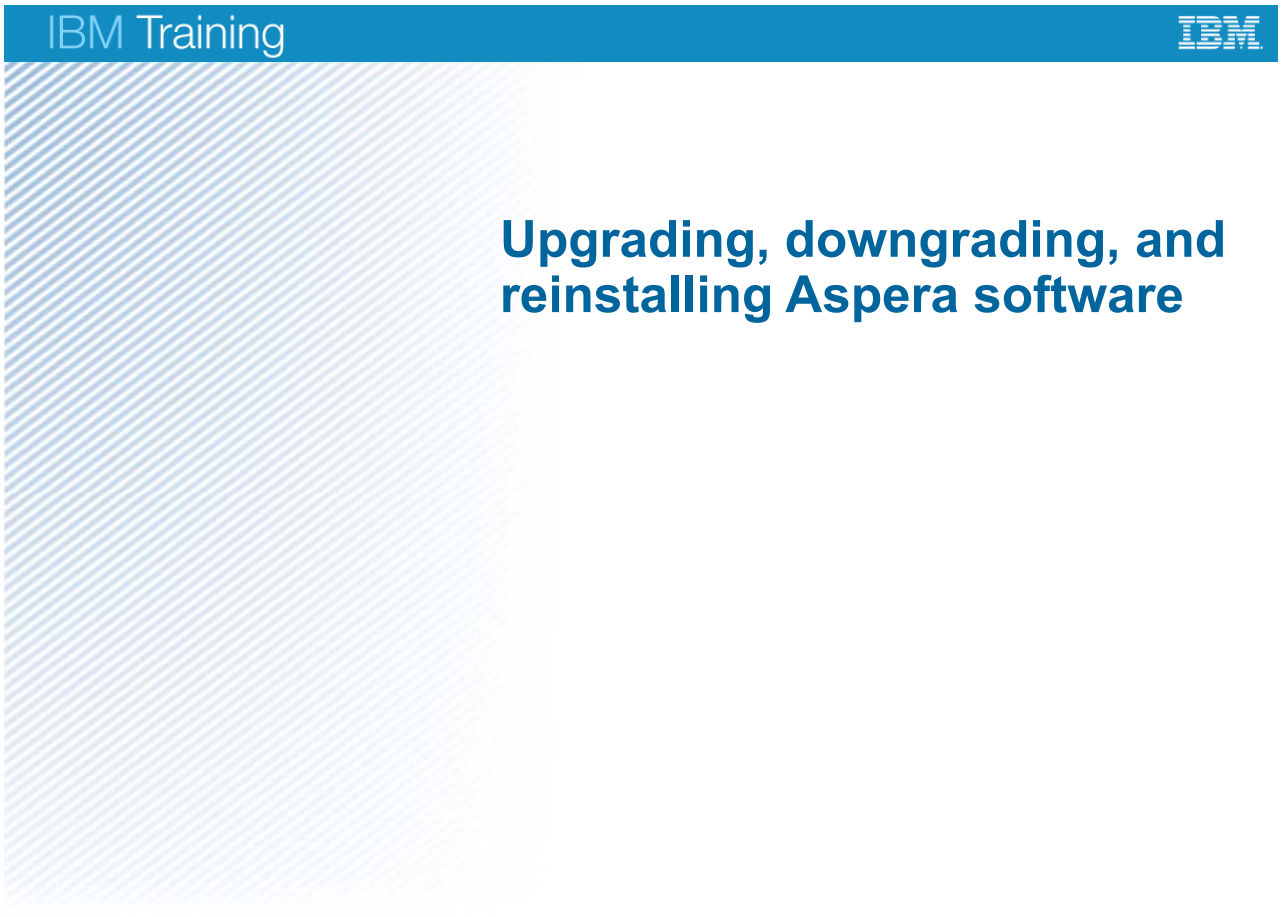


Figure 3-13. Upgrading, downgrading, and reinstalling Aspera software

IBM Training

Upgrading or Downgrading Aspera software: Overview

General considerations:

- No direct upgrade between Aspera transfer products
- Back up configuration, uninstall product, perform fresh installation
- Security level settings not preserved when upgrading versions older than 3.2.1
- Older installers do not check for newer installations
- Newer versions of Redis database not compatible with older versions

Preparation:

- Verify version of software (use ***ascp –A***)
- Upgrade process dependent upon software version
- If using Watch Folders, prepare WatchFolders for upgrade
- Back up configuration and settings files

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

- Back up Redit database

```
/opt/aspera/bin/asnodeadmin -b /filepath/database.backup
```

See “Installation and Upgrades” section of IBM Aspera High-Speed Transfer Server Administration Guide for details.

Figure 3-14. Upgrading or Downgrading Aspera software: Overview

Upgrading

The Aspera HST Server installer automatically checks for an older version of the product on your system. If an older version is found, the installer automatically removes it before installing the new version.

You cannot upgrade directly between different Aspera transfer products (such as from HST Endpoint or Desktop Client to HST Server). To upgrade, you need to back up the configuration, uninstall the product, and perform a fresh installation of the new version of the product.

When upgrading from Connect Server versions older than 3.2.1, the Connect Server system-level security settings are not preserved and must be reconfigured.

Downgrading

Older installers do not check for newer versions of the application. You must uninstall the newer version before continuing with your downgrade.

Newer versions of the Redis database are not compatible with older versions of the application. Your downgrade process depends on whether a backup of the older Redis DB is available, either as a separate backup file or as part of your backup of the var directory from the older version. See the Before Upgrading or Downgrading section of the *IBM Aspera High-Speed Transfer Server Administration Guide* for details about the Redis database.

Preparing for an Upgrade or Download

The steps that are required to prepare for an upgrade depend on your version. To view the current product and version, click `Tools > License` in the GUI or run the ascp `–A` command.

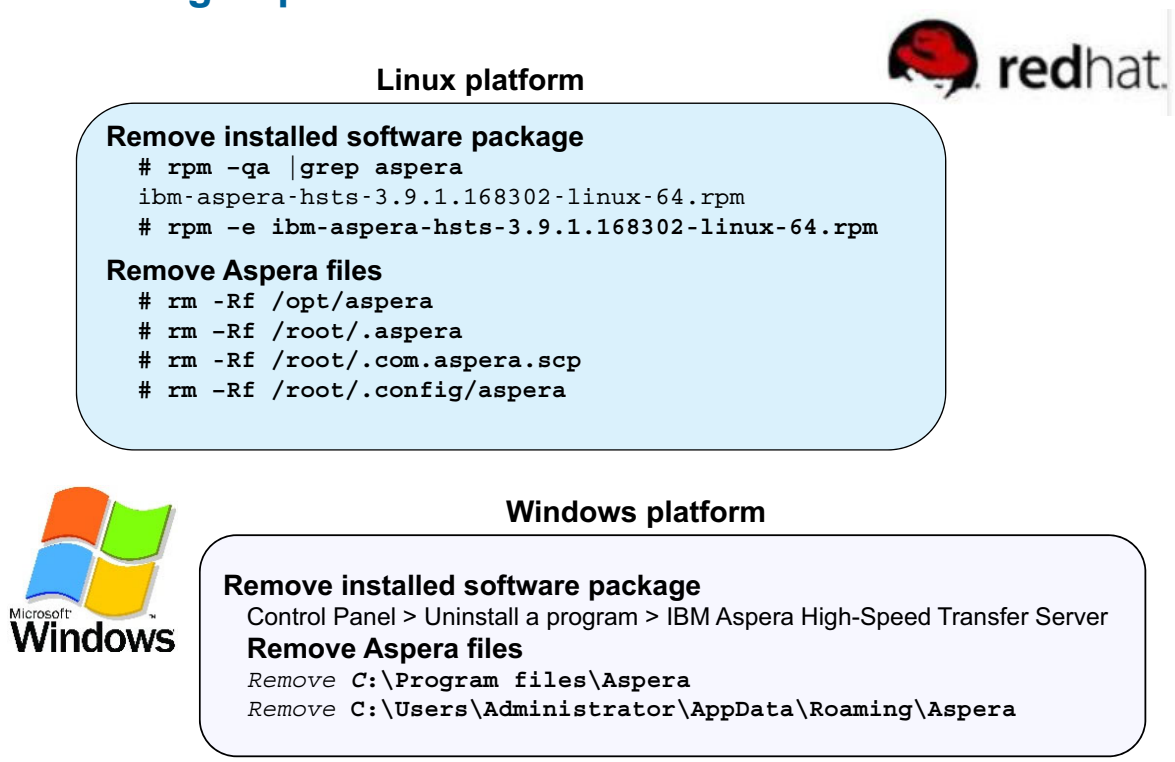
To prepare existing Watch Folders for upgrade, archive files in the source directory and change the configuration of existing Watch Folders to never overwrite.

Back up configuration and settings files: `/opt/aspera/etc` (contains server configuration, web configuration, user settings, and license information) and /opt/aspera/var (contains Pre/post processing scripts and web User Interface settings).

Back up the Redis database by using the asnodeadmin `–b` command.

IBM Training

Removing Aspera software



Installing IBM Aspera High-Speed Transfer Server © Copyright IBM Corporation 2020

Figure 3-15. Removing Aspera software

Upgrading the IBM Aspera software is a simple as running the Aspera installer routine, which retains the Aspera configuration and only updates those sections of software that need to be replaced. However, if your existing configuration contains errors or settings that you do not want to retain, you must perform some additional tasks to remove all Aspera configuration files and associated configuration files.

Linux platforms

Removing the software is the first step to implement a fresh installation. Use the standard Linux software installer routine on your system to remove the Aspera software. The example shows the `rpm –qa | grep` command to identify the specific package installed and the `rpm –e` command to remove the installed package.

After the software package is removed, remove all the Aspera files by deleting the `/opt/aspera` directory with the `rm –RF /opt/aspera` command.

Some additional files that are stored in the home directory of the account that was used for the installation. If these files are not removed, their content causes the new installation to retain the settings that were configured for that user, which corrupt the new installation. The example indicates that the software was installed using the root account, thus the files to be deleted are located in the `/root` directory:

- `.aspera.com`

- `.aspera.scp`
- `.config/aspera`

After these files are removed, a fresh installation with no remnants from the previous installation is completed.

Windows platforms

Windows systems use the Control Panel to remove the Aspera software as follows:

`Control Panel > Uninstall a program > IBM Aspera High-Speed Transfer Server`

After the software is removed, remove the Aspera files by deleting the `C:\Program Files\Aspera` directory**.**

Finally, as with a Linux installation, some additional configuration details are stored in the installer's directory. The example shows that the Administrator account performed the installation. Consequently, the `C:\Users\Administrator\AppData\Roaming\Aspera` directory must be removed to ensure a fresh installation.

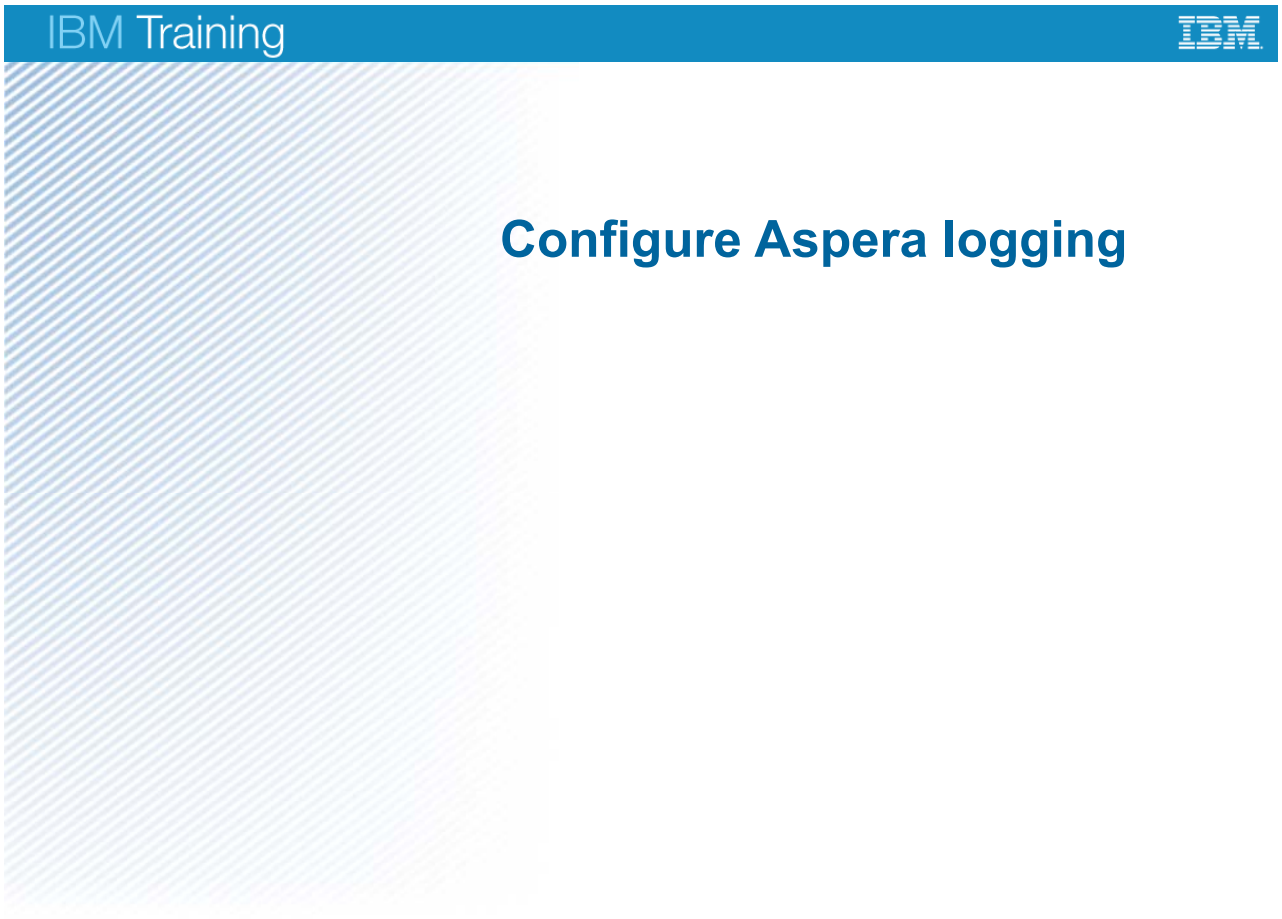


Figure 3-16. Configure Aspera logging

IBM Training

Redirect Aspera logging to *aspera.log*

Modify /etc/rsyslog.conf

Add following lines AFTER "$ModLoad imuxsock" entry

```
# Disable rate-limiting of log entries
$SystemLogRateLimitInterval 0
    $SystemLogRateLimitBurst 0
```

Not required, but recommended

Add "local2.none" parameter to "*.info;mail.none;authpriv.none;cron.none;local2.none"

Add entry "local2.info -/var/log/aspera.log"

Use TAB – NOT spaces

Allows asynchronous logging

Restart log service

service rsyslog restart OR ***/etc/init.d/rsyslog restart***

Configure log rotation

Figure 3-17. Redirect Aspera logging to aspera.log

It is recommended that you redirect your Aspera logging to a file confined to Aspera log information. By default, all Aspera log entries on Linux systems are written to `/var/log/messages`. You can redirect log entries to a different file (like `/var/log/aspera.log`) by making modifying the log configuration file.

Modify `/etc/rsyslog.conf` **or** `/etc/syslog.conf`

NOTE: Redhat/CentOS and other distributions usem `rsyslog` instead of *syslog*. In `rsyslog`, rate-limiting of the log messages is enabled. Suse (SLES) systems use `syslog-ng` (not addressed here – see the *IBM Aspera HSTS Administration Guide* for configuring Suse systems).

By default, rate limiting is activated if a process sends more than 200 messages in 5 seconds. While this limiting action is not common, it can cause problems for some Aspera transfers when many log entries are sent within a short time. Rate limiting can cause a loss of some Aspera log entries, which can result in incomplete log information that is required for analyzing performance and transfer problems. You can disable rate limiting modifying the `/etc/rsyslog.conf` file.

Add the following lines AFTER the $ModLoad imuxsock entry to disable rate-limiting by `rsyslog`:

```
$SystemLogRateLimitInterval 0
$SystemLogRateLimitBurst 0
```

The SystemLogRateLimitInterval parameter determines the amount of time that is being measured for rate limiting. The SystemLogRateLimitBurst defines the number of messages that must occur in the time limit defined by the SystemLogRateLimitInterval to trigger rate limiting. A value of zero disables the rate limiting function.

Enable separate logging for Aspera log entries and specify the location of the log file

Add the text local2.none to the end of the line containing the entry

```
*.info;mail.none;authpriv.none;cron.none
```

Associate the local2 parameter with the output file for redirected content.

NOTE: A hyphen is needed before the log file name, which allows for asynchronous logging. Also, the log file name should be separated from the log facility (`local2.info`) by tabs, not spaces.

Restart log service

After making these changes, you need to restart `rsyslogd` (or `syslogd`) by executing either the `service rsyslog restart` or `service syslog restart` command.

Configure log rotation

While not a required step, it is a good idea to configure log file rotation to include the separated Aspera log file. The following page discusses the procedure for configuring log file rotation.

IBM Training

Log file rotation

Option 1 - Add entry in **/etc/logrotate.d/syslog**

```
/var/log/messages /var/log/secure /var/log/maillog /var/log/spooler /var/ log/boot.log /var/log/cron /var/log/aspera.log
/var/log/aspera.log {
sharedscripts
postrotate
        /bin/kill –HUP ‘cat /var/run/syslogd.pid’ 2> /dev/null || true
        /bin/-HUP ‘cat /var/run/rsyslogd.pid’ 2> /dev/null ||  true
endscript
}
```

Option 2 – Edit /**etc/logrotate.conf**

```
/var/log/aspera.log {
rotate 10
size 100M
create 664 root
postrotate
/usr/bin/killall –HUP syslogd
endscript
compress
}
```

Option 3 - Create a separate **/etc/logrotate.d/aspera** configuration file

```
/var/log/aspera.log {
daily                    Rotates daily, regardless of file size
rotate 10
copytruncate             Copies and truncates original log file instead of creating new
compress
}
```


Figure 3-18. Log file rotation

Aspera recommends configuring log file rotation in order to ensure that log data is not overwritten unexpectedly. Different methods for configuring log rotation are available. The method that you use is dependent upon how much traffic the server manages.

Compressing and saving the log file once a week is a common setting and is achieved by using **Option 1** as shown.

- Option 1: Add the `/var/log.aspera.log` entry to the `/etc/logrotate.d/syslog` file.
- Option 2**:** Edit `/etc/logrotate.conf` by adding configuration information after the line # *system-specific logs may also be configured here*. The example compresses and rotates 10 log files whenever the `/var/log/aspera.log` file nears 100MB.
- Option 3: Create a separate `/etc/logrotate.d/aspera` configuration file containing similar entries as Option 2.

What you learned

- An appropriate login and password required to download all IBM Aspera software, as well as a valid license key
- Aspera services account is used by Windows to manage Aspera-enabled processes, including OpenSSH
- Configure *SSH* to use a non-standard TCP port, and to define the authentication method (login credentials, public and private key) by modifying the ***sshd_config*** file
- Configure the firewall to allow inbound connections from TCP and UDP ports 33001
- Windows installations can also require a range of both inbound and outbound UDP ports allowed by the firewall to support multiple concurrent transfer client connections
- Default configuration of the Transfer Server software provides a connection to an Aspera demonstration server that can be used to verify the ability to transfer file to your system
- The Transfer Server application provides a graphic interface for configuring the server, as well as performing transfers
- File transfers may also be performed using the ***ascp*** command at the command line

Figure 3-19. What you learned

IBM Training

Unit summary

- Identify the prerequisites for a successful deployment of IBM Aspera High-Speed Transfer Server
- Configure the system firewall to support Aspera transfers
- Secure access to Aspera services by modifying the SSH configuration
- Locate and install the appropriate IBM Aspera High-Speed Transfer Server software (Windows and Linux)
- Explain the purpose of the Aspera service account on Windows systems running IBM Aspera High-Speed Transfer Server software
- Verify installation success by transferring files to and from the Aspera Demo Server
- Configure Aspera log redirection

Figure 3-20. Unit summary

The Aspera FASP protocol is implemented within the ascp binary routine, which is included in Aspera Transfer Server software and other Aspera client software (for example, Aspera Desktop Client and Connect browser plug-in). The ascp binary can be accessed from the command line and has an extensive set of options for managing a transfer.

Because ascp is the binary that implements the FASP protocol, all FASP transfers require an ascp process running at each end. Aspera transfers are always between ascp processes!

By default, the FASP protocol (which is implemented in the ascp binary) uses UDP port 33001, unless the server and client are configured to use a different port for FASP-based transfers.

The ascp binary is able of performing several tasks besides handling transfers. Besides managing the transfer, the ascp program is responsible for authenticating with the transfer server as part of the initiation of a transfer (details presented later in this module). Aspera web applications pass authorization tokens to a client's Connect software, which then hands the token to the ascp routine. The ascp program presents the token to the server as part of the file transfer initiation after successful authentication via SSH.

It is important to remember that the ascp binary is NOT included in Aspera web-based application software. However, the transfers that are performed through the web-based applications do so through the ascp routine on the transfer server and ascp that is implemented by the Connect plug-in on the client system.

Review questions (1 of 2)

1. Which of the following statements best represents requirements for a successful installation of IBM Aspera High-Speed Transfer Server software? Select all that apply:
   A. Firewalls should have UDP and TCP ports 33001 open
   B. SSH must be configured to support TCP port 22
   C. On Linux systems, if SELinux is “enforcing”, it must be configured to allow Aspera products to run properly
   D. At least 2 IBM Aspera High-Speed Transfer Server systems must be configured to allow high-speed transfers between users in an organization

2. True or False: Successful installation of IBM Aspera High-Speed Transfer Server software requires the host fingerprints be implemented during the installation process

Figure 3-21. Review questions (1 of 2)

IBM Training

Review questions (2 of 2)

3. True or False: All IBM Aspera server software requires a license key

4. Which of the following statements best represents the reasons for configuring Aspera logs on Linux systems? Select all that apply:
   A. Aspera logging information differs from normal logging data and requires a special log file format
   B. The amount of data logged by Aspera may be extensive and separating it into a separate log can make it easier to manage
   C. Not all Aspera transfer logging data is written to the default */var/log/messages*
   D. Aspera logging data will be stored if it is not written to a separate log file

Figure 3-22. Review questions (2 of 2)

IBM Training

Review answers (1 of 2)

1. Which of the following statements best represents requirements for a successful installation of IBM Aspera High-Speed Transfer Server software? Select all that apply:
   - A. Firewalls should have UDP and TCP ports 33001 open
   - B. SSH must be configured to support TCP port 22
   - C. On Linux systems, if SELinux is “enforcing”, it must be configured to allow Aspera products to run properly
   - D. At least 2 IBM Aspera High-Speed Transfer Server systems must be configured to allow high-speed transfers between users in an organization

   The answer is A and C

2. True or False: Successful installation of IBM Aspera High-Speed Transfer Server software requires the host fingerprints be implemented during the installation process

   The answer is False. Configuring host fingerprints is highly recommended by Aspera to combat server impersonation (man-in-the-middle)

Figure 3-23. Review answers (1 of 2)

IBM Training

Review questions (2 of 2)

3. True or False: All IBM Aspera server software requires a license key

The answer is True.

4. Which of the following statements best represents the reasons for configuring Aspera logs on Linux systems? Select all that apply:
   - A. Aspera logging information differs from normal logging data and requires a special log file format
   - B. The amount of data logged by Aspera can be extensive and separating it into a separate log may make it easier to manage
   - C. Not all Aspera transfer logging data is written to the default */var/log/messages*
   - D. Aspera logging data will be stored if it is not written to a separate log file

   The answer is B

Figure 3-24. Review questions (2 of 2)

IBM Training

Lab Exercise 1

The lab exercise associated with this section requires you to install the IBM Aspera High-Speed Transfer Server software on Windows and Linux platforms.

Once you have installed the software, you connect to the Aspera Demo Server system using the Aspera GUI to download and upload files verifying that your installation is functional.

You also learn to use the Aspera GUI to manipulate files on your local system and the remote Aspera system.

The final task is to configure a Linux system to log Aspera data into a separate log file.

- Go to lab environment.
- **READ** the **Lab Intro** module for details (explains how to use the lab environment).
- Start the servers in the lab environment.
- Follow the steps in **Exercise 1.**
- Perform tasks on the Windows server (Singapore) and **BOTH** Linux servers (Denver and London).
- If you leave the environment for more than an hour or so, the systems are suspended and you need to restart them before continuing.
- **Don’t skip the questions**. Try to answer the questions that are asked as you perform the various tasks. The questions are designed to make you think about what the system is doing and to draw attention to specific values associated with transfers.

Figure 3-25. Lab Exercise 1

Unit 4. Configuring IBM Aspera High-Speed Transfer Server

Estimated time

02:00

Overview

This unit presents the various parameters available for configuring IBM Aspera High-Speed Transfer Server software

IBM Training

Unit objectives

- Navigate the Aspera GUI to access various configuration parameters
- Identify the kinds of global parameters that may be configured
- Define maximum bandwidth and default target rates for transfers
- Manage file permissions for inbound/outbound transfers
- Define and implement Vlinks

Figure 4-1. Unit objectives

This module addresses the use of the Aspera GUI to access various configuration parameters. The focus here is to understand the various parameters and the values that can be configured. Configuration of these same parameters by using the command-line utility asconfigurator, and the associated entries in the aspera.conf file are presented in the Command-Line Operations module of this course.

IBM Training

Aspera GUI transfers

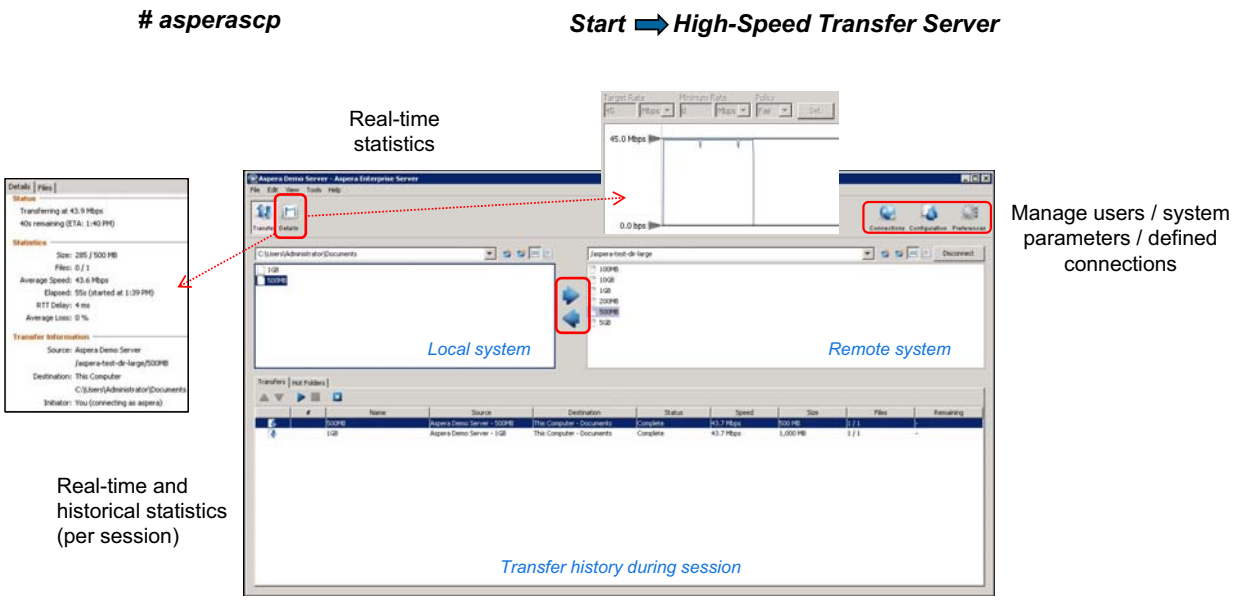


Figure 4-2. Aspera GUI transfers

Graphic User Interface

The Aspera HSTS includes an easy-to-use graphic interface that facilitates system configuration and transfers to and from the system. The GUI is presented when the Aspera Transfer Server software is started. To start the desktop application, run the asperascp command in a terminal window. To perform administrator tasks (such as server configuration, license updates, or configure email notification templates), start the GUI with root permissions.

System Management

The upper right corner of the screen provides buttons that can establish connections with other systems and enable the configuration of numerous parameters that control how transfers are managed. An administrator can use the Aspera GUI to set preferences. Preference for target transfer rates, the maximum number of simultaneous transfers, where to send system notifications, and identifying proxy servers can all be configured.

Live and Historical Statistics

The Aspera HSTS GUI interface provides both current and historical information about transfers. By default, the left side of the application indicates the contents of a directory on the local Aspera Transfer Server system, while the right side indicates the contents of the remote location. This interface provides a graphic mechanism for moving files between the two systems. However, these screens can also indicate the status of transfers in process, and provide a graphic representation of

the speed for the transfer. Users can toggle between these two views by using the Transfer and Details buttons that are on the left side of the screen.

Historical information is also provided at the bottom of the screen, indicating the transfer statistics for the objects or files that were transferred. This section of the screen also provides information about in-process transfers, but with less detail than that provided by the Details option.

IBM Training

Server GUI menus and pages

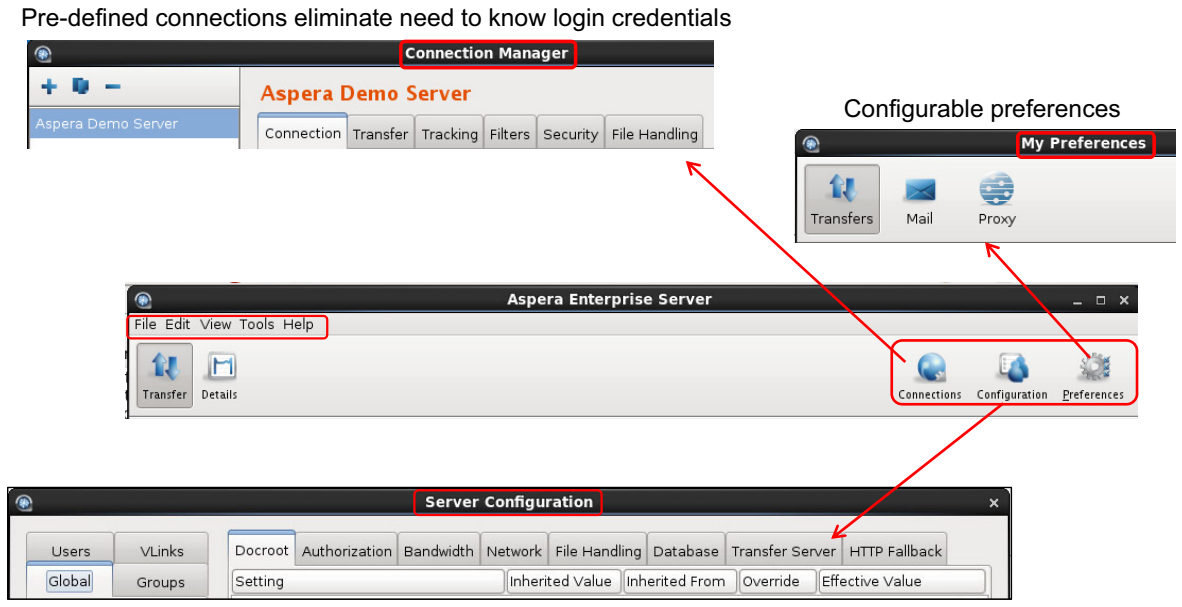


Figure 4-3. Server GUI menus and pages

In addition to providing a graphic interface for transferring data, the Aspera Transfer Server GUI also provides an easy-to-use interface for performing most of tasks necessary to manage the operations of server.

NOTE: The Configuration options that are discussed here are only displayed when you open the Aspera GUI with administrative privileges.

Five menu options appear on the left side of the opening window. These options allow management of local files and directories, selection of the information that is shown for local files, access to tools for various tasks, and viewing help information.

File: options for managing files and directories on the local Aspera HSTS system

Edit: simple editing commands, for example, copy and paste

View: selects view of files, List or Details

Tools: menu of basic system management functions

Help: displays a PDF version of the *IBM Aspera HSTS Administration Guide*, or runs a routine to identify and resolve basic configuration issues

Three buttons in the upper right corner of the opening screen of the Aspera GUI enable the administrator to configure the server:

Connections: Configures and manages connections to remote Aspera servers

Configuration: Configures server parameters that manage various aspects of the Aspera HSTS environment

Preferences: Allows the administrator the ability to define their preferences for data transfers, email notifications, and the configuration of proxy servers.

The following pages of this module are designed to present the parameters you can configure from each of these options.

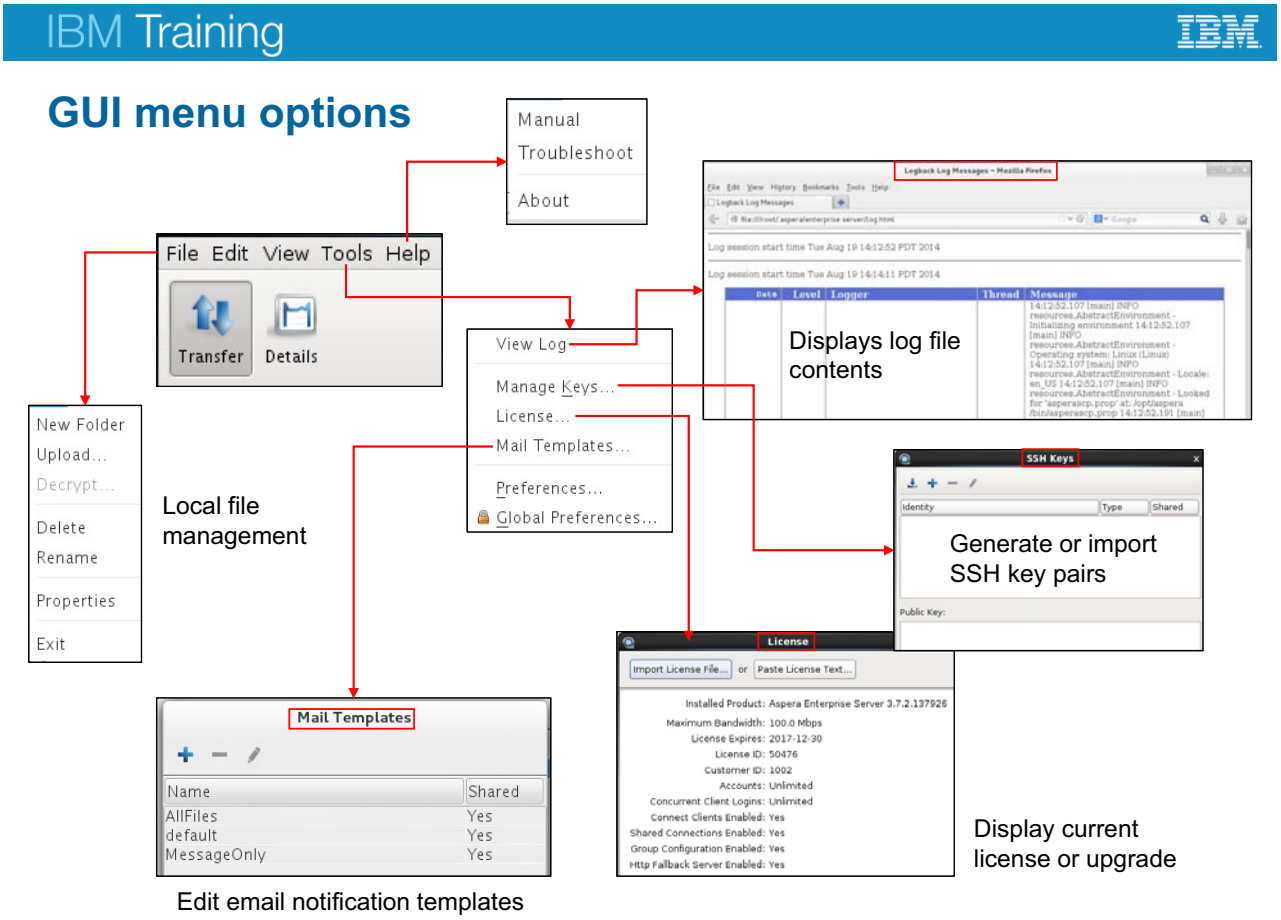


Figure 4-4. GUI menu options

The menu options that are displayed on the left side of the opening window provide access to several critical functions.

File: provides access to option that manages local files and directories. You can create new folders, upload files, decrypt encrypted files, delete, and rename local files.

Edit: provides options to manipulate selected files.

View: selects the information that is provided about files (list of names or details of file).

Tools: opens a menu where you can select tools that can help managing the Aspera HSTS environment:

View Log: opens a log of Aspera HSTS management events that occurred on the server

Open Log Folder: (only on Windows server) opens a directory where individual log files are stored for other applications that are components of the Aspera HSTS environment

Manage Keys: provides a graphic interface for creating or importing public and private key data.

License: opens a window where you can import or paste new license keys. This screen is the same as you encountered during the initial software installation, but is used again if you upgrade your Aspera HSTS to increase the transfer rate license. Any new license that is

installed from this window updates the Aspera HSTS to recognize the values identified in the new license key. The changes in licensing take effect after restarting the Aspera GUI.

Mail Templates: brings up a menu from which you can edit mail templates that are used for notifications. See the Aspera HSTS Administration Guide for details about configuring email notifications for both the Aspera HSTS global settings and individual user configuration.

Preferences and Global Preferences: opens a window with links to pages that define configuration of transfer speeds, email servers (for email notifications), and proxy services. Users can also specify their own preferences from the “Preferences” button at the right side of the GUI.

NOTE: Only users with administrative privileges can set Global preferences.

Help: opens a pdf version of the *IBM Aspera HSTS Admin Guide* or starts a tool that identifies basic configuration problems or issues, along with possible solutions.

IBM Training

GUI access to Aspera server logs

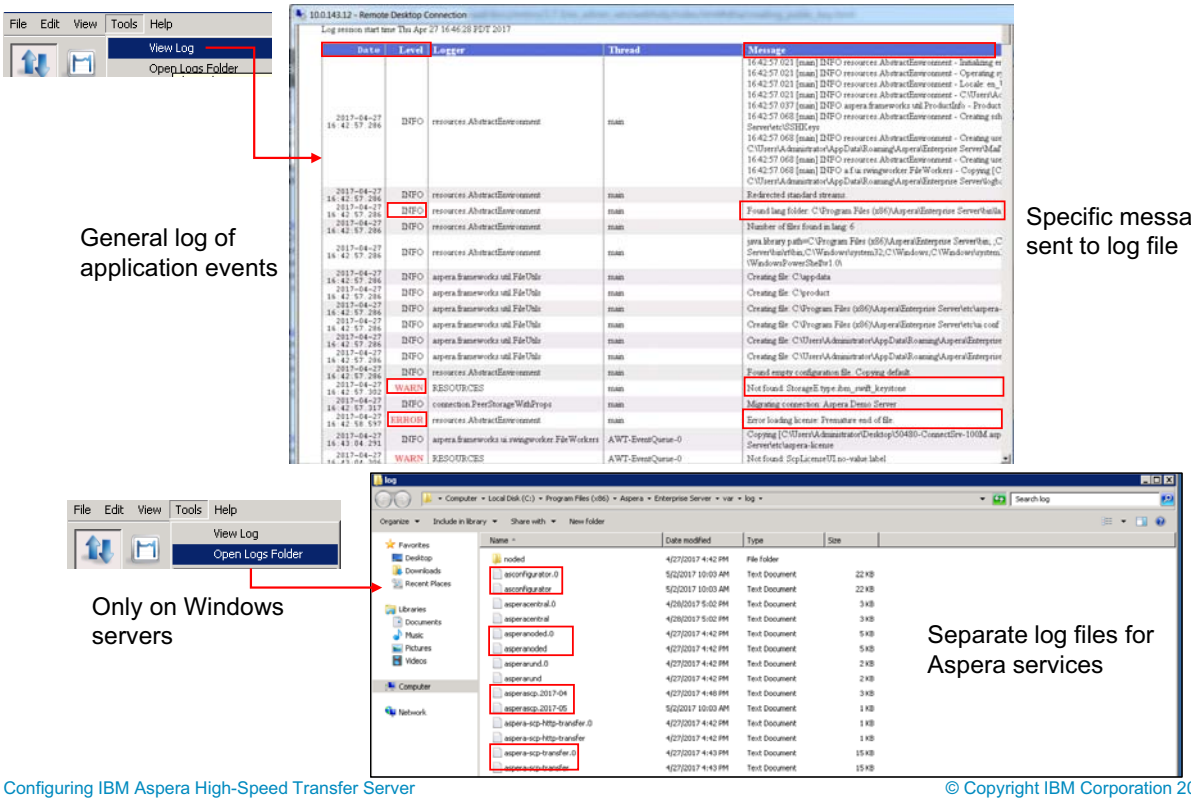


Figure 4-5. GUI access to Aspera server logs

The Aspera HSTS maintains extensive historical information through its logging function. The log files include detailed information about events that are associated with every Aspera HSTS service and component, and can be useful for review and support requests.

You can view log files by using the Tools menu. Depending upon what operating environment you are using, the Aspera HSTS application provides slightly different options for log files. On Linux systems, the only option available for logs is the View Log option. If you are running Aspera HSTS in a Windows environment, you see two options that are associated with log files View Log and Open Logs Folder.

The View Log option opens a window that displays general logging information about tasks that the Aspera HSTS application performed. The location of the actual log file is operating system-dependent, for example, C:\Program Files (x86)\Aspera\Enterprise Server\var\log on Windows systems and /var/log/messages on Linux systems.

While viewed from within the Aspera GUI, the log file is named log.html, and is located in the /root/.aspera/enterprise server directory in Linux environments or C:\Users\Administrator\AppData\Roaming\Aspera\Enterprise Server directory on Windows systems. The contents of these log files that are displayed from within the Aspera application extract relevant information from other files.

Output of all Aspera utilities and services in Linux environments are stored in a single file by default – /var/log/messages. You can extract the specific log information that you want by piping the file contents to grep. You can also redirect Aspera HST logging data to a different file (for example, /var/log/aspera.log) by modifying the /etc/rsyslog.conf or /etc/syslog.conf file. The procedure for creating separating Aspera log entries is discussed in another module of this training.

Entries in the general log file show date and time, and can be identified with any one of three levels of severity. These levels are INFO (provides neutral information, WARN (indicates a non-fatal event), and ERROR (an unrecoverable event that impacts the ability to run the application). The file also indicates what routine generated the entry for the log file (Thread), and the Messages field identifies the message that was sent to the file from the Aspera service.

For example, the highlighted INFO entry provides basic information that is recorded as the application started, for example, a message that the system found the language folder.

The ERROR entry indicates an unrecoverable problem when starting, such as experiencing a problem when loading the license file and not being able to continue.

The information that is provided by the general log file can seem cryptic at times, but the data can give you insight when attempting to resolve a problem in Aspera HSTS operation.

The Windows version of the Aspera HSTS application provides another option under the Tools menu > the Open Logs Folder option. Selecting this option opens a window that shows multiple log files, each associated with a specific service or component of Aspera HSTS. Depending upon how the system was implemented, various utilities and services, such as asconfigurator, aperanoded, asperacentral, ascp, and aspera-scp-transfer can have their own log file. The name of the log file indicates the service or component that is generating the data that is written to the file. The asconfigurator routine is called by the GUI when configuration changes are made, and is responsible for updating the aspera.conf file with those changes.

asconfigurator.log: stores event information about the configuration of the Aspera HSTS application

asperanoded.log: stores log information about events that are associated with Aspera Node API functions

asperacentral.log: stores events associated with file transfers between servers

aspera-scp-transfer.log: record of all FASP-based transfer statistics and events

On Windows systems, Aspera logs are automatically rotated based on the log size. After a log file reaches 10 MB, the system begins logging to a new file with a .1.log suffix and continue until .9.log is created. The maximum number of log files created this way is 10, and if the .9.log file reaches 10 MB, then older files are overwritten. The log file with no number added is the current working log and is identical to the last numbered log. You can want to archive these log files for security or analysis purposes. The Aspera Support Knowledgebase article *How to Archive Aspera Logs on Windows* provides the details about this task.

IBM Training

Creating GUI connections

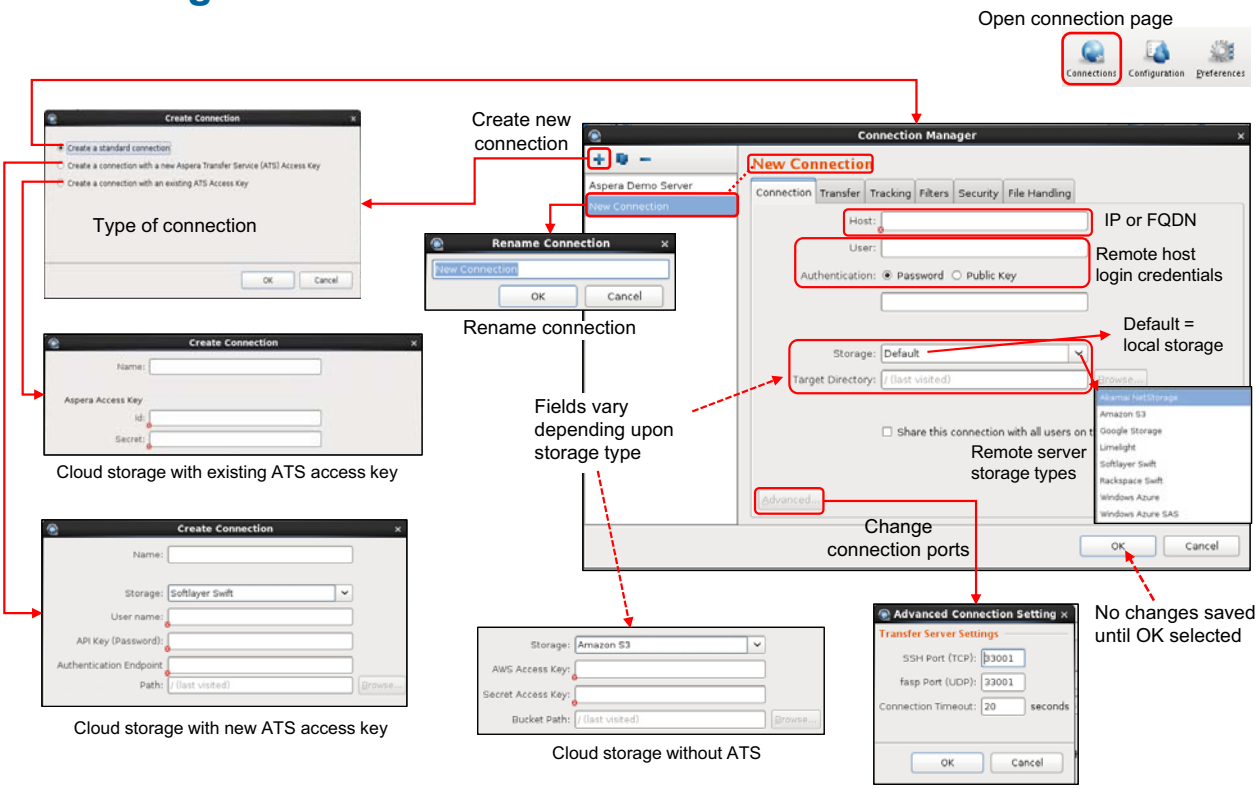


Figure 4-6. Creating GUI connections

The IBM Aspera High-Speed Transfer Server software provides a graphic interface for transferring data as a client with other Aspera Transfer Server systems. You can configure details about connecting with other Aspera servers by clicking the Connections button of the Aspera GUI. From here, you can identify the connection parameters and how to deal with files and events that are encountered during transfer interactions with the other Aspera server.

If connections to other servers are defined, you can access the details about the connection by clicking the name that is shown in the left pane of the interface.

You can create new connections by clicking the + symbol in the Connection Manager window to open a menu that indicates the type of connection you want to create:

Create a standard connection – Use this option to connect with all connections that do not use the Aspera Transfer Service to connect to cloud storage.

Create a connection with a new Aspera Transfer Service (ATS) Access Key – Use this option to create a connection to cloud storage and generate a new ATS access key to authenticate the connection.

Create a connection with an existing ATS Access Key – Use this option to create a connection to cloud storage with an existing Aspera access key.

You can also delete connections by highlighting the connection name and clicking the – symbol. You can also duplicate an existing connection's values by clicking the duplication icon that is found between the + and – symbols.

Create a Standard Connection

Notice that the name New Connection is displayed and highlighted on the left side of the window. You can assign a name to this connection by clicking the New Connection name. You can use the entry in the listing of connections at the left of the window. Or you can click the New Connection title at the top of the window to open another window where you can input whatever name you want for this connection.

Six tabs across the top of the connection window can be used to configure details about the connection. How transfers are handled when connecting with this server, whether encryption is used by those transfers, and how files are verified and written when transferred can all be configured.

Connection

When first opened, the Connection tab is displayed which provides the following fields:

Host: The name of the remote server to which a connection is made (typically a full-qualified domain name for the remote server).

User: The account that should be used to authenticate with the remote server.

Authentication: Select the type of authentication used to access this server. If a password is selected, input the password in the field provided. If you are using public and private key authentication, identify the key to use from the menu. If the key you want to use is not displayed, you can click the Manage Keys button to import an existing public key or create a new key pair.

Storage Type: The menu for this field defines the type of file system that the remote server uses. The Default choice represents local storage on the remote server and requires you identify the Target Directory (the starting directory for the connection – must be under the docroot directory) on that server.

Other storage options are available:

Akamai NetStorage

Amazon S3

Google Storage

Limelight

IBM Cloud Swift

Rackspace Swift

Windows Azure

Windows Azure SAS

Selecting anything but Default changes the fields that are displayed to reflect the required values for each storage type.

For more information, see the *Using the Connection Manager* section of the *Aspera HSTS Admin Guide*.

Required fields are indicated with a small x next to each field, and are highlighted in a different color

You choose special storage if you have FULL ACCESS to that storage on the cloud-based system.

NOTE: Connecting to object storage requires an Aspera License Entitlement Engine (ALEE)-enabled Aspera on the remote storage

Target Directory: If you select the Default store type, you need to specify the storage directory to use on the remote server. By default, this value indicates the root directory on the remote server, which represents the document root for the transfer user that is configured for the connection.

If you selected a cloud-based storage option, you are prompted for the required information that needed by the cloud provider you selected.

You can select the SSH and FASP ports to connect with the remote system by clicking the Advanced button at the bottom of the window. You can also specify the Connection Timeout value when the remote server doesn't respond in a timely manner.

By default, the new connection is only available to the administrative user who creates the connection. If you want other users of the system to be able to use this connection, put a mark in the checkbox that is associated with Share this connection with all users on this computer.

After you save the primary connection information, you can verify the connection by clicking Test Connection.

By default, pre-defined connections are limited to the user who created them (defining connections does not require administrative privileges). You can change this behavior by placing a mark in the Share this connection with all users on the computer checkbox.

NOTE: Changes that are made in any or all tabs are not saved until you click OK.

Selecting Cancel discards any unsaved changes that are made in the Connection Manager, including the addition and removal of connections.

IBM Training

Connections GUI – transfer parameters

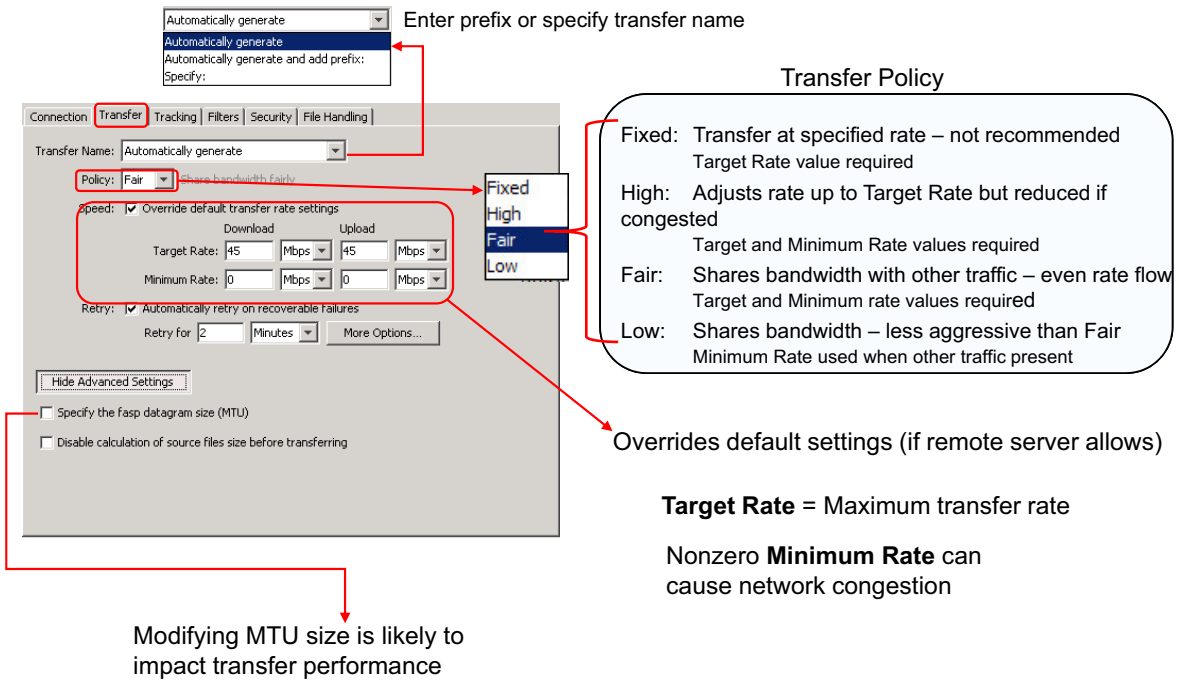


Figure 4-7. Connections GUI – transfer parameters

Parameters that define transfer behaviors for this connection between your Aspera software and the remote Aspera server are defined in the “Transfer” window. The following parameters can be set from this window:

Transfer Name

This field indicates how the system should name files that are transferred to or from this server. Files can be automatically named, can include a specified prefix value, or can be uniquely named. If you select either the Automatically generate and add prefix: or Specify: option, another field is displayed where you can input the value that you want to use. The default value is Transfer of root or Transfer of Administrator, depending upon the operating system on the server.

Policy: The transfer policy and speed determine how the network resources are used for *FASP* file transfers. Use the menu to select the transfer policy you want to apply for this connection:

Fixed: *FASP* 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 predictable time. This policy typically occupies most of the network bandwidth, and is not suggested in most file transfer scenarios. In this mode, a maximum (target) rate value is required.

High: *FASP* monitors the network and adjusts the transfer rate to fully use the available bandwidth up to the maximum rate. When congestion occurs, a *FASP* session with high policy

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.

Fair: *FASP* monitors the network and adjusts the transfer rate to fully utilize the available bandwidth up to the maximum rate. When other types of traffic increases and congestion occurs, *FASP* shares bandwidth fairly by transferring at a steady rate In this mode, both the maximum (target) and the minimum transfer rates are required.

Low: Similar to Fair mode, the Low (or Trickle) policy uses the available bandwidth up to the maximum rate, but is much less aggressive when sharing bandwidth with other network traffic. When congestion increases, the transfer rate is decreased, potentially all the way down to the minimum rate until other traffic decreases.

Speed: This checkbox provides a means of setting specific Target and Minimum Rate values for downloads from and uploads to this remote server, rather than using the default Target and Minimum Rates defined for your system.

NOTE: A more detailed discussion of server bandwidth settings is presented later in this module

The optional rate fields are displayed when the Override default transfer rate settings checkbox is selected. The Target Rate value indicates the maximum rate that is attempted when transferring files to or from the remote server. The target rate value also defines the initial rate that is attempted for transfers. The target rate can be dynamically adjusted down throughout the life of a transfer, depending upon the Policy value you select, but cannot be exceeded at any time during a transfer. The Target Rate value should not exceed the maximum physical capacity of the network, nor should it exceed the maximum rate that is designated by the Aspera license key. When determining the Target Rate value, consider the receiving system’s capacity to write data to it’s storage device.

The Minimum Rate value references the absolute minimum rate that is allowed for transfers. The default Minimum Rate value is set to 0 Mbps, but can be set higher. However, setting the Minimum Rate too high can cause network congestion for other traffic. Aspera recommends leaving the Minimum Rate value set to zero.

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 can set the initial and maximum retry intervals by clicking the More Options button.

Initial interval: The first retry waits for the initial interval. You can use the menu to specify the value 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. You can also specify this time in seconds, minutes, or hours.

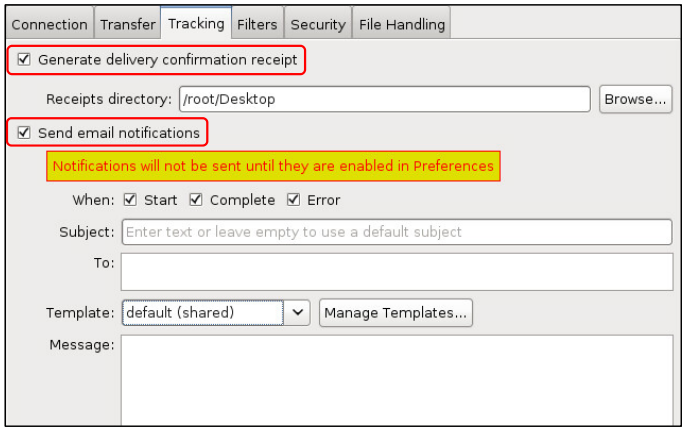
NOTE: Unrecoverable transfer failures include significant events such as server or network failure, not failures due to network congestion.

Advanced Settings can be used to set the Maximum Transmission Unit (MTU) size for FASP transfers (296 - 10000 bytes). The Advanced Settings can also disable the client-side determination of file size before transferring.

NOTE: Modifying the MTU size can impact transfer performance.

IBM Training

Connections GUI: tracking parameters



Identify location for storing delivery confirmations

Specify trigger event and email data to send email notifications

Email notification MUST be enabled in “Preferences” settings

Enable Global email notifications with “Global Preferences” button

Figure 4-8. Connections GUI: tracking parameters

The Tracking tab opens a window that provides options for tracking FASP transfer sessions, including the delivery confirmation receipt and the email notifications that are sent.

If the Generate delivery confirmation receipt checkbox is marked, the Receipts directory field is displayed, prompting for a location to store delivery confirmations of transfers.

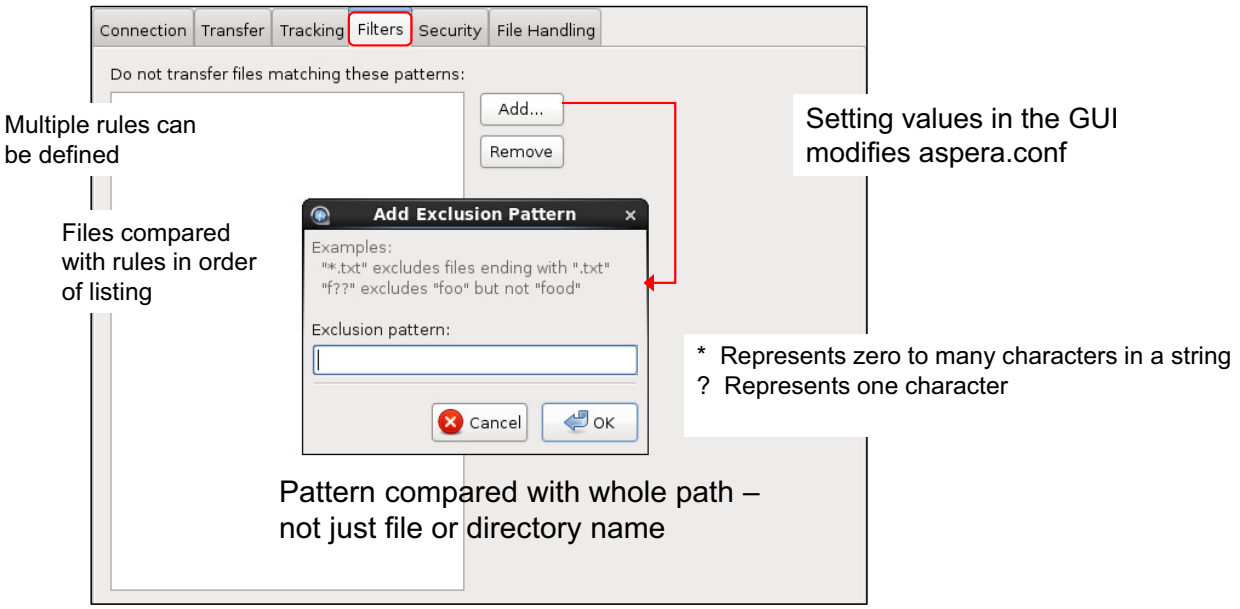
The systems can send email notifications based on specified events, for example, start of transfer, completion of transfer, and error during transfer. The email message uses an appropriate email template, which you can customize as needed. When you select the Send email notifications option, additional information must be provided.

NOTE: To receive email notifications of transfer events, you must enable this feature in the Mail preferences settings. You must also provide appropriate credentials for sending emails. Email preferences can optionally be configured as a global setting.

IBM Training

Connections GUI: filters

Filters apply only when the server is acting as a client



More details in “Applying Filters to Include and Exclude Files” section of IBM Aspera High-Speed Transfer Server Administration Guide

Figure 4-9. Connections GUI: filters

Filters refine the list of files (or directories) designated for transfer. You can indicate which files in the transfer list to skip or include.

At runtime, ascp looks for filters in two locations: on the ascp command line, and in aspera.conf.

Filters can be set in the aspera.conf file either from the GUI, or by modifying it directly with an editor or asconfigurator.

When filtering rules are found in aspera.conf, they are applied before rules on the command line.

If no filtering rules are specified, ascp transfers all source files in the transfer list.

Multiple filtering rules can be defined, and if a file does not match on the first rule, it is then compared with the second rule, and so forth. If a file does not match any of the defined rules, it is included in the transfer.

For more information, see the *Applying Filters to Include and Exclude Files* section of the *IBM Aspera HSTS Administration Guide*.

IBM Training

Connections GUI: security

All transfers use SSL by default (Encryption in transit and at rest add extra encryption)

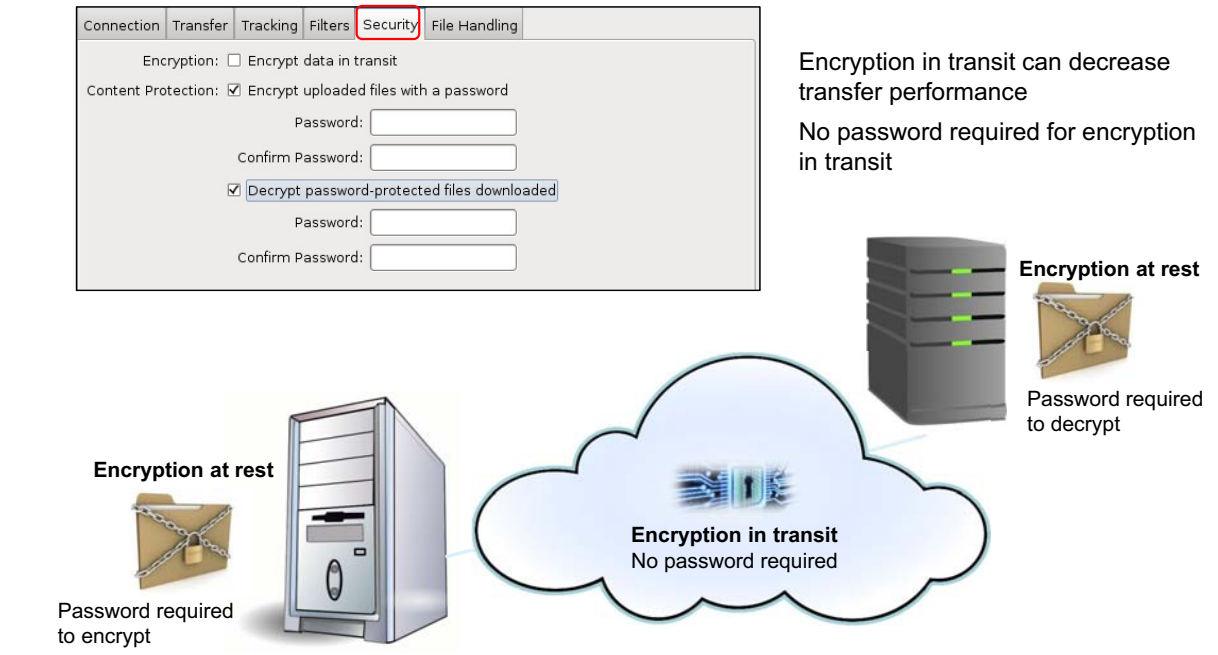


Figure 4-10. Connections GUI: security

The Security tab that is associated with a connection can be used to enable transfer and at rest encryption.

While FASP transfers use SSH encryption, choosing to encrypt data in transit provides another layer of encryption.

SSH. Encrypting data in transit does not require an outside password, as FASP is responsible for encryption in transit. However, encryption at rest (content protection) does require you to specify a password that is used to encrypt and decrypt the transferred file.

NOTE: Encrypting files in transit decreases transfer performance, especially at higher transfer speeds or with slower computers.

Notice that you can choose to encrypt uploaded files with a password, which must be used to decrypt the file at the transfer destination. Correspondingly, if downloaded files are encrypted, you must configure the password that is used by the sending system to decrypt them on the receiving system.

NOTE: You can configure encryption for either uploads or downloads, or both, depending upon your security needs

IBM Training

Connections GUI: File Handling parameters

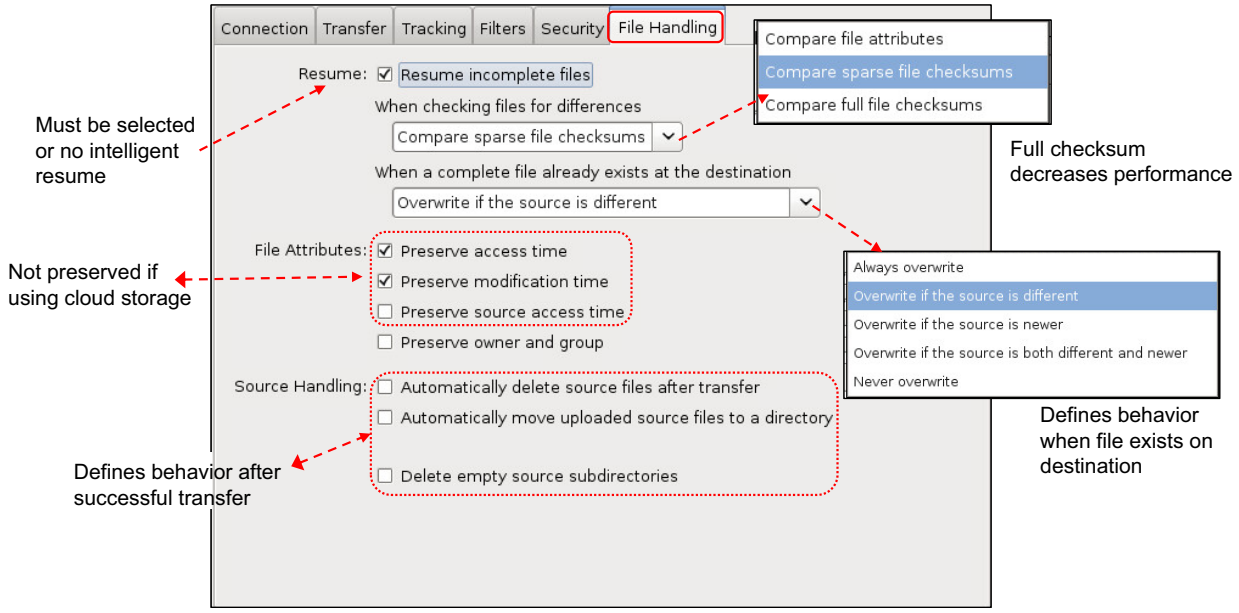


Figure 4-11. Connections GUI: File Handling parameters

The File Handling tab of **Connection Manager** defines how the server should manage transfer files.

Resume

The Resume section of the File Handling tab enables the resume feature, when checked. Marking the Resume incomplete files checkbox activates the additional parameter fields that need to be configured to manage the resume process.

When checking files for differences

Compare file attributes: compares the sizes of the existing and original file and if they are the same, then the transfer resumes, otherwise the original file is transferred again.

Compare sparse file checksums: calculates a sparse checksum on the existing file and if the file matches the original, resumes the transfer. If the checksum does not match, the original file is transferred again.

NOTE: A sparse checksum is calculated by using several samples in the data block, not the entire set of data.

Compare full file checksums: calculates a full checksum on the existing file and if the file matches the original, resumes the transfer. If the checksums do not match, the file is transferred again.

When a complete file already exists at the destination: select an overwrite rule when the same file exists at the destination

File Attributes

The File Attributes section defines attributes of files when they are transferred:

Preserve Access Time: sets the access time of the destination file to the same value as the access time of the source file.

*Preserve Modification Time***:** sets the modification time of the destination file to the same value as the modification time of the source file.

Preserve Source Access Time: keeps the access time of the source file the same as its value before the transfe.r

NOTE: Access, modification, and source access times cannot be preserved for node and Shares connections that are using cloud storage.

Source Handling

The Source Handling section defines what to do with successfully transferred files.

Automatically delete source files after transfer: deletes the successfully transferred files from the source after completion.

Automatically move uploaded source files to a directory: If selected, you must select the directory where the files are to be moved (directory must exist to select it).

Delete empty source subdirectories: removes empty source directories (frequently used when deleting source files).

IBM Training

Importing and exporting connections

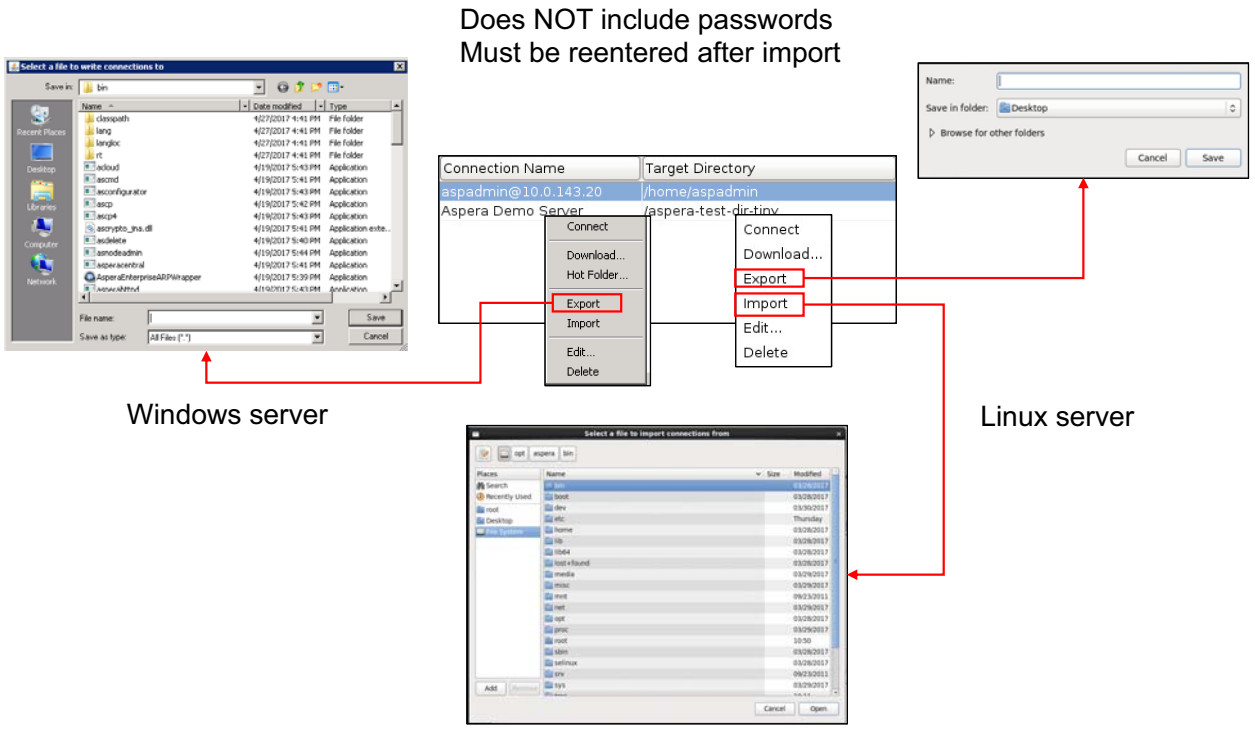


Figure 4-12. Importing and exporting connections

Connections can be exported to a file, and can be imported on another system. Exported connections can minimize the administrative effort that is required when many remote connections are implemented across multiple systems.

Click the right mouse button anywhere inside the right side of the Connections page to open the menu where you can select the Export or Import option.

NOTE: The actual menu that you see is slightly different, depending upon the system where the Aspera software is installed. The Windows version also includes the Hot Folder option, which is not included on Linux systems. Configuration and usage of hot folders is discussed in the Advanced Features module of this training course.

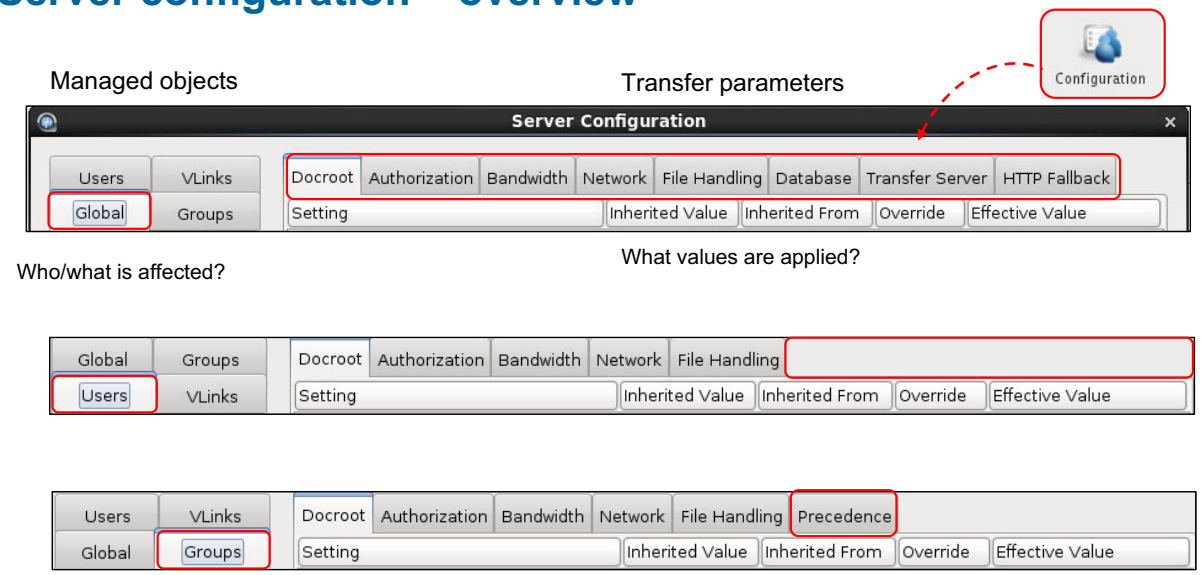
The Export option opens another window that prompts for the name of the file where you want to store the list connection parameters.

The Import option opens a window from which you can select a list of exported connections.

Exported and imported connections do not include passwords. Because password encryption uses a per-user and per-machine cryptographic key, the encrypted passwords cannot be used on other systems. And, because the passwords are not transferred, they must be reentered after the connection is imported.

IBM Training

Server configuration – overview



Values entered written to **aspera.conf** file

Global settings apply to all unless user or group parameters defined individually

Many parameters are the same for users and groups as global configuration

Figure 4-13. Server configuration – overview

Aspera transfer servers are configured by modifying the aspera.conf file. A text editor, the asconfigurator utility (discussed later in this module), or the Aspera GUI are tools for modifying the aspera.conf file.

Clicking the Configuration button opens the Server Configuration window. From this window, you can configure numerous parameters that control the server's characteristics for transferring data. Two groupings of tabs are included that lead to other configuration menu windows. The tabs on the left side of the screen define the object (who or what) affected by the configuration parameters. The tabs on the right present pages that contain the object, parameters for configuration. For example, if the Global tab is selected as the managed object, then the tabs on the right side of the screen can open pages where parameters are configured that apply to all users.

NOTE: Global parameters apply to all users, unless specific user accounts are configured with different values.

Managed Objects

Four tabs can be selected to identify the object to manage:

Users: user accounts must be configured to authenticate transfers. More details about managing users are presented in a separate module of this training course.

Global: sets default parameter values for the system that are used unless otherwise configured for specific user or group accounts

Vlinks: 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. Two default Vlinks are created during the installation process, but they are not activated. Vlinks can be assigned globally, to a specific user, or to a group.

Groups: configures 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 are applied.

NOTE: Aspera HSTS doesn't create user groups in the operating system for you, so you must ensure that the groups exist at the operating system level before adding them to your Aspera HSTS environment

NOTE: Many of the parameters set by the various tabs are the same for Users, Groups, and Global objects. However, some minor differences do exist.

Global Parameter Values

A collection of values that can be defined in the Transfer Parameters section of the Server Configuration window control transfers:

Docroot: Defines what directories and files are accessible when performing transfers.

Authorization: Sets up authentication and authorization parameters.

Bandwidth: Defines bandwidth values for incoming and outgoing transfers.

Network: Specifies network configuration parameters such as which network interface to use for transfers.

File Handling: Configures details about how the server manages transfer files.

Database: Defines parameters for configuring a Database Logger function (a feature that records all transactions to a MySQL database).

Transfer Server: Settings required to configure system as a transfer server.

HTTP Fallback: HTTP Fallback serves as a secondary transfer method when the internet connectivity required for Aspera accelerated transfers (that is, UDP port 33001, by default) is unavailable. When HTTP Fallback is enabled and UDP connectivity is lost or cannot be established, the transfer continues over the HTTP protocol (or HTTPS protocol).

User Parameter Values

The Users configuration page defines the same parameters as the Global configuration page. But the Database, Transfer Server, or HTTP Fallback tabs are not available. These tabs define parameters that do not pertain to individual users or groups, so they are not included.

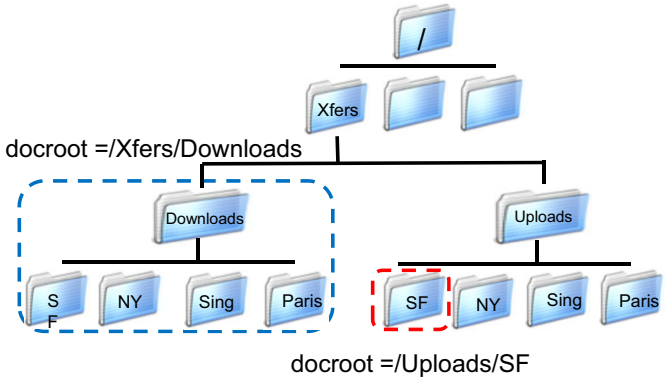
Group Parameter Values

The Groups configuration page is similar to the Users page, but with the additional configuration page for Precedence. Details of how Aspera works with group precedence is discussed in the Managing Users and Groups module of this training course.

Parameters that are defined in the Server Configuration tab pages are written into the aspera.conf file, which is used by the Aspera HSTS software when it is started.

IBM Training

Document root (docroot) parameters



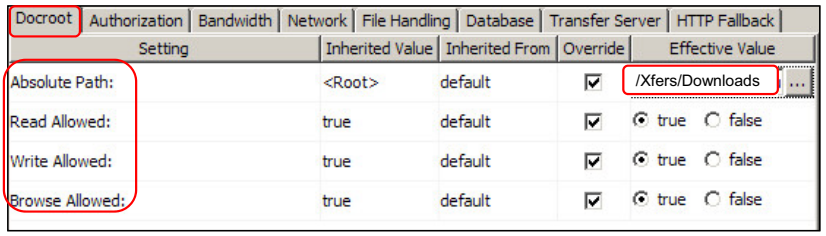
Defines access to file system

Assigned to user/group or global

Default value is undefined

Entries written to **aspera.conf**

Multiple **docroot** values allowed
(conditional based on IP address)
Must be configured manually
<absolute peer_ip="ip_address" >path</absolute>



Format for object storage
protocol://user:password@object)URL/path/[?storage_configuiration]
s3://s3.amazonaws.com/my_bucket

Figure 4-14. Document root (docroot) parameters

A docroot is the area of the file system that is accessible to Aspera users and is identified as a directory. The purpose of specifying a docroot value is to limit user access on the Aspera HSTS file system. The docroot directory value defines the highest level within the server's file system that the user can access. All files below the docroot directory are available to the user.

The Docroot tab in the Server Configuration lists parameters that control what part of the server's file system users are able to access, and their permissions. Four values can be defined in this tab:

Absolute Path

The Absolute Path is a path on local storage to the docroot, the highest level of the file system that is accessible to Aspera users. Assigned users can access directories and files below the docroot entry point.

NOTE: The default empty value gives users access to the entire file system. You must specify a docroot value (mark the Override checkbox and provide a value) to limit access.

You can set a global docroot value, and further restrict access to the file system by group or individual user.

Docroot paths require specific formatting, depending upon where the server's storage is located:

Local storage: set the full path name of the wanted location.

Cloud-based Object Storage: the docroot path syntax is typically a protocol followed by URL-encoded storage access credentials and a path within that storage. The example indicates a docroot setting for Amazon AWS service, but other cloud storage providers are similar. Details about specifying cloud-based docroot values can be found in the *Setting Docroots for Object Storage and HDFS* section of the Aspera HSTS Admin Guide.

Multiple docroot Values

It is also possible to define multiple docroot values and make them conditional based on the IP address from which the connection is made. However, multiple docroot values cannot be configured from within the GUI, but can be defined with appropriate entries in the aspera.conf file:

```
<absolute peer_ip="ip_address">path,/absolute>
```

See the Document Root section of the *IBM Aspera HSTS Administration Guide* for details about configuring multiple docroot values.

Permissions

You can also control Aspera users assigned the docroot value by defining permissions for them in the Docroot window:

Read Allowed

Setting this parameter to true allows users to transfer from the designated area of the file system as specified by the Absolute Path value.

Write Allowed

Setting this parameter to true allows users to transfer to the designated area of the file system as specified by the Absolute Path value.

Browse Allowed

Setting this parameter to true allows users to browse the directory.

IBM Training

Authorization parameters

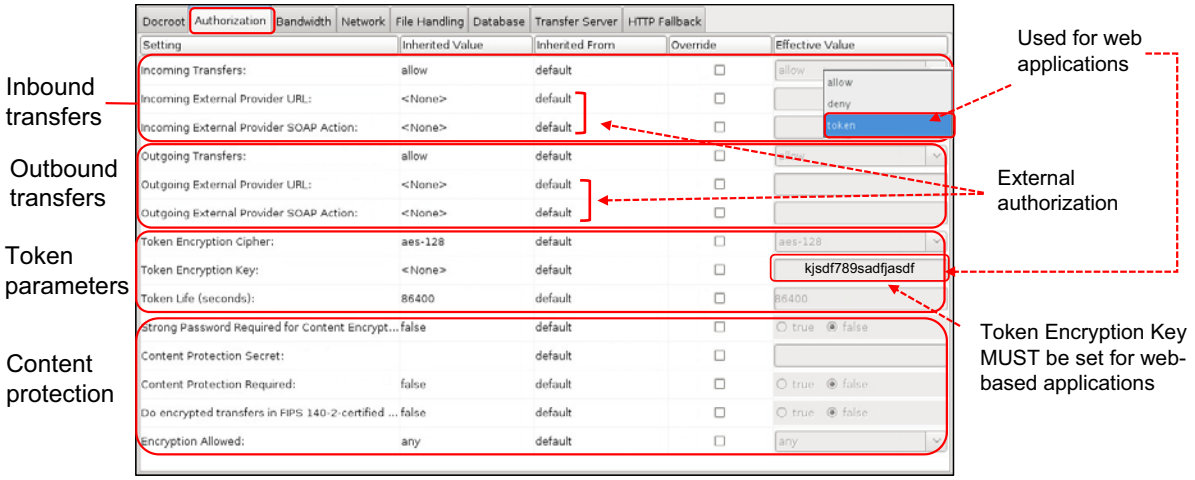


Figure 4-15. Authorization parameters

The Authorization tab contains parameters that define what kind of transfers are allowed, and what is required when those transfers occur. Numerous parameters can be configured. You can define these authorization parameters for individual users, groups, or as global settings.

For more information, see the appropriate *IBM Aspera HSTS Administration Guide*.

Incoming and Outgoing Transfers

The default setting of allow enables users to transfer to or from this computer, depending upon whether the Incoming or Outgoing parameters are configured. Setting this value to deny prevents all transfers to (or from) this computer. When set to token, only transfers initiated with valid tokens are allowed to transfer to (or from) this computer. Web applications such as Aspera Faspex or Aspera Shares use tokens for their transfers.

External Provider URL: enter the URL of the external authorization provider for incoming transfers. Aspera servers can be configured to check with an external authorization provider. This SOAP authorization mechanism can be useful to organizations with custom authorization rules. The default empty setting disables external authorization.

External Provider SOAP Action: SOAP action required by the external authorization provider for incoming transfers.

Token Parameters

When tokens are to be used for transfers, you must set a token encryption text phrase that is used to generate encrypted authorization tokens. You can also select the Token Encryption Cipher (aes-128, aes-192, or aes-256) and other values associated with token usage.

NOTE: Aspera suggests setting a token encryption key of at least 20 random characters

Content Protection

If uploads and downloads are to be password protected, it is necessary to identify the password parameters in the Authorization window.

*Strong Password Required for Content Encryptio*n: When set to true, requires the password for content encryption to contain at least six characters. At least one of these characters must be non-alphanumeric, at least one must be a letter, and at least one must be a numeric digit.

Content Protection Required: When set to true, users are required to enter a password to encrypt the files on the server when they upload a file. Users downloading files are given the option to decrypt during transfer.

Encryption Allowed

Describes the type of transfer encryption that is allowed for transfers. 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.

As with all configuration parameters, you can define values for global control, but provide specific users or groups with their own authorizations.

IBM Training

Bandwidth parameters

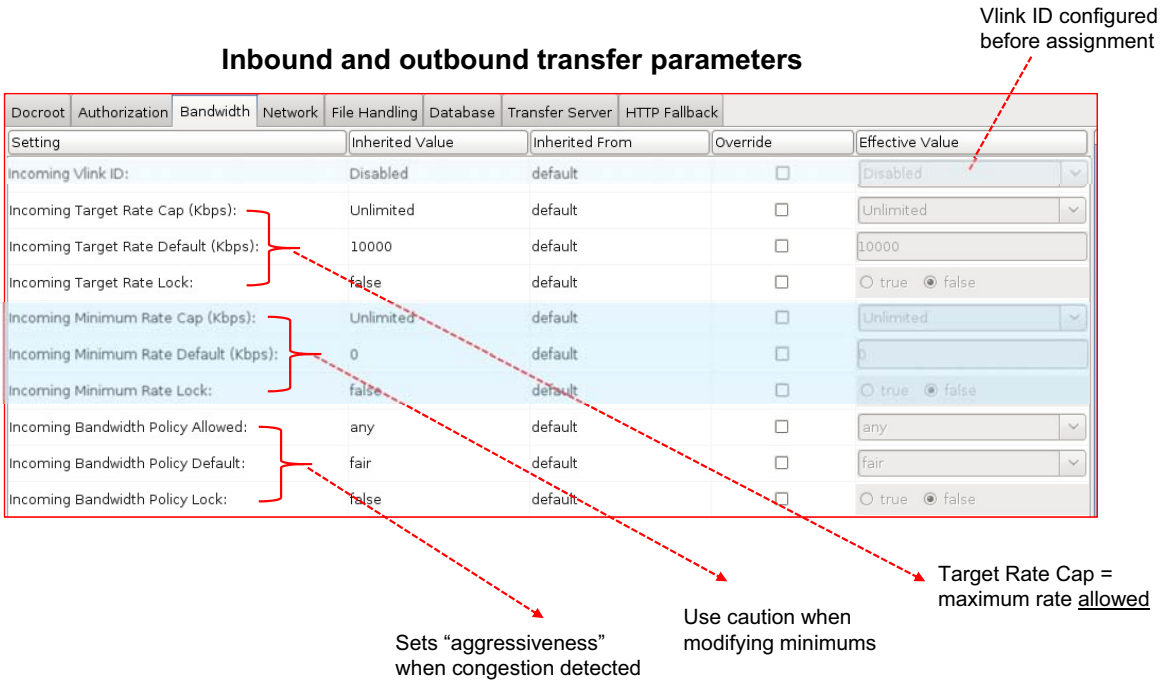


Figure 4-16. Bandwidth parameters

Managing the rate that data is transferred is a significant feature of Aspera HSTS. The “Bandwidth” tab of the “Server Configuration” window defines many parameters that control and influence both inbound and outbound transfers.

Vlink ID: selects Vlink ID used for transfers from the menu. Vlink parameters must be configured before assigning them for transfers. While the example indicates only Incoming values, you can also assign Vlink IDs for outbound transfers. Vlink IDs can be assigned for global recognition, meaning that any transfer connections that are not assigned with other Vlink IDs are assigned to the Vlink ID defined for the Global setting. Vlink IDs assigned to specific users or groups use the assigned Vlink ID rather than the Vlink assigned in the Global configuration.

Target Rate Cap: Target Rate Cap is the maximum target rate that a transfer can request, in kilobits per second. No transfer can 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 are denied.

NOTE: If a Vlink ID is assigned to incoming or outbound transfers, the Vlink cap rate is applied. The Vlink Cap Rate is applied regardless of the Target Rate Cap value (assuming the Vlink ID cap rate is not higher than the Target Rate Cap).

Target Rate Default: This value represents the initial rate for transfers. Users can 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 transfer policy configured as Fixed.

Target Rate Lock: Represents the initial rate for transfers, in kilobits per second. After a transfer is started, its target rate can 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.

Minimum Rate Cap: Revel below which a transfer does not slow, despite network congestion or physical network availability. The default value of Unlimited effectively turns off the Minimum Rate Cap.

Minimum Rate Default: Represents the initial minimum rate for transfers.

Minimum Rate Lock: 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.

Bandwidth Policy Allowed: Sets allowed bandwidth policy for transfers. Aspera transfers use fixed, high, fair, and low policies to accommodate network sharing requirements. When set to any, the server does not deny any transfer based on policy setting. When set to high, transfers with a policy of high and less aggressive transfer policies (for example fair or low) are allowed. When set to fair, transfers of fair and low are allowed, while fixed transfers are denied. When set to low, only transfers with a bandwidth policy of low are allowed.

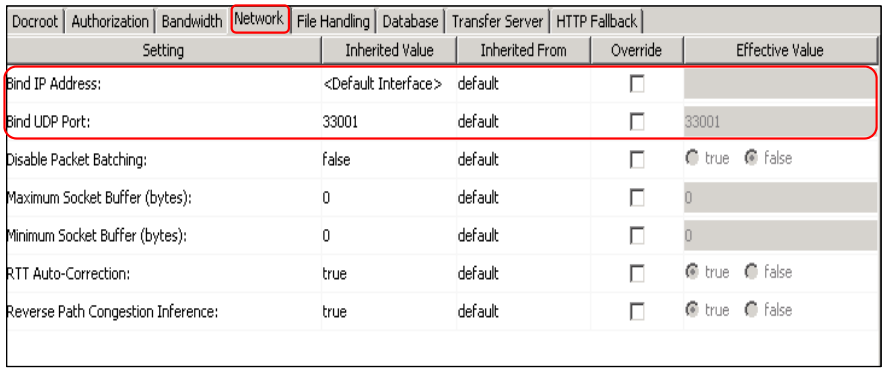
Bandwidth Policy Default: Sets the default bandwidth policy for transfers. Client applications can override the default Policy values.

Bandwidth Policy Lock: After an incoming transfer is started, its Policy can 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.

IBM Training

Network parameters

Binds server-side transfers to specific interface



Most parameters for problem resolution or performance tuning
Modify when working with Aspera Support

Use care when modifying any of these parameters

Figure 4-17. Network parameters

The Network tab on the Server Configuration page provides an interface for configuring various network parameters. Most of these parameters do not require modification for normal operation, but can be useful in resolving problems or tuning transfer performance. In general, the default values work well for most environments and should not be modified.

Bind IP Address

One of the items that can be used in normal environments is the Bind IP Address parameter. This parameter specifies an IP address for ascp to bind its UDP connection when initiating transfers. If a valid IP address is given, ascp sends and receives UDP packets ONLY on the interface corresponding to that IP address.

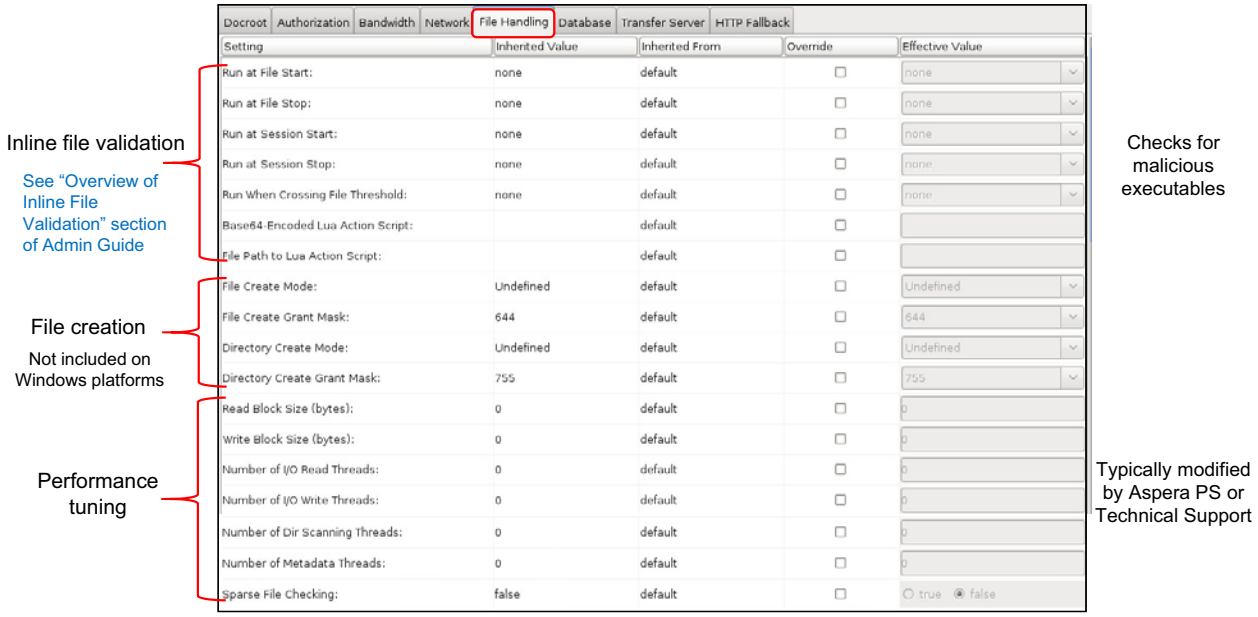
NOTE: This address should only be modified (default value is 127.0.0.1) if you understand the security ramifications of doing so and secured the SOAP service.

Other parameters can be modified from this tab when resolving specific problems with transfers or tuning transfer performance. For more information, see the *General Configuration Reference* section of the *IBM Aspera HSTS Administration Guide*.

NOTE: You should not modify these parameters without a clear understanding of the impact your changes might have

IBM Training

File handling parameters



For more information, see "General Configuration Reference/ File Handling" section of Admin Guide

Figure 4-18. File handling parameters

The File Handling tab of the Configuration Server lists numerous adjustable parameters that can be used to control how transfer files are managed, including file block size, overwrite rules, and exclude patterns. The example shows some of the parameters that are included on this page. More parameters that can be set from this page are discussed on other overheads later in this module.

Inline File Validation Parameters

If an executable file that contains malicious code is uploaded to the server, an external product that integrates with the Aspera environment can activate the malicious code. Inline file validation is a feature that enables file content to be validated while the file is in transit, and when the transfer is complete.

The validation check can be made with a Lua script or with a RESTful call to an external URL. The following parameters set the method of file validation:

- Run at file start
- Run at file stop
- Run at session start
- Run when crossing file threshold

NOTE: It is possible for the file transfer to complete before the file threshold validation response comes back. For example, ascp doesn't pause file transfers during file threshold validation. So, a complete transfer can happen even with validation failure.

For more information, see the *Overview of Inline File Validation* section of the *IBM Aspera HSTS Administration Guide*.

File Creation Parameters

Four parameters can be set to designate how you want permissions set for files that are transferred to the Aspera HSTS system:

File Create Mode: Specifies 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. This parameter takes effect when the server is a non-Windows receiver.

File Create Grant Mask: Used to determine mode for newly created files when a File Create Mode value is not specified. If specified, file modes are set to their original modes plus the Grant Mask values. This parameter takes effect when the server is a non-Windows receiver and when File Create Mode is not specified.

Directory Create Mode: Specifies directory creation mode (permissions). If specified, this parameter creates directories with these permissions regardless of the "Directory Create Grant Mask" value and permissions of the directory on the source computer. As with other parameters, this parameter takes effect when the server is a non-Windows receiver.

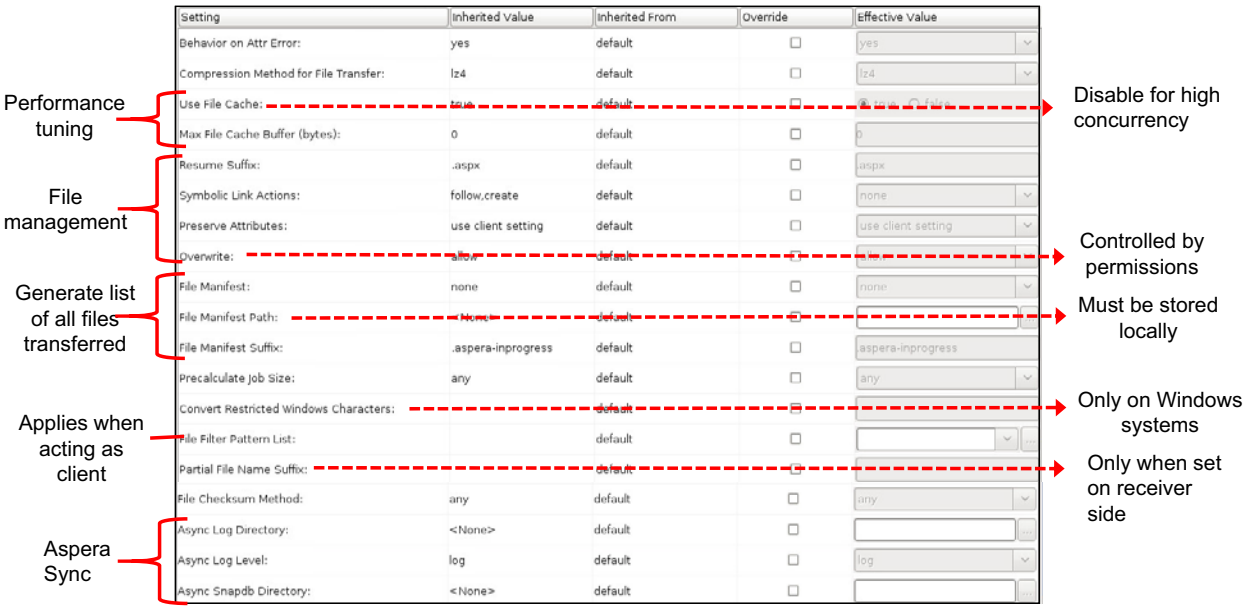
Directory Create Grant Mask: Used to determine mode for newly created directories when the Directory Create Mode parameter is not specified. If specified, directory modes are set to their original modes plus the Grant Mask values. Again, this parameter takes effect when the server is a non-Windows receiver and when a Directory Create Mode value is not specified.

Performance Tuning Parameters

Several parameters are associated with tuning for transfer performance. Typically, these parameters should be modified only when instructed by Aspera Professional Services or Technical Support.

IBM Training

File handling parameters (2)



See “General Configuration Reference/ File Handling” section of Admin Guide

Figure 4-19. File handling parameters (2)

File Cache (Performance Tuning)

File-level memory caching improves data write speed on Windows systems in particular, BUT, uses more memory. Aspera suggests that you use file caching on systems that are transferring data at speeds close to the performance of storage. Disable file caching for systems with high concurrency, as memory use grows with the number of concurrent transfers.

File Management

Several parameters can be set to manage transferred files, including:

- The file name extension that is used for temporary metadata files that are used for resuming incomplete transfers.
- How Aspera should deal with symbolic links on the server.
- Whether the attributes of files should be preserved when written on the receiver system.

NOTE for Windows Platforms: If you change the resume suffix value, you must restart the Aspera Sync service in order for hot folders to pick up the new setting

NOTE: Setting the Overwrite parameter to deny does not prevent clients from overwriting files. The file permissions set for the target directory control that action.

File Manifest

The file manifest is a file that contains a list of everything that was transferred in a transfer session. The file name of the manifest itself is automatically generated based on the transfer session's unique ID. The File Manifest Path value specifies the location where each manifest is written. If no path is specified, the file is generated under the destination path at the receiver, and under the first source path at the sender.

File Filter Pattern

You can specify patterns for excluding or including files with the specified pattern in their name. Inclusions start with + followed by a white space and the wanted pattern, and exclusions start with - followed by white space and the desired pattern. You can also use the * and ? characters to match every character (*) or a single character (?).

See the *Applying Filters to Include and Exclude Files* section in the *IBM Aspera HSTS Administration Guide* for more details about using filter patterns.

NOTE: This option works only when the server is acting as a client. Servers cannot exclude files or directories that remote client systems upload or download.

Partial File Name Suffix: Specifies the file name extension that is created on the destination computer while the file is being transferred (only takes effect when set on the receiver system). After the file is finished transferring, this file name extension is removed. If a hot folder is the upload destination, define the partial file name suffix, even if it means setting it to the default value (.partial). Setting this value prevents partial files from being downloaded from a hot folder.

File checksum method: sets the type of checksum method that is used to calculate for transferred files. Check the override box and select the preferred method value: md5, sha1, or any.

Aspera Sync Logging

Aspera Sync is an add-on product that offers multidirectional asynchronous file replication and synchronization, by using the services of Aspera HSTS. If Aspera Sync is implemented, the Async parameters allow for control of logging information. These parameters only impact implementations that use Aspera Sync, and have no effect on other IBM Aspera HSTS functions.

Async Log Directory: Identifies an alternative location for the Aspera Sync (separate product) server's log files. If empty, log files go to the default location, or the location specified by the client with -R.

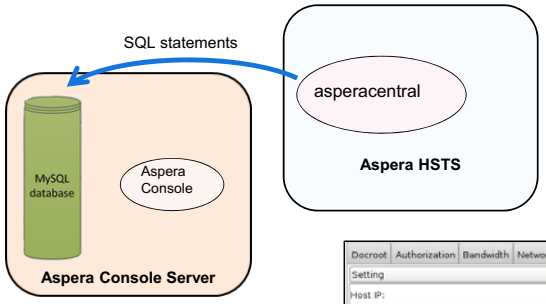
Async Log Level: Selects the amount of detail that is written to the Sync server activity log. The choices are: disable, dbg1, and dbg2.

Async Snapdb Directory: Identifies an alternative location for the Aspera Sync server's snapshot DB files.

IBM Training

Database parameters

Configures parameters related to MySQL database that stores server and transfer data



SQL statements created and “pushed” to remote database

Deprecated feature for use w pre 3.0 versions of Aspera Console

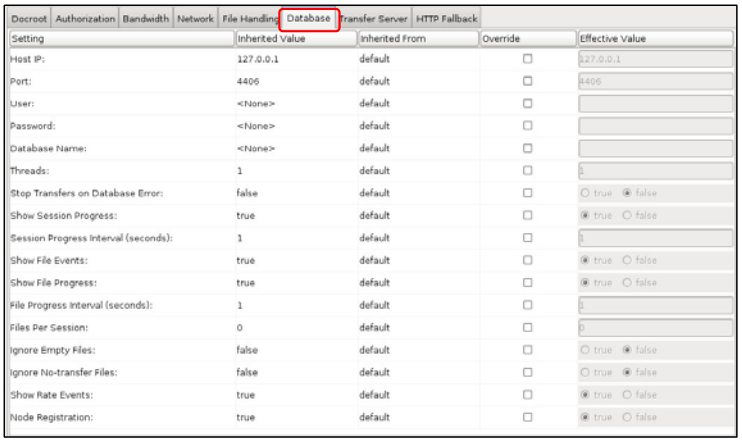


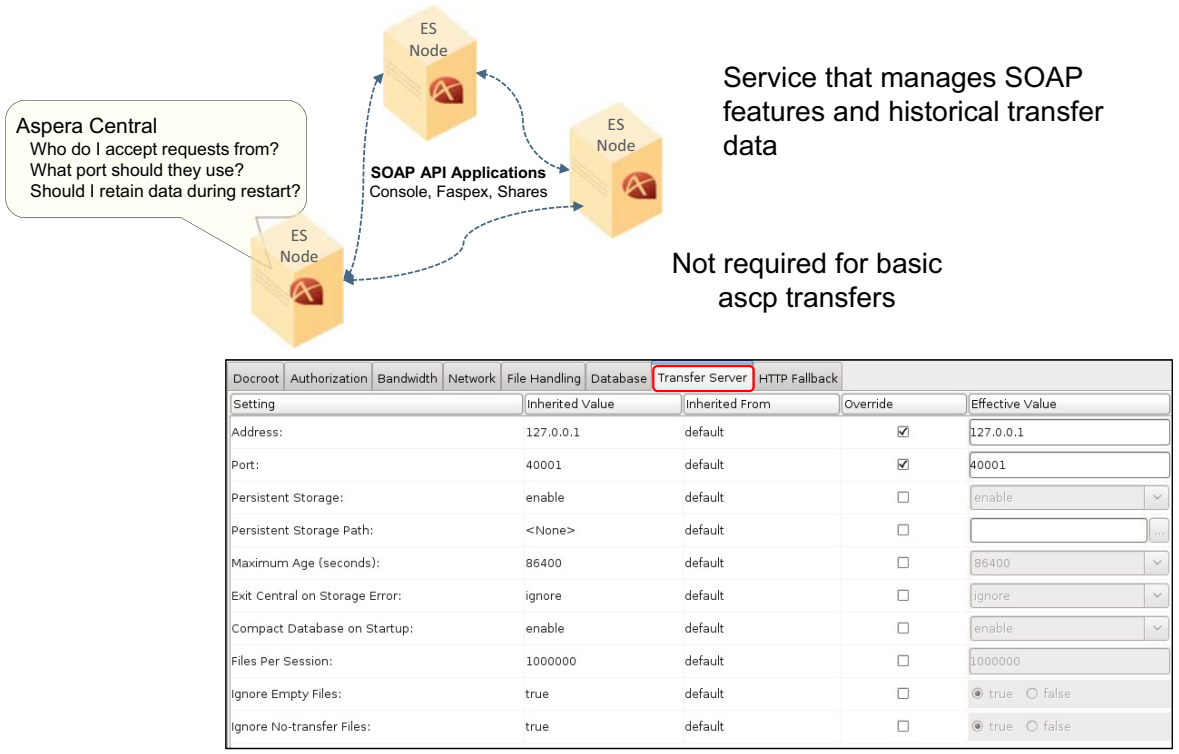
Figure 4-20. Database parameters

The Database parameters define the configuration details for the deprecated Database Logger function, which facilitates logging of all transfers to a MySQL database. This feature is required if you are using a version of Aspera Console that is older than 3.0. Newer versions of Console use Node API to poll transfer servers to collect data, rather than depending upon each server to create SQL statements and pushing those statements to the Console system.

The asperacentral daemon runs on IBM Aspera HSTS and is responsible for providing the logging information to the MySQL server. The parameters in the Database tab specify how asperacentral communicates with the remote database.

IBM Training

Transfer server parameters



Configuring IBM Aspera High-Speed Transfer Server

Figure 4-21. Transfer server parameters

The Transfer Server tab opens a page that lists parameters that are used by the server's asperacentral service. If the server interacts with IBM Aspera Faspex, IBM Aspera Shares, or IBM Aspera Console, the Persistent Storage parameter must be enabled. The default values are usually sufficient for most Aspera deployments.

IBM Training

HTTP fallback parameters

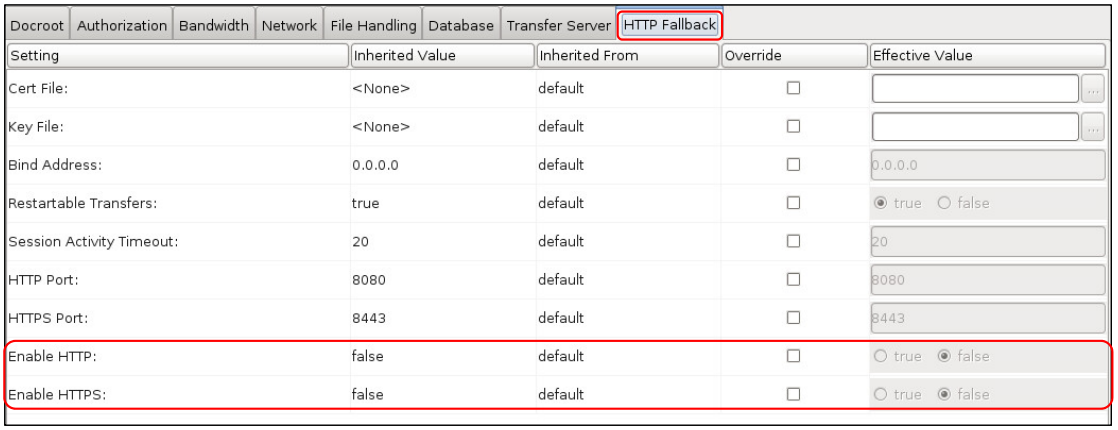


Figure 4-22. HTTP fallback parameters

Aspera systems can be configured to support the use of HTTP for data transfers when FASP transfers cannot be performed for some reason. Transferring files over HTTP results in much slower data transfers, but at least the transfer completes. The fallback function must be configured to support HTTP or HTTPS. The Restartable Transfers parameter is also included on this page (default is enabled) which must be enabled in order for transfers to resume at the point of interruption.

More information about configuring HTTP fallback is presented in the Advanced Features module of this training course

IBM Training

Vlinks

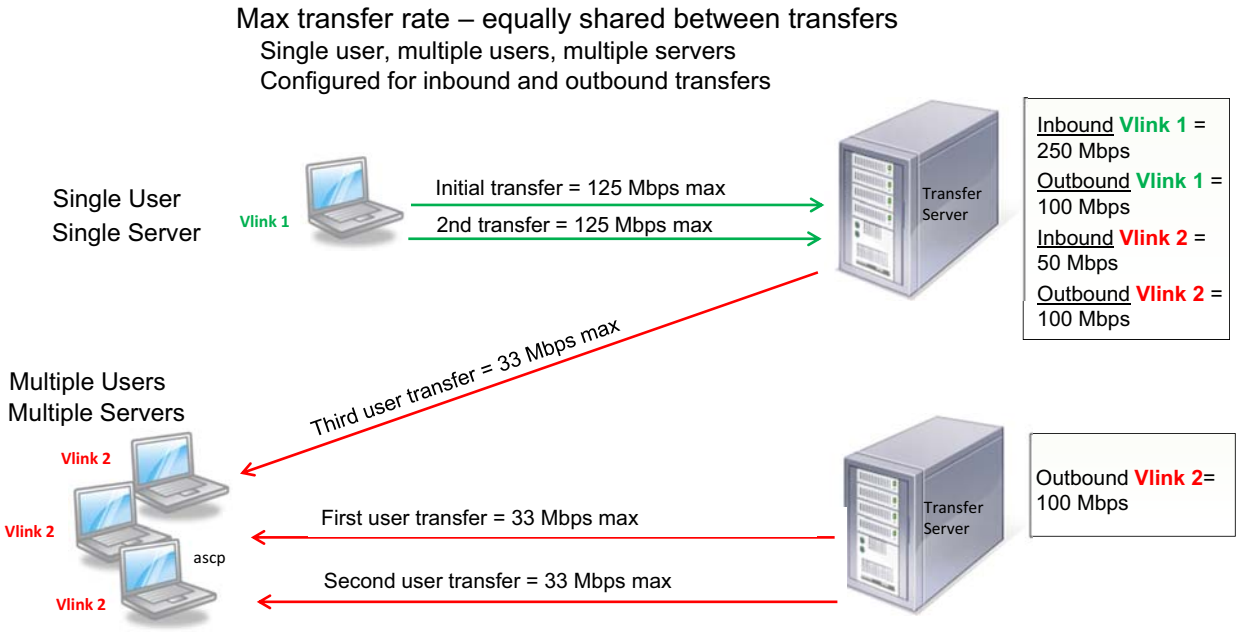


Figure 4-23. Vlinks

IBM Training

Creating Vlinks

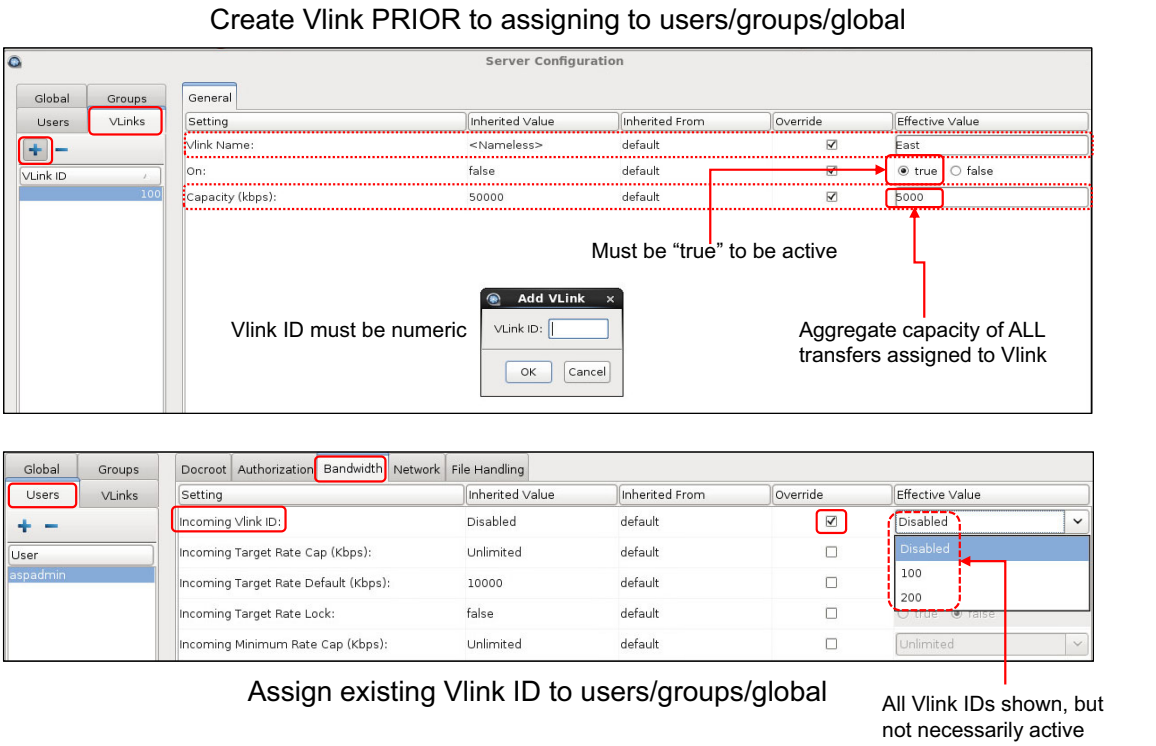


Figure 4-24. Creating Vlinks

To assign a Vlink ID value to a user, group or globally, the Vlink ID must be created and configured. The Vlinks tab opens a page that displays existing Vlink IDs, and offers a link to create new ones or delete existing ones.

Click the **+** symbol at the left of the screen to create a new Vlink ID. The Add Vlink window opens, prompting you for a numeric value for the new Vlink ID.

NOTE: Vlink IDs must be a numeric value.

The Vlinks configuration page is displayed when the new Vlink ID is input. Three parameters can be configured for each Vlink ID:

Vlink Name

The name value has no impact on bandwidth control, so you can use any name that makes sense for your needs. The system uses the Vlink ID rather than the name for managing bandwidth limits, but a meaningful name can be more useful when associating users with a number.

On

The On parameter must be explicitly set to true before the Vlink ID is available for assignment.

NOTE: The default setting for this parameter is false, so it is important to remember to change the setting to true (click the Override checkbox to edit the Effective Value).

Capacity (kbps)

This value sets the virtual bandwidth cap in Kbps. When you assign a Vlink ID to a transfer (user, group, global), the total bandwidth of all transfers that are assigned to this Vlink ID is restricted to this value.

After Vlink IDs are configured, they can be assigned to users, groups, or globally from the Bandwidth configuration page. The menu provides the option to select which Vlink ID you want to assign, assuming that Vlink ID was enabled on the Vlinks page.

IBM Training

What you learned

- The Aspera GUI can be started by using the ***asperascp*** command or from the Aspera HSTS icon
- The GUI provides menus for defining connections to remote platforms, which can be made available to regular users who do not need to know the login credentials for the remote system
- The GUI can be used to configure various parameter values for individual users, groups, or globally used values
- The various tabs available to a user with administrative permissions include “Docroot”, “Authorization”, “Bandwidth”, “Network”, “File Handling”, “Database”, “Transfer Server”, and “HTTP Fallback”
- The “Bandwidth” tab provides a page where you can set the maximum bandwidth that is allowed and the default target rate for inbound and outbound transfers
- The “Docroot” page specifies permissions for files and directories under the “Absolute Path”
- Vlinks allows for “virtual” bandwidth caps that provide an aggregate bandwidth for all transfers that are assigned to the same Vlink ID
- Vlink bandwidth caps can function across multiple Aspera transfer servers
- Vlink IDs must be created before they can be assigned to users, groups, or globally
- Vlink IDs can be assigned to transfers, but do not function if not explicitly made active

Figure 4-25. What you learned

IBM Training

Unit summary

- Navigate the Aspera GUI to access various configuration parameters
- Identify the kinds of global parameters that may be configured
- Define maximum bandwidth and default target rates for transfers
- Manage file permissions for inbound/outbound transfers
- Define and implement Vlinks

Figure 4-26. Unit summary

Review questions (1 of 2)

1. Which of the following values are valid values for the “Transfer Policy” parameter on a production system? Select all that apply.
   A. Fixed
   B. Fair
   C. Low
   D. High

2. True or False:
   A Vlink ID can be assigned any value.

Figure 4-27. Review questions (1 of 2)

IBM Training

Review questions (2 of 2)

3. Which of the following statements best describes why a transfer might fail when attempting to download a file with a connection defined in the Aspera GUI? Select all that apply:
   A. The “Absolute” parameter for the transfer user on the remote system does not allow access to the requested file
   B. The remote Aspera system transfer user has the “Authorization/Outbound” parameter set to “deny”
   C. The remote Aspera system transfer user has the “Authorization/Inbound” parameter set to “deny”
   D. The remote Aspera system transfer user has a “Target Rate” parameter setting that is greater than the network bandwidth available

4. True or False:

   You can use the Aspera GUI to configure “HTTP Fallback” services for an individual transfer user

Figure 4-28. Review questions (2 of 2)

IBM Training

Review answers (1 of 2)

1. Which of the following values are valid values for the “Transfer Policy” parameter on a production system? Select all that apply.
   A. Fixed
   B. Fair
   C. Low
   D. High

   The answer is A

2. True or False:

   A Vlink ID can be assigned any value.

   The answer is False. Vlink ID values must be numeric and must be a value in the range 1 - 255

Figure 4-29. Review answers (1 of 2)

IBM Training

Review answers (2 of 2)

3. Which of the following statements best describes why a transfer might fail whe attempting to download a file by with connection defined in the Aspera GUI? Select all that apply:

   A. The “Absolute” parameter for the transfer user on the remote system does not allow access to the requested file
   B. The remote Aspera system transfer user has the “Authorization/Outbound” parameter set to “deny”
   C. The remote Aspera system transfer user has the “Authorization/Inbound” parameter set to “deny”
   D. The remote Aspera system transfer user has a “Target Rate” parameter setting that is greater than the network bandwidth available

   The answer is A and B

4. True or False:

   You can use the Aspera GUI to configure “HTTP Fallback” services for an individual transfer user

   The answer is False. The HTTP Fallback is typically configured when using Aspera web-based application and must be configured as a system-wide value that affects all transfer users

Figure 4-30. Review answers (2 of 2)

IBM Training

Lab Exercise 2

In this exercise, you modify various Aspera parameters to manage global and individual transfers.

You use the Aspera GUI to define a connection on one Linux system that facilitates transfers between that server and the other Linux system. You then share that connection with users who do not have administrative permission on the system.

- **Make sure you work on the correct server!**

  You switch between the two Linux servers frequently, so make sure you perform the tasks on the correct system.

  Look for the underlined **Switch to the Denver server** or **Switch to the London server** to confirm which server you should be using for the following tasks.

- **Don't skip the questions!**

  Try to answer the questions that are asked as you perform the various tasks.

  The questions are designed to make you think about what the system is doing and to draw attention to specific values associated with transfers.

Figure 4-31. Lab Exercise 2

Unit 5. Managing Aspera users and groups

Estimated time

01:00

Overview

This unit addresses the basic configuration for adding and managing Aspera transfer users and groups

How you will check your progress

- Exercise

IBM Training

Unit objectives

- Distinguish between a system user account and a transfer user account
- Identify what system user account parameters need to be modified to properly support Aspera transfer services
- Create Aspera transfer users and groups
- Describe the precedence of configurations for user, group, and global settings
- Verify user account's ability to perform FASP-based transfers

Figure 5-1. Unit objectives

This module focuses on the parameters that are associated with managing users and groups that use the Aspera GUI. Command-line options for these configurations are discussed in the Command-Line Operations module of this course.

IBM Training

System user versus transfer user

C:\Program Files (x86)\Aspera\Enterprise Server\etc\

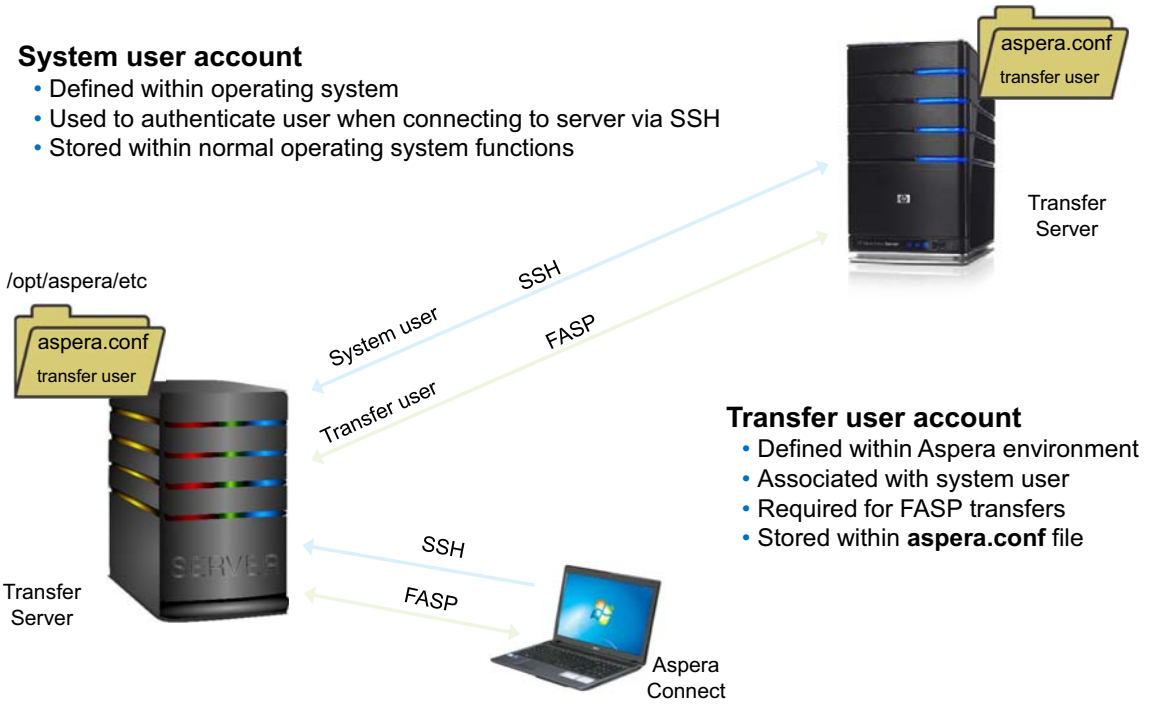


Figure 5-2. System user versus transfer user

Before discussing the management of Aspera user accounts, it is important to understand the relationship between operating system users and Aspera transfer users. Aspera software depends upon SSH to authenticate users with the operating system. After a user is authenticated with the operating system, an ascp process is started, which is responsible for communicating with the remote ascp process to transfer files with the FASP protocol. The system user account data is stored within whatever structure the operating system uses for managing users.

The Aspera user account is referred to as the transfer user account, as it is this account that defines what the user can and cannot do within the Aspera environment.

It is necessary for each Aspera user account to be associated with a system account, which is why you can add an Aspera user only from a list of system users. The transfer user data is stored within the `aspera.conf` file, which is separate from the system user store method.

It is useful to understand this distinction when troubleshooting transfer problems. For example, if the problem seems to be an authentication issue, then the likely source is in the operating system configuration. If the problem appears to be within the transfer process, then it makes sense to focus on the Aspera configuration to identify the problem.

IBM Training

Creating transfer user accounts

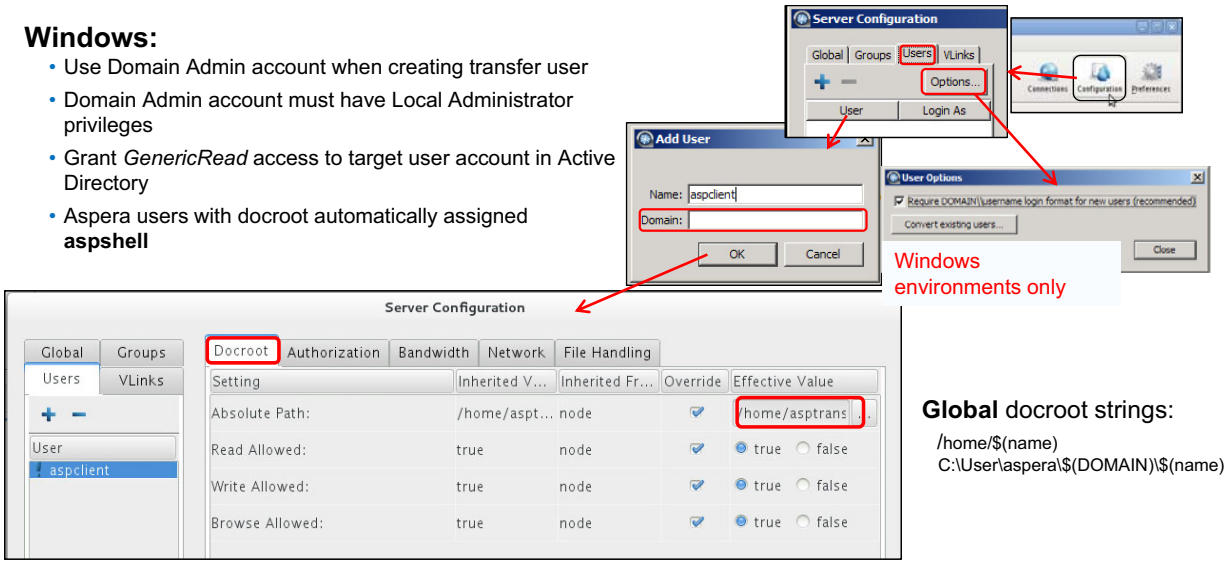
System or domain user account must exist!

Linux

Change login shell to ***aspshell***

Windows:

- Use Domain Admin account when creating transfer user
- Domain Admin account must have Local Administrator privileges
- Grant *GenericRead* access to target user account in Active Directory
- Aspera users with docroot automatically assigned **aspshell**



Global docroot strings:

/home/$(name)
C:\User\aspera\$(DOMAIN)\$(name)

Figure 5-3. Creating transfer user accounts

As mentioned previously, the IBM Aspera HSTS software uses existing operating system user accounts. If the system user accounts do not exist on the server where Aspera IBM Aspera HSTS is configured, you need to add them before proceeding. Depending upon customer requirements, you can create shared accounts for performing Aspera transfers (for example, transfers or asp1), or you can configure individual user accounts for Aspera transfers (for example, bsmith, sjones). The actual account names are not important, but you must create transfer users within the IBM Aspera HSTS application that are the same as the system user accounts.

NOTE: Trying to create a transfer user that does not exist as a system user generates an error message.

Transfer user accounts must be configured on the IBM Aspera HSTS and are used to manage transfer requests by Aspera clients and other Aspera servers.

Modify System User Accounts

If you are running Linux on your server, you might want to modify the account's login shell to limit the user's file manipulation operations. This task involves editing the `/etc/passwd` file and changing the selected user's shell from the default shell, for example, `/bin/bash` to `/bin/aspshell`. The aspshell allows a user to run only Aspera uploads and downloads from the IBM Aspera HSTS system, establish connections within the Aspera application, and browse, delete, rename, or list contents from within the Aspera environment.

Create Transfer Users

Open the Server Configuration window and select the Users tab from the object management section.

Click the + icon to open the Add User window where you input the user's name.

If you are running IBM Aspera HSTS on a Windows system, you can also input the user account's domain name. The Windows version of the Aspera GUI also provides an Options button for requiring Windows domain users to use the DOMAIN/Username format when logging in to their account. This restriction applies only when creating new users. However, you can also convert existing user accounts that do not require the DOMAIN/Username login format with the Convert existing users button.

Note: You cannot add a username with an @ symbol, except when using the user@domain format. For more information, see the *Product Limitations* in the *Aspera HSTS Windows User Guide*.

As discussed previously, a docroot value can be specified to restrict a user's access to a specific directory.

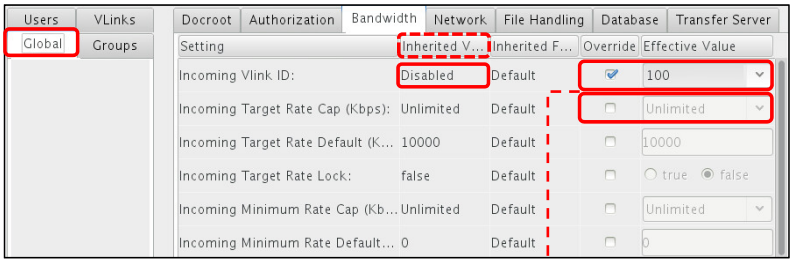
If all transfer user accounts are located with a pattern in their path, you can use a substitutional string to assign an independent docroot value to each account without setting it for each user. For example, rather than setting each user's docroot value, you might set the Global docroot value to be something like `/home/$(name)` or `C:\aspera\$(DOMAIN)\$(name)`. The value in the Global setting is applied to all users (unless you checked the Override box in the user's account and created a unique docroot value).

IBM Training

Global versus user parameters

Simple account configuration
Flexibility to meet all requirements

Global parameters



Inherited Value = user default
Override = explicit value

User parameters

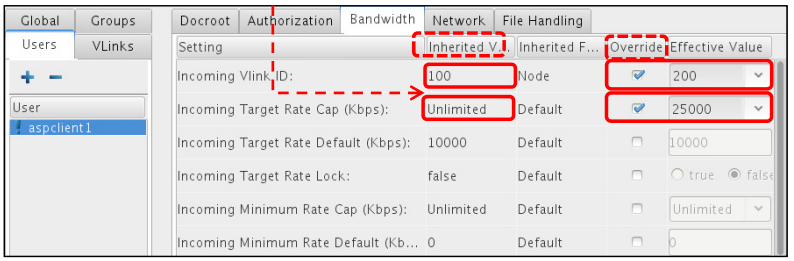


Figure 5-4. Global versus user parameters

In addition to configuring docroot values for users, you might also need to configure other user-specific parameters that override the previously set Global parameters.

Defining global parameter values ensures that all transfers operate with a minimum setting, even if a parameter in a user account was not specified. Global values also minimize the amount of configuration that must be done for each transfer user account.

Global parameters define the values that act as the default, unless accounts are configured to override the global default. As previously discussed, the Global configuration settings define the default values for various parameters. These default values will be assumed for all user accounts that are created after the Global parameter is set. However, they do NOT change settings for existing accounts with the parameter that is explicitly set.

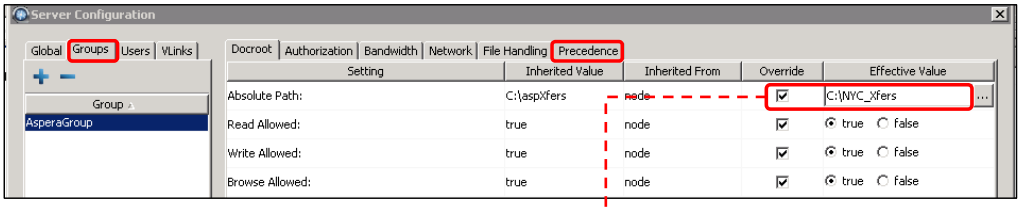
As far as the Server Configuration settings are concerned, the value set in the Global tab becomes the Inherited Value for User accounts. User accounts can be explicitly set by marking the Override checkbox and setting a value to use instead of the global value. Default values are assigned to all Global parameters when the IBM Aspera HSTS software is installed. Overriding the default values in the Global configuration results in a new Global value, which becomes the default value that is used for that parameter for all new accounts. For example, the example the Inherited Value of the Global Bandwidth parameter Incoming Vlink ID might be Disabled, but the Effective Value can be changed to a value of 100. Therefore, the User accounts reflect an Inherited Value for their Incoming Vlink ID of 100, and an Inherited From value of Node, rather than Default. The Effective

Value defined by the Global parameter is the Inherited Value for the User parameter, which can still be set for the User parameter.

IBM Training

Transfer groups

Groups parameters

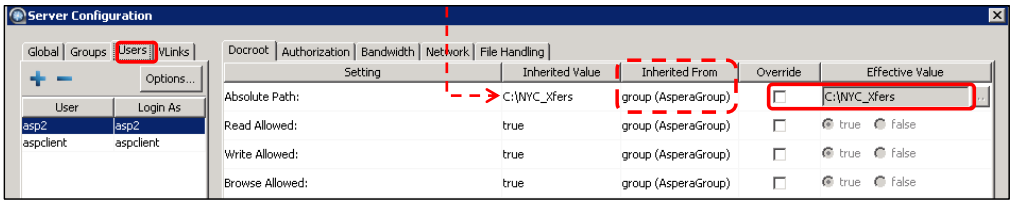


Values for all users in group*

Minimizes effort
Standard values

*If not assigned to individual user

Users parameters



Managing Aspera users and groups

Figure 5-5. Transfer groups

You can also configure transfer settings that are associated with your system's user groups. If users within a group do not have individual transfer settings and the Group parameters is configured within Aspera Transfer Server application, then the group's parameter values are applied for that user. If you are configuring multiple transfer user accounts, you can save time by identifying groups of users that need the same parameter values and creating those groups before creating individual user accounts. This approach minimizes the need to configure parameters for each individual user account, and hopefully avoids transfer problems due to parameter values that are not configured for each user.

NOTE: IBM *Aspera HSTS does* not create user groups on the operating system for you, so you must ensure that the groups currently exist before adding them to IBM Aspera HSTS

Configuring group parameters is almost the same as configuring global or user parameters, but with the Groups tab selected, and with the additional Precedence parameters.

The parameter values defined for the Groups become the Inherited value for user accounts, unless overridden by the user's configuration. Any values not set in the user's account, assume the values that are defined in the Groups setting. The next page discusses how the system determines the precedence for configuration parameters.

IBM Training

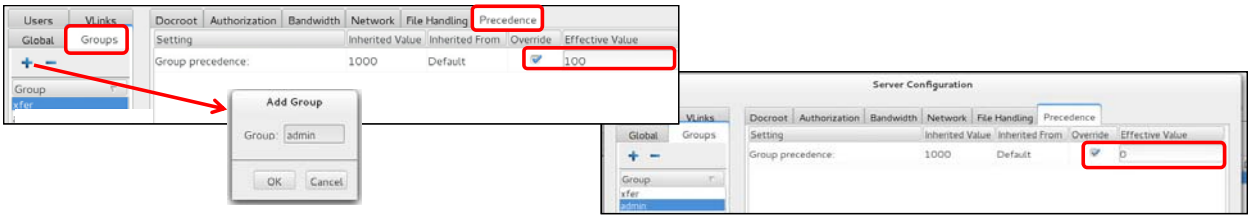
Transfer groups precedence

Zero is highest precedence level

Precedence for multiple groups

1 – User values
2 – Group values
3 – Global values
4 – Default values

Options	User asp1's Settings	Group admin's Settings (Precedence 0)	Group xfer's Settings (Precedence 100)	Global Settings	Default Settings (Precedence 1000)
Target rate	**5 Mb**	10 Mb	15 Mb	40 Mb	45 Mb
Min rate	Not specified	**2 Mb**	8 Mb	3 Mb	0 Mb
Policy	Not specified	Not specified	**Low**	Fair	Fair
Docroot	Not specified	Not specified	Not specified	**/home/$(name)**	Not specified
Encryption	Not specified	Not specified	Not specified	Not specified	**Any**



Precedence values specified in ***aspera.conf*** file

Figure 5-6. Transfer groups precedence

Aspera transfer servers give precedence to parameter settings as follows:

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

If a user is a member of multiple groups, a precedence setting can be assigned to each group. The example shows the setting values that a user, asp1, is assigned in bold.

In this example, asp1 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 100.

Based on the system's order of precedence and the group Precedence settings, the following values apply to the user's transfers:

Target Rate – 5 Mb

Min Rate = 2 Mb

Policy = Low

Docroot = /home/$(name)

Encryption = Any

If the Precedence value for the admin group was changed to 200, the following values would be used:

Target Rate – 5 Mb

Min Rate = 8 Mb

Policy = Low

Docroot = /home/$(name)

Encryption = Any

A group's Precedence value is set from the Group tab of the Server Configuration window, as shown.

No groups are defined by default. You can add groups by clicking the + link and providing the group's name.

NOTE: Groups must exist at the system level before attempting to add them in IBM Aspera HSTS. Add them using the operating system's utility.

The Precedence values that you set for groups are written to the aspera.conf file when you save your changes. Alternatively, you can assign Precedence values by manually editing the aspera.conf file. Open the file and locate the entry for each group name, add the <precedence> option, and assign a precedence value as shown:

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

What you learned

- System user accounts are used to validate a connection to the IBM Aspera Transfer Server via SSH
- Aspera transfer user accounts are defined within Aspera and are tightly coupled with system user accounts
- Linux system user accounts should have their login shell set to **aspshell**
- Transfer users and groups can be created from within the “Server Configuration” page of the Aspera GUI
- Global parameter values can be defined to act as a default for any parameters that are NOT set for a group or a user account
- Parameter values assigned to a user take precedence over corresponding group parameters, which take precedence over global parameter values
- A precedence value can be assigned to groups to manage how permissions are managed for users in multiple groups

Figure 5-7. What you learned

IBM Training

Unit summary

- Distinguish between a system user account and a transfer user account
- Identify what system user account parameters need to be modified to properly support Aspera transfer services
- Create Aspera transfer users and groups
- Describe the precedence of configurations for user, group, and global settings
- Verify user account's ability to perform FASP-based transfers

Figure 5-8. Unit summary

Review questions

1. Which of the following statements are most correct regarding Aspera transfer users? Select all that apply:
   A. Transfer and system users are associated to authenticate logins and manage file access by using system permissions
   B. The primary purpose of transfer users to define connections
   C. Transfer user **docroot** values are automatically configured to be the associated system user's home directory
   D. Transfer user accounts can be configured to use different parameter values from those values that are assigned to other users or groups

2. True or False:
   Aspera configuration parameters provide a way to define the order in which a user's group membership are used to determine file access

Figure 5-9. Review questions

IBM Training

Review answers

1. Which of the following statements are most correct regarding Aspera transfer users? Select all that apply:
   - A. Transfer and system users are associated to authenticate logins and manage file access by using system permissions
   - B. The primary purpose of transfer users to define connections
   - C. Transfer user **docroot** values are automatically configured to be the associated system user's home directory
   - D. Transfer user accounts can be configured to use different parameter values from those values that are assigned to other users or groups

   The answer is A and D

2. True or False:
   Aspera configuration parameters provide a way to define the order in which a user's group membership are used to determine file access

   The answer is True

Figure 5-10. Review answers

IBM Training

Lab Exercise 3

This lab steps you through the tasks of adding and configuring transfer users and transfer groups.

Managing Aspera users and groups

Figure 5-11. Lab Exercise 3

Unit 6. Using command-line operations

Estimated time

02:00

Overview

This unit briefly introduces the use of the Aspera command-line utilities

How you will check your progress

- Exercise

IBM Training

Unit objectives

- Run the appropriate asuserdata command to print all possible aspera.conf entries and their associated asconfigurator commands
- Use asconfigurator utility to modify aspera.conf entries
- Transfer files and directories between Aspera servers using the ascp command

Figure 6-1. Unit objectives

IBM Training

Aspera Transfer Server command-line utilities

asnodeadmin: Configure and manage Node API users

astokengen: Creates token with embedded data (for example, send and receive, source path, destination path, user information, etc.) for transfer authorization

asuserdata: Prints data about users, groups, and validates contents of *aspera.conf* file

asconfigurator: Display or set aspera.conf entries

ascp: Used to initiate and facilitate transfers

asprotect : Encrypts files stored on server

asunprotect : Decrypts encrypted files stored on server

Not Aspera-specific

openssl – Generates RSA Private Key and Certificate Signing Request and generates self-signed certificates

ssh-keygen – Generates an SSH key-pair required for configuring public and private key authentication

Figure 6-2. Aspera Transfer Server command-line utilities

Several command-line utilities are included with the IBM Aspera HSTS software. These commands include:

asnodeadmin:Configure and manage Node API users.

astokengen: Creates token with embedded data (for example, send and receive, source path, destination path, user information, etc.) for transfer authorization.

asuserdata: Prints data about users, groups, and validates contents of *aspera.conf* file.

asconfigurator: Display or set **aspera.conf** entries.

ascp: Used to initiate and facilitate transfers.

asprotect : Encrypts files that are stored on server.

asunprotect : Decrypts encrypted files that are stored on server.

NOTE: For more information about the available commands, see the *IBM Aspera HSTS Administration Guide*. In some cases, more information is available about each utility by running the command with the -help option.

IBM Training

asuserdata: List parameters and *asconfigurator* syntax

 # /opt/aspera/bin/asuserdata -+

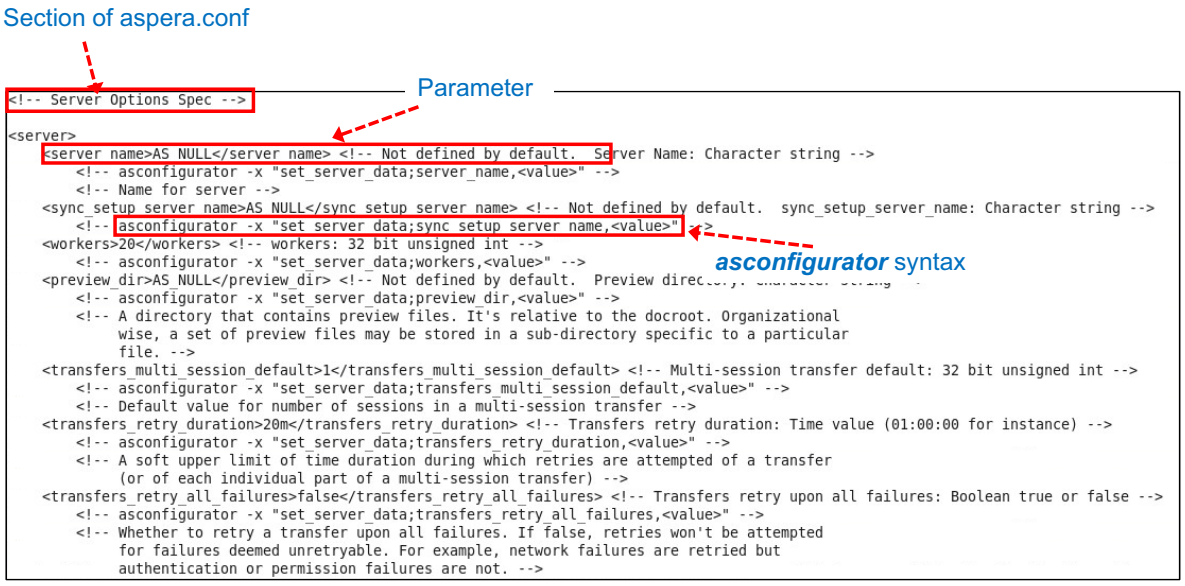


Figure 6-3. asuserdata: List parameters and asconfigurator syntax

The asuserdata -+ command prints a list of all parameters that can be implemented within the aspera.conf file, AND, shows appropriate asconfigurator syntax for modifying the parameter from the command line.

The output of the asuserdata -+ command lists all possible parameter values that are associated with the various sections, for example, User Options, Trunk Options, Server Options, and more).

The example shows the first few lines of the Server Options section, which highlights how parameters are listed and the way to modify the parameter with the asconfigurator command.

IBM Training

The *asconfigurator* utility

- Called by Aspera GUI
- Modifies **aspera.conf** without risk
- Maybe used in scripts

Non-Graphic environments
Use in scripts

- Can create new configuration file
- Can use existing **aspera.conf** as template for new

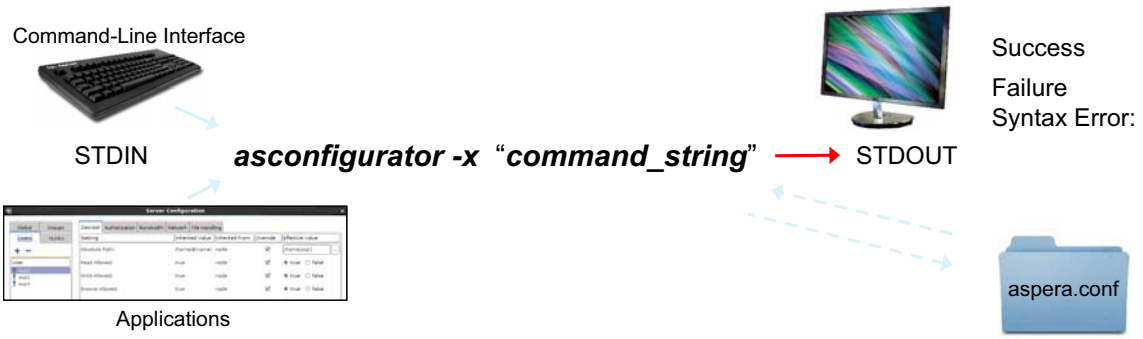


Figure 6-4. The asconfigurator utility

Instead of manually editing the aspera.conf file or implementing their own XML parsers, most Aspera administrators use the `asconfigurator` program. As a command-line utility, `asconfigurator` provides the same level of confidence about the proper structure of the aspera.conf file as the Aspera GUI. The `asconfigurator` utility modifies the `aspera.conf` file based on the arguments provided. The asconfigurator command parses, validates, and writes well-formed XML code while confirming that the values entered for parameters are valid. However, unlike the GUI, the asconfigurator utility does not require a graphic environment, so it is ideal for modifying aspera.conf remotely on systems where a graphic interface is not available. Additionally, asconfigurator command-line entries can be placed into a script, thus providing an automated method for making multiple consistent changes to the aspera.conf file.

Command Location

All Aspera transfer servers (Linux, Windows, or Mac) include the asconfigurator utility. The location of the asconfigurator command is dependent upon the operating system of the system where the Aspera Transfer Server software is installed. The asconfigurator command must be ran with administrative privileges. The `asconfigurator` command is located in the `/opt/aspera/bin` directory on Linux systems.

Windows systems place the utility in different directories for IBM Aspera High-Speed Transfer Server and IBM Aspera High-Speed Endpoint systems:

`C:Program Files (x86)\Aspera\Enterprise Server\bin` and `C:Program Files (x86)\Aspera Point-to-Point\bin`

You might want to set you PATH variable to include the `/opt/aspera/aspera/bin` directory, or the appropriate directory on your system.

Syntax

The asconfigurator utility functions by taking input from STDIN and processes the data that is passed from another application, such as the Aspera GUI. However, also accepts arguments from the command line, which is the focus of this module. The most common use of `asconfigurator` is to modify the default aspera.conf file. However, it is also possible to create a new configuration file, or to copy an existing aspera.conf file to a new configuration file.

Command Output

The output of the `asconfigurator` command indicates whether it was successful or not, and if successful, the modified value is displayed.

IBM Training

asconfigurator command syntax

asconfigurator -x "command;parameter,value"

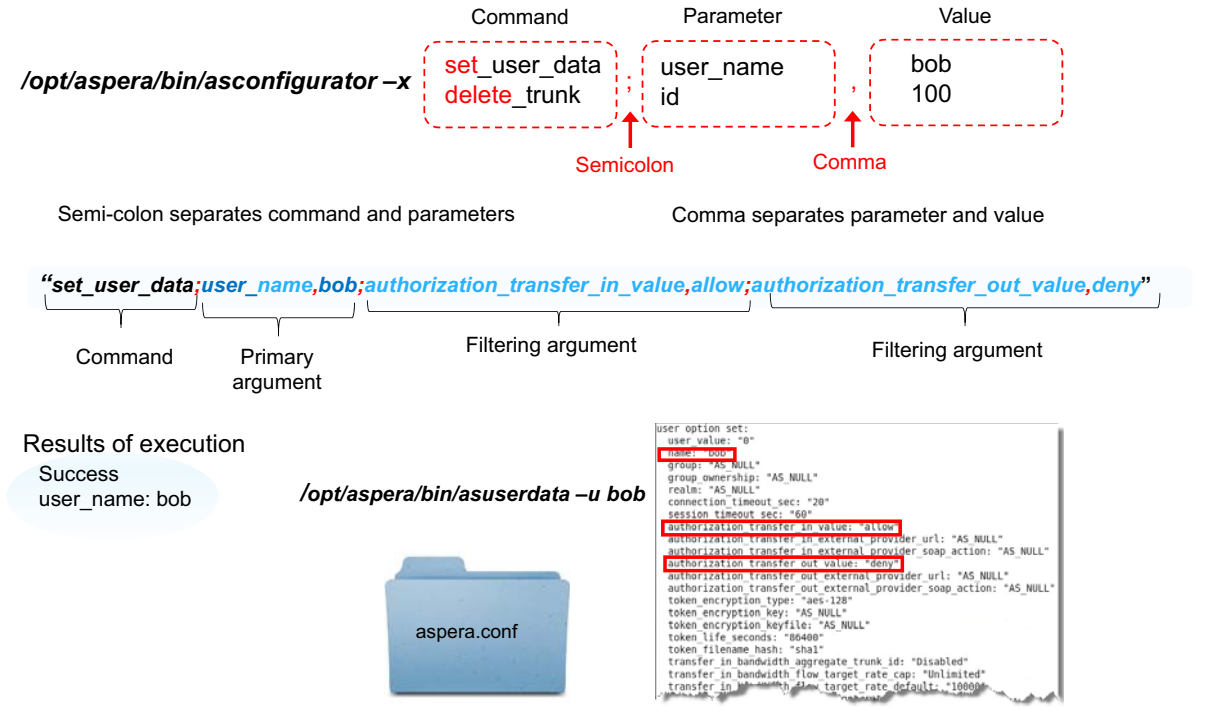


Figure 6-5. asconfigurator command syntax

Set or Delete

The `asconfigurator` command is either a set command (for setting a configuration) or a delete command (for removing a configuration). You can enter one or more sets of parameters and values, which are separated by semicolons

Command

The first argument of the command string is a command that takes the form of either `set_xxx_data` or `delete_xxx`. For example, set_user_data, set_group_data, and set_server_data are valid values of the `set_xxx` command. The command statement is separated from the parameter with a semi-colon (;).

Filtering Arguments

The asconfigurator command recognizes numerous filtering arguments. Filtering arguments define the specific configuration parameters to modify. Filtering arguments consist of a parameter and a value that is associated with that parameter.

Notice that the principal statement and each filtering argument are separated by a semi-colon symbol (**;**). The value that is associated with the parameter is separated from the parameter name with a comma (**,**).

The first argument after the principal statement is the primary filtering argument, and all subsequent arguments that are provided must be associated with the primary argument.

The asconfigurator command recognizes only one primary argument, but multiple filtering arguments can be included, if they apply to the primary argument.

The example shows command string to configure the transfer server so that the user bob can upload files to the server, but not download them. The command (set_user_data) is immediately followed by the semi-colon. The first filtering argument identifies the user_name parameter with bob as the corresponding value. The next filtering arguments the authorization_transfer_in_value with an associated value of allow, and the authorization_transfer_out_value, with an associated value of deny all apply to the primary filtering argument value of user_name,bob.

Results of Execution

As mentioned previously, the output of the asconfigurator utility indicates whether it was successful or failed. The example indicates that bob's account was modified to allow inbound transfers, but deny outbound transfers.

Displaying Configurations

The asuserdata command can be used to display current configurations. The example shows how to display all the configurations defined for the user bob. The output includes the value of all parameters for the user, including those parameters that are not configured (indicated by AS NULL values).

IBM Training

Examples of *asconfigurator* usage

Set hostname
asconfigurator –x "set_server_data;server_name,example.aspera.com"

Set global docroot value to "/Transfers"
asconfigurator –x "set_node_data;absolute,/Transfers"

Change server name and replace SSL/TLS certificate, keeping all other information about server
asconfigurator –x "set_server_data;server_name,myserver;cert_file,/etc/newcert.pem"

Define target rates for outbound transfers for existing group
asconfigurator –x "set_group_data;group_name,FASPex;transfer_out_bandwidth_flow_target_rate_default,10000"

Remove multiple users
asconfigurator –x "delete_user;user_name,asp3"
asconfigurator –x "delete_user;user_name,rlandis"

See "Asconfigurator Reference" section of IBM Aspera High-Speed Transfer Server Admin Guide for more examples

Figure 6-6. Examples of asconfigurator usage

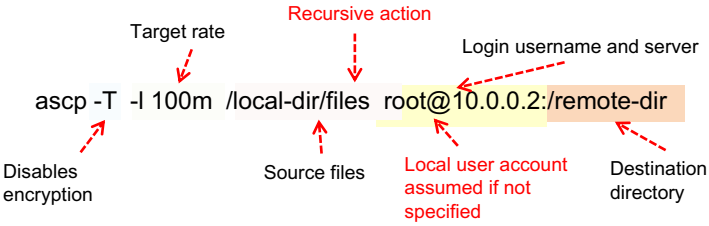
The examples that are shown are provided to assist you in understanding what each type of command sequence is accomplishing.

More details about various parameters and values can be found in the Asconfigurator Reference section of the *IBM Aspera HSTS Administration Guide*.

IBM Training

ascp: Command-line transfers

ascp [*options] source destination*



Command-line entries

- **username@:** used to identify transfer user at source or destination host
  - If not included, defaults to username that is running command
  - If connecting to Windows server as domain user, domain name stripped off
- **Source**: file or directory to transfer - - multiple arguments separated by spaces
- Destination: single file or directory - - use "/" to reference docroot value
- Avoid: / \ " : ? > < & * |

Figure 6-7. ascp: Command-line transfers

The ascp program is a command-line FASP transfer program. The ascp program is powerful, recognizing numerous options that are used to specify every detail of transfer parameters.

Syntax and Environmental Variables

The general syntax for using ascp is shown. The annotated example shows the use of options to send local files to a remote system, by using the root account on the remote system. The example indicates how to transfer all files in the `/local-dir/file`s directory to the 10.0.0.2 server authenticating with the login name of root, with a transfer target rate of 100 Mbps, and NOT adding encryption.

NOTE: If the remote user account is not included, ascp assumes the login name of the user that is running the command. If connecting to a Windows server as a domain user, the domain name is stripped from the username, requiring you to specify the domain and user.

You can set environmental variables on the server that are recognized when transferring from that server.

Options

Numerous options are available for use with the ascp command. For more information about ascp arguments, see the *ascp Command Reference* section of the *IBM Aspera HSTS Administration Guide*.

IBM Training

Command-line transfer: general examples

Transfer local files to remote destination – setting target rate

ascp -l 100m local-dir/files "User Name@10.0.143.2:/remote directory"

Transfer local files to remote destination, then remove source files

ascp --remove-after-transfer local-dir asp1@10.0.143.2:/remote-dir

Transfer local files to remote destination – using specific UDP port

ascp -l 100m -O 42000 /local-dir/files user@10.0.143.2:/remote-dir

Transfer files to remote shares destination – using server “10.0.143.2”

ascp /local-dir/files user@10.0.143.2:”/inbound/nyc/”

Using command-line operations

Figure 6-8. Command-line transfer: general examples

The example provides some sample command-line transfers. For more sample commands, see the *ascp General Examples* section of the *IBM Aspera HSTS Administration Guide*.

The following command initiates a transfer with a target rate of 100 Mbps, but authenticates with a login username that includes a space:

```
ascp -l 100m local-dir/files "User Name@10.0.0.2:/remote directory"
```

The following command transfers the contents of the local directory to the remote server and remove all of the transferred files (except the source directory) on the source system after the files are transferred:

```
ascp –remove-after-transfer local-dir asp1@10.0.0.2:/remote-dir
```

The following command transfers files by using a non-default UDP port:

```
ascp -l 100m -O 42000 /local-dir/files user@10.0.0.2:/remote-dir
```

The following command transfers files to network shares location (\\inbound\nyc), through the 10.0.143.2 server

```
ascp /local-dir/files  root@10.0.143.2:”//inbound/nyc”
```

IBM Training

Environment variables

Recognized by scripts

Set variable=value

ASPERA_DST_PASS=*password*	*Password for destination*
ASPERA_PROXY_PASS=*proxy_server_password*	*password for Aspera Proxy Server*
ASPERA_SCP_COOKIE=*cookie*	*Set cookie to be associated with transfers*
ASPERA_SCP_DOCROOT=*docroot*	*Set docroot (overrides value in aspera.conf)*
ASPERA_SCP_FILEPASS=*password*	*Set passphrase to encrypt/decrypt files*
ASPERA_SCP_KEY=*"- - - - BEGIN RSA PRIVATE KEY ...."*	
ASPERA_SCP_PASS=*password*	
ASPERA_SCP_TOKEN=*token*	
ASPERA_SRC_PASS=*password*	

Linux and Windows

Figure 6-9. Environment variables

If you have values that are used frequently, you can configure environmental variables in both Linux and Windows environments that ascp supports. The ascp command recognizes some (but not all) operating system variables.

IBM Training

What you learned

- The ***asuserdata –+*** command displays the available parameters for **aspera.conf** AND the associated ***asconfigurator*** syntax
- The ***asconfigurator*** utility is called by the Aspera GUI when changes are made through the GUI, and ensures the integrity of the **aspera.conf** file
- The ***asuserdata*** command can also be used to display configuration settings for **aspera.conf** category sections
- The ***ascp*** command provides a command-line utility for transferring files via FASP
- Environmental variables can be defined for use with the ***ascp*** command
- Extensive options are available when using the ***ascp*** command

Figure 6-10. What you learned

IBM Training

Unit summary

- Run the appropriate asuserdata command to print all possible aspera.conf entries and their associated asconfigurator commands
- Use asconfigurator utility to modify aspera.conf entries
- Transfer files and directories between Aspera servers using the ascp command

Figure 6-11. Unit summary

IBM Training

Review questions

1. True or False

   You can use ***asconfigurator*** on an IBM Aspera Transfer Server to define default connections to remote IBM Aspera servers

2. Which of the following examples of ***ascp*** commands uploads a directory to a remote system by using standard SSH encryption? Select all that apply:

   A. ascp user_x@remote.com:filey .
   B. ascp –T xyz/ user_x@remote.com:/
   C. ascp userx@remote.com:xyz/filey
   D. ascp xyz/ userx@remote.com/

Figure 6-12. Review questions

IBM Training

Review answers

1. True or False

   You can use ***asconfigurator*** on an IBM Aspera Transfer Server to define default connections to remote IBM Aspera servers

   The answer is False. While asconfigurator can be used to configure most Aspera parameters, pre-defined connections can be configured only with the Aspera GUI.

2. Which of the following examples of ***ascp*** commands uploads a directory to a remote system by using standard SSH encryption? Select all that apply:

   A. ascp user_x@remote.com:filey .
   B. ascp –T xyz/ user_x@remote.com:/
   C. ascp userx@remote.com:xyz/filey
   D. ascp xyz/ userx@remote.com/

   The answer is D

Figure 6-13. Review answers

IBM Training

Lab Exercise 5

In this lab, you use the **asconfigurator** command to modify the **aspera.conf** file rather than the Aspera GUI.
The second part of the exercise uses the ***ascp*** command with various options to perform transfers from the command line.
Numerous options for both commands are supported, so use the Administration Guide to determine the usage of some of the commands.

- Pay close attention to the use of quotation (“) marks and other options.
- Use the GUI to confirm the changes that you make with **asconfigurator**.

Figure 6-14. Lab Exercise 5

Unit 7. Configuring advanced features

Estimated time

02:00

Overview

This unit addresses several features that are not required for basic configuration of the IBM Aspera Transfer Server, but are commonly implemented on production systems

How you will check your progress

- Exercise

IBM Training

Unit objectives

- Configure IBM Aspera High-Speed Transfer Server to use custom SSL certificates and token authorization
- Outline the process of configuring HTTP Fallback
- Manipulate files using the Aspera Pre/Post feature
- Configure and manage Node API settings
- Distinguish between the hot folders and Aspera Watch Service
- Explain the procedure for implementing hot folders on Windows platforms
- Implement Aspera Watch Folders

Figure 7-1. Unit objectives

IBM Training

Authentication and authorization

Access via SSH or HTTPS?
- SSH best suited when all systems part of same administrative domain
- HTTP best suited for access with arbitrary clients or from internet at large

SSH Services Authentication
- *SSH user and password or user and key:* most common usage of SSH authentication
- *Authorization Token*: used by Aspera for authorization only – transfers to or from server only allowed if initiated with private web key protected by valid authorization token
- *SSH and Access key:* uses Aspera web private key but requires access key and secret for access to storage

HTTPS Services Authentication (Node API)
- *Node user and password* – basic authentication
- *Access key and secret* – basic authentication via Node API – returns access key for access
  - No docroot configured
  - Restriction required
  - Use with no defined storage fails

Figure 7-2. Authentication and authorization

Aspera supports different methods for authenticating and authorizing users when they need to access resources on the Aspera server. In the past, the primary method that was used for authenticating users was SSH, which requires either a username and password challenge or username and public key. More recently, Aspera supports the use of Node API and access key and secret authentication methods, which use different approaches for authentication and authorization.

The choice of which approach to use is up to your organization. In general, SSH authentication works best when all the computer systems are part of the same administrative domain, meaning they are typically within the same organization and administered by the same administrators.

HTTP (Node API) authentication is considered to be a better authentication method when access is from arbitrary client systems or primarily from the internet at-large.

SSH Authentication

SSH username and password or username and key: The most common usage of SSH authentication is connecting to the SSHD service and providing a valid username and password, or by providing a valid username along with a properly configured SSH private key. This method is easy to configure and provides a reasonable level of security (use of SSL keys is considered more secure than using a password). The ascp process is designed to contact the SSH service on a remote system to authenticate when a transfer is started, and can work with either form of SSH validation).

Authorization Token: While token based authentication is widely used in web-based applications, Aspera web applications do NOT always use tokens for authentication. Aspera web applications use tokens for authorization to allow a specific operation, for example, uploading or downloading a specific file. When a web application makes a request to upload or download data, the application server contacts the transfer server and requests a token on behalf of the user. The token is created and sent to the application server. The application server then sends the token to the user's browser, along with the IP address of the server to contact, and the name of the user account to use for authentication. The ascp process on the user's system contacts the transfer server's SSH service, and if properly authenticated, then presents the token it received along with the request to perform the transfer. The token defines the resources that the user can access, the direction of the transfer and other authorization details that specify what the user is allowed to do.

SSH and Access key: An access key is an API key, which is used to identify the source or user who made the request. Access keys are commonly used when access to a storage environment is needed and the environment is configured to support access keys to preserve the security of the storage environment.

HTTPS Services Authentication

Node user and Password: If users that are not part of the organization's administrative domain need access, or if unknown internet users need access, creating SSH credentials for every user might not be not feasible. So, the solution is to provide authentication on the web application with the username and password of a Node API account. The application server then authenticates with the transfer server on behalf of the user. Aspera uses this approach when a Shares or Faspex user requests transfers. The application server authenticates the user with login and password authentication. After the application server authenticates the user, it contacts the transfer server with the Node API credentials, and requests a token for the user to access the resource requested. In this case, the token is an authorization token.

Access key and Secret: It is also possible to configure Aspera transfer servers to use basic authentication that is combined with the use of an access key and secret to provide access to protected resources that require access keys. Aspera applications support access key and secret authentication when connecting to cloud environments.

IBM Training

Configure SSL/TLS certificate

Self-signed certificate (default)

/opt/aspera/etc/aspera.conf

```
<server>
        <server_name>server_IP_or_name</server_name>
        <http_port>9091</http_port>
        <https_port>9092</https_port>
        <enable_http>false</enable_http>
        <enable_https>true</enable_https>
        <cert_file>/opt/aspera/etc/aspera_server_cert.pem</cert_file>
</server>
```

/etc/init.d/asperanoded restart
Or
service asperanoded restart

Figure 7-3. Configure SSL/TLS certificate

By default, Aspera transfer servers use a self-signed certificate (aspera_server_cert.pem) located in the /opt/aspera/etc directory. If a non-certificate authority (CA)-signed certificate is acceptable, it is not necessary to make an entry in the aspera.conf file. However, if you are using a CA-signed certificate, you do need to make an appropriate entry in the aspera.conf file.

Configuring an Aspera Transfer Server to use your organization's CA-signed certificate involves modifying the transfer server's aspera.conf file to identify the location of the SSL certificate to be used.

Open the /opt/aspera/etc/aspera.conf file and locate the <server> section.

Add the <cert_file> tag and provide the full path name of the certificate file to be used. As mentioned, Aspera provides a self-signed certificate in the /opt/aspera/etc/aspera_server_cert.pem file, which is used by default. This certificate can be used if you do not have a CA-signed certificate file readily available (you are waiting for the CA assignment), or you do not require a CA signed certificate. If you require a CA assigned certificate file, ensure that a copy of the certificate file is on the transfer server, and include the full path name to it in aspera.conf.

After you modify the aspera.conf file, restart the node service by running the /etc/init.d/asperanoded restart command.

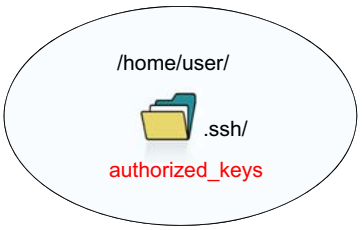
NOTE: Restarting the node service requires administrative rights (root).

For more information about certificates, see the *Authentication and Authorization* chapter of the *IBM Aspera High-Speed Transfer Server Administration Guide*.

IBM Training

Public keys: Transfer user on transfer server

MUST be configured for Shares and FASPex transfer users



1 Create .ssh directory in user's home directory
mkdir /home/shares/.ssh

2 Create "*authorized_keys"* file in .ssh directory
cat /opt/aspera/var/aspera_id_dsa.pub >> /home/shares/.ssh/authorized_keys

3 Change ownership and permissions of ".ssh" directory and "*authorized_keys"* file
chown –R shares:shares /home/shares/.ssh
chmod 700 /home/shares/.ssh
chmod 600 /home/shares/.ssh/authorized_keys

Verify public key authentication is enabled in **/etc/ssh/sshd_config** file
`PubkeyAuthentication yes`

Restart sshd – `sudo service sshd restart` *or* `sudo /etc/init.d/sshd restart`

Figure 7-4. Public keys: Transfer user on transfer server

If user authentication is to be performed by using public keys rather than passwords, it is necessary to configure a transfer user's account to use a recognized public key value. Users of both Shares and Faspex must be configured to support a public key. The example shows an entry for a Shares transfer user. When Faspex is installed on the transfer server, it automatically configures a transfer user with the name faspex. This transfer user account needs a public key file created the same way as for the shares user discussed previously.

The first step is to create a new directory called `.ssh` in the transfer user's home directory.

Next, you need to create a file called `authorized_keys` in the newly created `.ssh` directory.

The final step is to set the ownership of the `.ssh` directory and all files within it to match the transfer user account name. The `–R` option to the chown command indicates the recursive option, meaning that the chown command should affect the named directory and all files beneath it. The value `aspuser:aspusersgroup` references the specific transfer user's name and their group association.

In order for IBM Aspera HSTS to use public key authentication, you must modify the `/etc/ssh/sshd_config` file to ensure the PubkeyAthentication parameter is set to yes AND you restart the `sshd` service:

`sudo service sshd restart` ***OR*** `sudo /etc/init.d/sshd restart`

IBM Training

Configuring new RSA key pair

Create new RSA key pair on client system

ssh-keygen

Transfer key pair to remote server

ssh-copy-id

cat ~.ssh/id_rsa.put | ssh username@remote_system “mkdiir –p ~/.ssh && touch ~/.ssh/authorized_keys && chomod –R go=~/.ssh && cat >> ~/.ssh/authorized_keys”

Figure 7-5. Configuring new RSA key pair

If you are accessing the IBM Aspera transfer server from a client system other than using an IBM Aspera Transfer server as the client, you need to get the client’s public key to the IBM Aspera Transfer Server.

Generate RSA key pair

By default, ssh-keygen creates a 2048-bit RSA key pair, which is secure enough for most use cases (you can optionally pass in the -b 4096 flag to create a larger 4096-bit key). You are prompted to enter the file name to save the key (the default value is the login user’s home directory and the .ssh directory is created if it doesn’t exist).

You are also prompted to enter a pass phrase, which adds an extra layer of security to prevent unauthorized users from logging in. You now have a public and private key that you can use to authenticate. The next step is to place the public key on your server so that you can use SSH-key-based authentication to log in.

Copy public key to server

Copy by using ssh-copy-id

The quickest way to copy your public key to a Linux host is to use a utility called ssh-copy-id. Due to its simplicity, this method is recommended if available. If you do not have ssh-copy-id available to you on your client system, you can use one of the two alternative methods: copying the key

with password-based SSH, or manually copying the key and placing on the client system. The ssh-copy-id tool is included by default in many operating systems, so you might have it available on your local system. For this method to work, you need password-based SSH access to your server. To use this command, you specify the remote host that you would like to connect with and the user account that you have password SSH access to. This account is the account to which your public SSH key is copied.

Copy public key by using ssh

If you do not have ssh-copy-id available, but you can access the account on your server with password-based SSH access, you can upload your keys by using a conventional SSH method. Use the cat command to read the contents of the public SSH key on your local computer and pipe it that through an SSH connection to the remote server.

On the client system, you must confirm that the `~/.ssh` directory exists and has the correct permissions under the account you are using.

You then put the public key content into a file that is called `authorized_keys` within the `.ssh` directory.

Copy public key manually

If password-based SSH access to your server is not available, you need to complete the process manually.

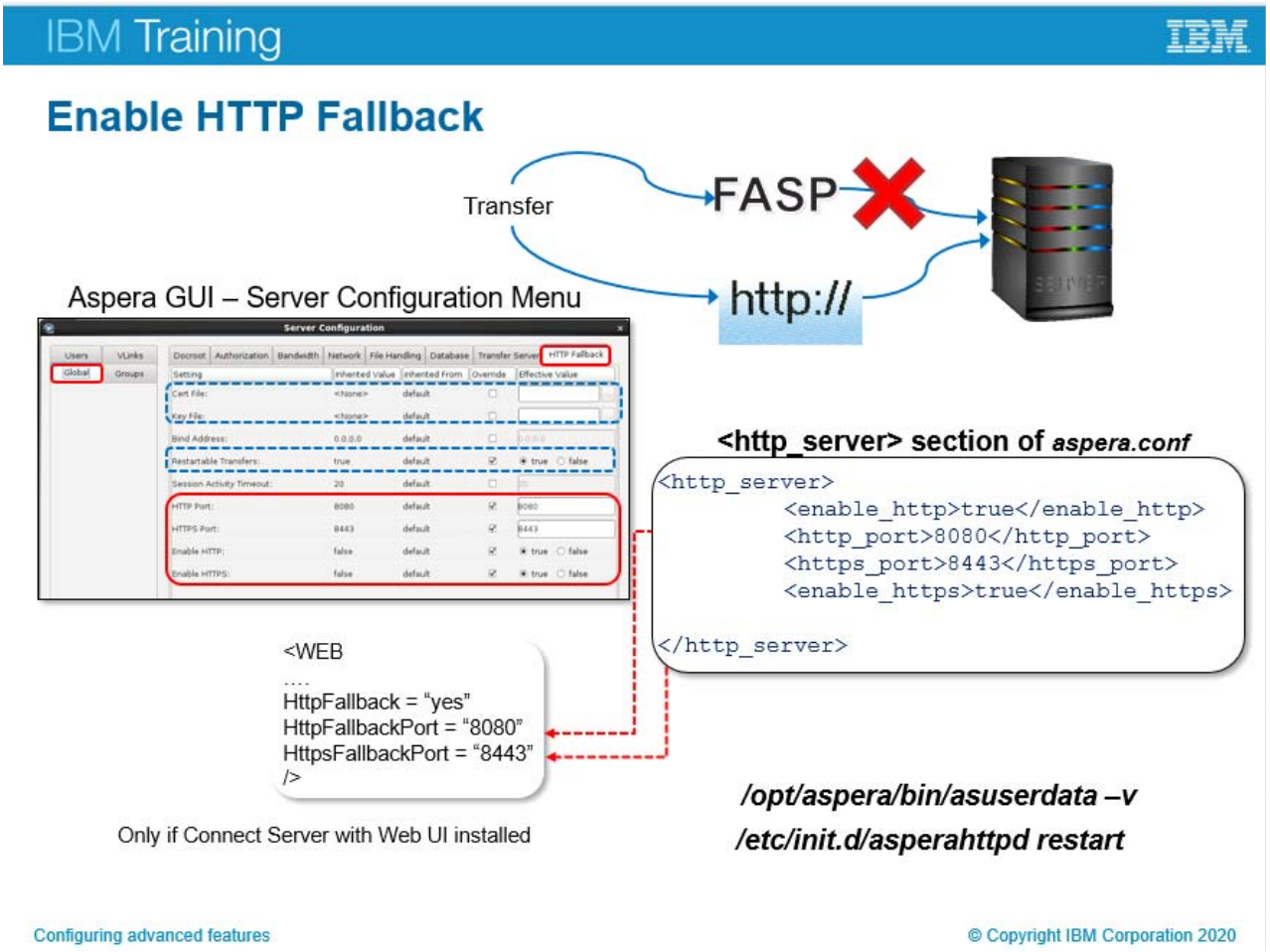


Figure 7-6. Enable HTTP Fallback

HTTP Fallback is a feature that provides a secondary transfer method when the internet connectivity that is required for Aspera transfers is unavailable. When HTTP Fallback is enabled, if a transfer cannot be completed via the configured UDP port (33001 by default), the transfer is attempted by using HTTP as the fallback method of transfer.

The HTTP Fallback service is not running on the standard HTTP ports (80 or 443). In fallback mode, the Aspera Connect client connects to the Aspera HTTP server on a port other than 80 or 443. The reason that the Aspera HTTP Fallback service is not run on port 80 is that a web application is already running on those ports; either the Aspera Transfer Server, or your own custom developed one.

If you decide to enable HTTP Fallback, you need to modify the `aspera.conf` file. These changes can be made with the Aspera GUI, or by directly editing the `aspera.conf` file.

Enable HTTP Fallback from Command Line

If you modify the `aspera.conf` file manually, you must locate the `<http_server>` section within aspera.conf file. If the section does not exist, create it.

Add the parameters to enable HTTP and HTTPS, and those entries that specify the ports to use for these services. You can use different port values, but the port values that are specified in the `<http_server>` section of aspera.conf must match the HttpFallbackPort and HttpsFallbackPort values you specified in the `<Web>` section.

Additionally, if the transfer server is configured with the Web UI, you need to enable HTTP Fallback by entering the variable HttpFallback in the <Web> section of the aspera.conf file and setting it to a value of yes. This entry is only needed if the server acts as a stand-alone system.

Don't forget to run /opt/aspera/bin/asuserdata –v to validate the changes you made to the aspera.conf file.

NOTE*:* After you change any of the settings in this section, remember to restart the AsperaHTTPD server (/etc/init.d/asperahttpd in Linux). In Windows environments, use the Services pane to restart the AsperaHTTPD service. On Linux systems, you run the /etc/init.d/asperahttpd restart command.

Enable HTTP Fallback from GUI

You can enable HTTP Fallback with the Aspera GUI. You need to start the GUI with administrative rights and go to the Server Configuration menu. Select the Global tab, then the HTTP Fallback tab to open the menu. Select the Enable HTTP and Enable HTTPS parameters and confirm that the ports are the ones you want to use. Click Apply to save your changes.

Other parameters are shown on the HTTP Fallback menu that you might want to modify. The Restartable Transfers option can be set to true to support ascp's ability to resume file transfers from the point of interruption. While a default action for most transfers, restarting transfers from where they left off when in HTTP Fallback mode must be explicitly enabled. You enable this function with the GUI or by manually adding the <restartable_transfer> parameter to the aspera.conf file and setting it to true.

The Cert File and Key File parameters can be set to identify specific SSL certificate and key files in HTTP Fallback situations. If your organization has certificates and keys, you can enter the path to those files in the GUI (you can also enter them manually). If you do not enter entries for these parameters, the system uses the default certificate and key files that are provided with the Aspera software that is installed on the server.

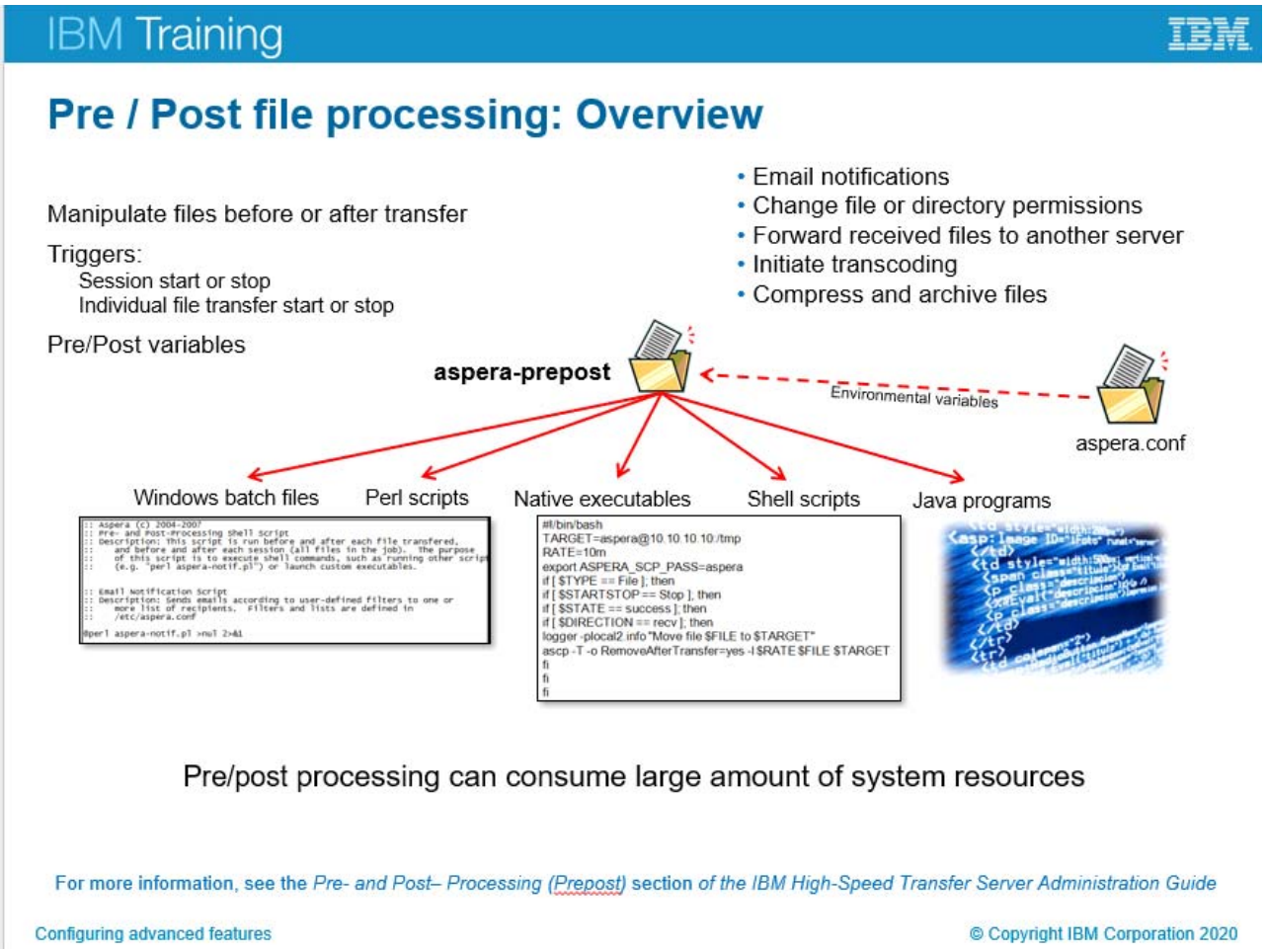


Figure 7-7. Pre / Post file processing: Overview

A powerful feature of IBM Aspera HSTS is the Prepost function. Prepost is the term that is used to describe the ability to manipulate files before or after transfers are completed. You can specify whether the action should be performed on each file individually, or on all files involved in the session.

The prepost feature is implemented in an executable file that is created on the server named aspera-prepost (or aspera-prepost.bat on Windows servers). The file's location is operating system dependent. On Windows servers, the file is located in the `C:\Program Files (x86)\Aspera\Enterprise Server\var` directory as a batch file, while Linux systems store this file in the `/opt/aspera/var` directory. The aspera-prepost file is a script that can also call other routines, such as other batch files, Perl scripts, native executable programs, Linux shell scripts, or Java programs.

The prepost feature recognizes numerous standard variables that can be referenced in scripts that are configured in the aspera.conf file.

The prepost feature can be used for all kinds of file manipulations. The built-in email notification function is an example of the prepost function.

The following pages discuss the procedure for configuring Pre/Post processing, but more details of the pre/post feature are available in the Pre- and Post– Processing (Prepost) section of the *IBM Aspera HSTS Administration Guide*.

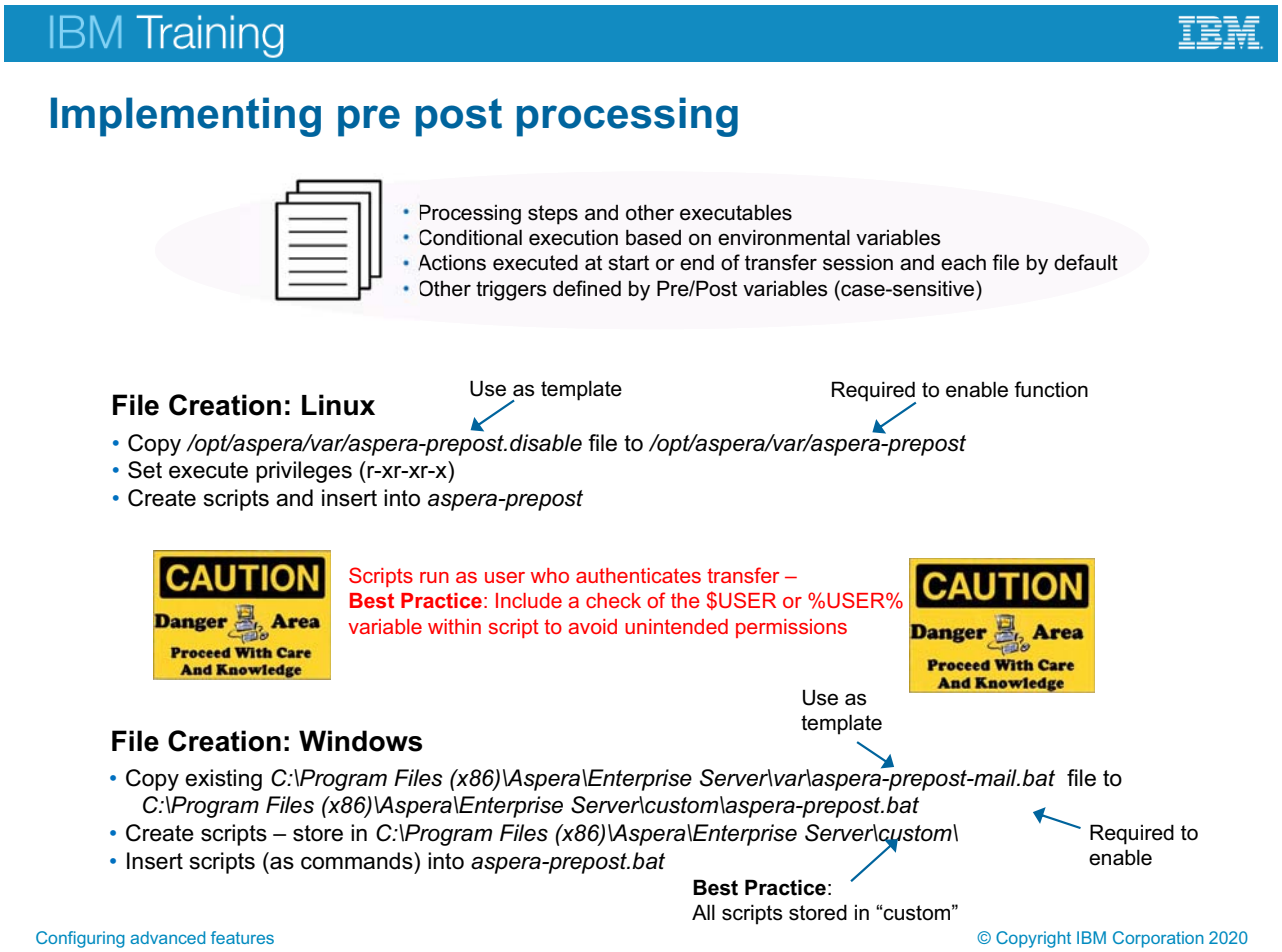


Figure 7-8. Implementing pre post processing

The aspera-prepost file does NOT exist by default on either Windows or Linux systems, so must be created to enable Pre/Post function. A batch or shell script file is provided by default, which can be used as the foundation of your customized aspera-prepost file. This sample file is named `/opt/aspera/var/aspera-prepost.disable` on Linux systems and `C:\Program Files (x86)\Aspera\Enterprise Server\var\aspera-prepost-email.bat` on Windows servers. While named differently, both files include a call to a Perl program (`aspera-notif.pl`) which causes email messages to be sent to a list of recipients after the defined actions are completed.

NOTE: Even though the perl command is entered in the template file, it is not enabled on the system until the actual *aspera-prepost* file is created and entries are made in the aspera.conf file.

Copy the appropriate aspera-prepost file to a new file (`/opt/aspera/var/aspera-prepost` on Linux or `C:\Program files (x86)\aspera\Enterprise Server\custom\aspera-prepost.bat`) which you edit to include any scripts or programs you want to run before or after file transfers.

NOTE: The `C:\Program Files (x86)\Aspera\Enterprise Server\custom` directory exists by default. An Aspera Best Practice on Windows servers is to create the aspera-prepost file in the `C:\Program Files (x86)\Aspera\Enterprise Server\custom`. Any batch files that you include the aspera-prepost script file should be placed in this directory.

NOTE: Set read and execute permissions of the aspera-prepost file on Linux systems. The aspera-prepost processing script contains the details of the actual processing steps (the actual

lines of scripting code to perform an action). But, it is better to create the script files as individual files and call them from within the aspera-prepost file. You can also include other programs and routines to be run by listing them within this file.

It is common for the aspera-prepost script to check for various conditions (based on environmental variables), trigger external executables based on the condition (discussed later in this module).

CAUTION: Any script run as a result of an entry in an active aspera-prepost file runs with the same permissions as the user who authenticated the transfer! Therefore, it is important to confirm that each script called is confirmed to perform exactly as expected before adding the script to the aspera-prepost file. An Aspera Best Practice recommendation is to include a check of the $USER or %USER% variable within the script to confirm what user permissions are being used when the script is run.

IBM Training

The aspera-prepost file

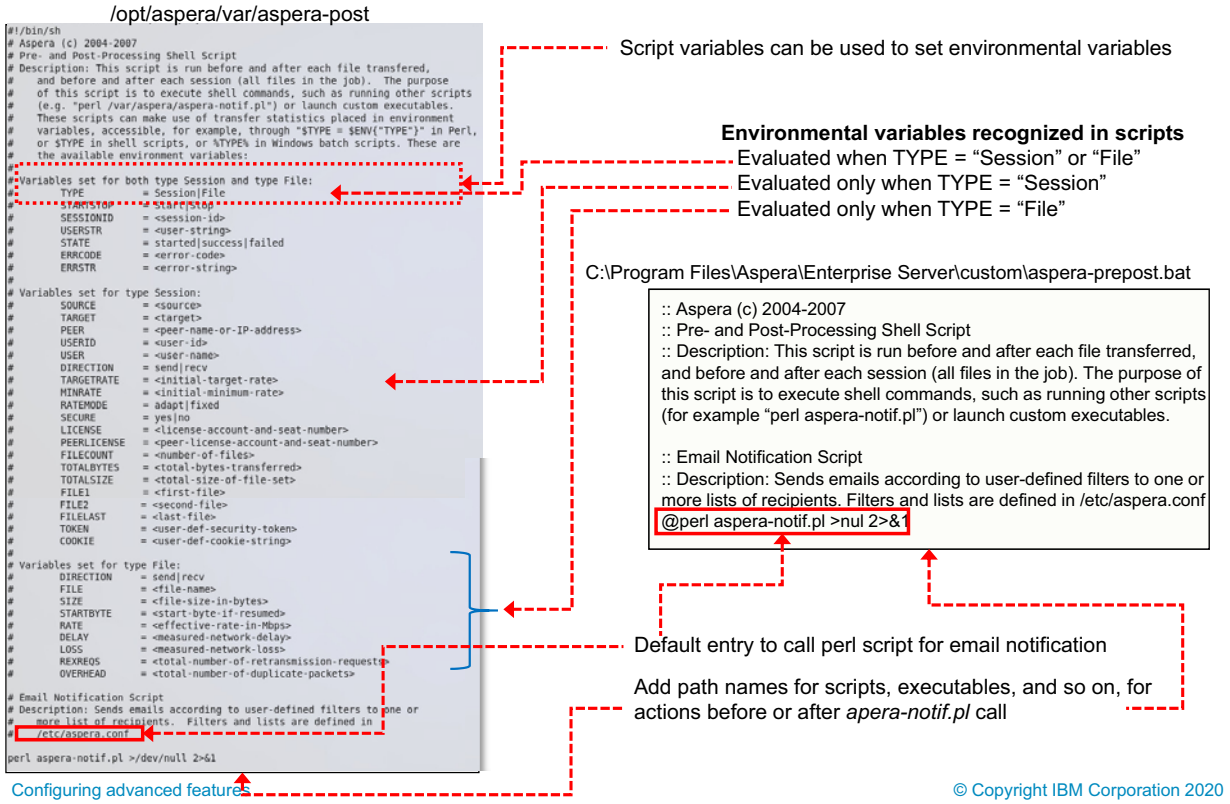


Figure 7-9. The aspera-prepost file

The actual content of your initial aspera-prepost file (the copy you made from the `/opt/aspera/var/aspera-prepost.disable` file on Linux systems or `C:\Program Files (x86)\Aspera\Enterprise Server\var\aspera-prepost-email.bat` file on Windows systems depends upon operating system. The Linux file identifies some of the environmental variables that can be used within scripts, while the Windows version doe not.

NOTE: Not all valid variable names are presented in the Linux file. For a complete listing of variables that can be used within scripts, and examples of their format requirements, see the Pre/Post Variables section of the *IBM Aspera HSTS Administration Guide*.

IBM Training

Prepost script examples

Change permissions after receiving and creating log entry

```
#!/bin/bash
if [ $TYPE == File ]; then
    if [ $STARTSTOP == Stop ]; then
    echo “The file is: $FILE” >> /tmp/p.log
    chmod 777 $FILE
    fi
fi
```

Linux shell script

Windows batch files deployment

Create scripts with .bat extension in
C:\Program Files (x86)\Aspera\Enterprise Server\custom
Include script name in aspera-prepost.bat

Send notification when files larger than 1 GB

```
set FILESIZE=1073741824
if  “%TYPE%” == “Session” (
    if “%STARTSTOP%” == “Stop” (
        if %TOTALSIZE% GEQ %FILESIZE% (
        “C:\Perl\bin\perl.exe” aspera-notif.p
        )
    )
)
```

Active Perl software might need to be installed

Windows batch file

Forward files to another computer then remove

```
#!/bin/bash
TARGET=aspera@10.0.143.2:/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 –plocal2.info “Move file $FILE to $TARGET”
            ascp –T –o RemoveAfterTransfer=yes –l $RATE $FILE
$TARGET
            fi
        fi
    fi
fi
```

This script requires the remote server’s host key be cached before execution

Linux shell script

Linux scripts deployment

Create shell script(s)
Make script file executable
Include script path name in aspera-prepost

See “Pre- and Post-Processing” section of Admin Guide for more examples

Configuring advanced features

Figure 7-10. Prepost script examples

Linux shell scripts and Windows batch files can be created as separate files and then referenced in the appropriate aspera-prepost file. These files perform the identified tasks when triggered by the completion of a transfer (indicated by a value of Stop value stored in the STARTSTOP variable).

Notice the Windows batch file example calls perl to run the `aspera-notif.pl` script. By default, perl software is not installed on Windows servers, so it must be added if you want to create and run perl scripts

NOTE: If Perl is not already installed on your Windows server and you want to use perl scripts in your pre/post processing, you can download and install Active Perl software from the following URL: www.activestate.com/store/activeper/download

The default script referenced in the `C:\Program Files (x86)\Aspera\Enterprise Server\var\aspera-prepost-email.bat` assumes that perl software is available on the server. This script is responsible for generating email notifications about file processing. So, even if you do not add other custom scripts, you must install perl software to use the function of the built-in email notification script.

The pre/post processing function recognizes several default variables, which are referenced within an aspera-prepost script to manage conditional situations. See the *Pre/Post Variables* section of the *IBM Aspera HSTS Administration Guide* for a listing of valid variables.

Pre/post variables are case-sensitive, so it is important to use the correct case when referencing them.

IBM Training

Built-in pre/post email notification

- Perl software must be installed
- *aspera-prepost* file must be implemented
- Add to **aspera.conf** for configuration

Email notification requires SMTP server
Reachable on network
Cannot use any external authentication or SSL

```
<CONF version="2">
  ...
  <EMAILNOTIF>
   <MAILLISTS
     mylist = "asperausers@training.com, admin@acme.com"
     myadminlist = "admin@acme.com"
   />

   <FILTER
     MAILLISTS = "mylist"
     TARGETDIR = "/content/users"
   />

   <MAILCONF
     DEBUG = "0"
     FROM = "asperaserver@training.com"
     MAILSERVER = "mail.acme.com"
     SUBJECT = "Transfer %{SOURCE} %{TARGET} - %{STATE}"
     BODYTEXT =
      "Aspera transfer: %{STATE}%{NEWLINE}%{TOTALBYTES}
bytes in
      %{FILECOUNT} files: %{FILE1}, %{FILE2}, ...%{FILELAST}."
   />
  </EMAILNOTIF>
</CONF>
```

Who is notified
Lists all users who receive email
Multiple lists can be used with multiple filters

Set filter variable values
Define conditional filters
Multiple filters allowed

Basic notification functions
"SUBJECT" & "BODYTEXT" values can reference processing variables - %(variable)

Configuring advanced features

Figure 7-11. Built-in pre/post email notification

As mentioned on the previous page, IBM Aspera HSTS also provides a built-in pre/post processing application that sends customized email notifications on transfer events. The call to start this application is included in the template file you use to create your final aspera-prepost file (perl `aspera-notif.pl`). This call is also included in the working copy of the aspera-prepost file you create.

In order for email notifications to be sent based on pre/post processing results, you must ensure that the perl software is available on the Aspera Transfer Server. However, to run this perl script routine, you must configure the aspera-prepost processing file. You must also add entries to the aspera.conf file to configure the basic details.

<EMAILNOTIF> Section

You need to create the entire `<EMAILNOTIF>` section in the aspera.conf file, as it does not exist by default. All other pre or post-processing data is entered within the opening and closing tags of this section.

<FILTER> Section

The variables that are defined in the `<FILTER>` section identify values for making conditional decisions. For example, the `MAILLISTS` conditional filter is set equal to the text string mylist, which in turn is expanded to identify a list of user email addresses (asperausers@training.com and admin@acme.com) in the `<MAILLISTS>` section. Notice that another value (myadminlist) is also

listed in the <MAILLISTS> section, with different email addresses. Other entries can reset the MAILLISTS variable to reference myadminlist when certain conditions occur.

<MAILCONF> Section

The entries under this section define the general email configuration that is required to successfully send emails, and what to include in the message of the email. Notice that the SUBJECT and BODYTEXT variables can reference other variables whose values are inserted into the message. For example, the SUBJECT variable references the SOURCE, TARGET, and STATE variables, all of which are referenced and populated when the aspera-notif.pl script is run.

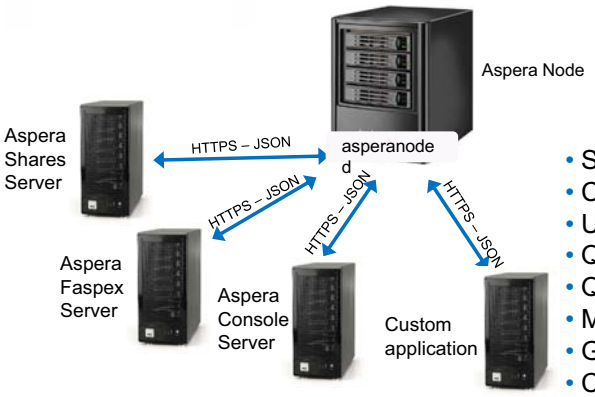
IBM Training

Aspera Node API: Overview

RESTful interface for full programmatic control of server environment

Daemon process (asperanoded) provides node-specific services

- HTTPS and HTTP interfaces
- JSON format API
- Authenticated API with own application-level users (node users)
- Node admin utility for node user and password management



- Send transfers over HTTP or HTTPS
- Create and delete files and directories
- Upload and download files and directories to or from serve
- Query what files and directories are on server
- Query what transfers were made to or from server
- Modify attributes of in-progress transfers
- Get transfer events from server
- Create access keys for permissions

Figure 7-12. Aspera Node API: Overview

As Aspera products evolved from their original command-line interface to web-based applications that interact with multiple transfer servers, the need arose for a standard method of interacting with those transfer server's file systems.

A node or a transfer server is a method of referring to the traditional Aspera Transfer Server from the perspective of web applications such as Shares or Faspex. Web applications like these can interact with multiple transfer servers, so instead of being limited to a single system, a Shares or Faspex instance can interact with multiple Aspera servers and their file systems.

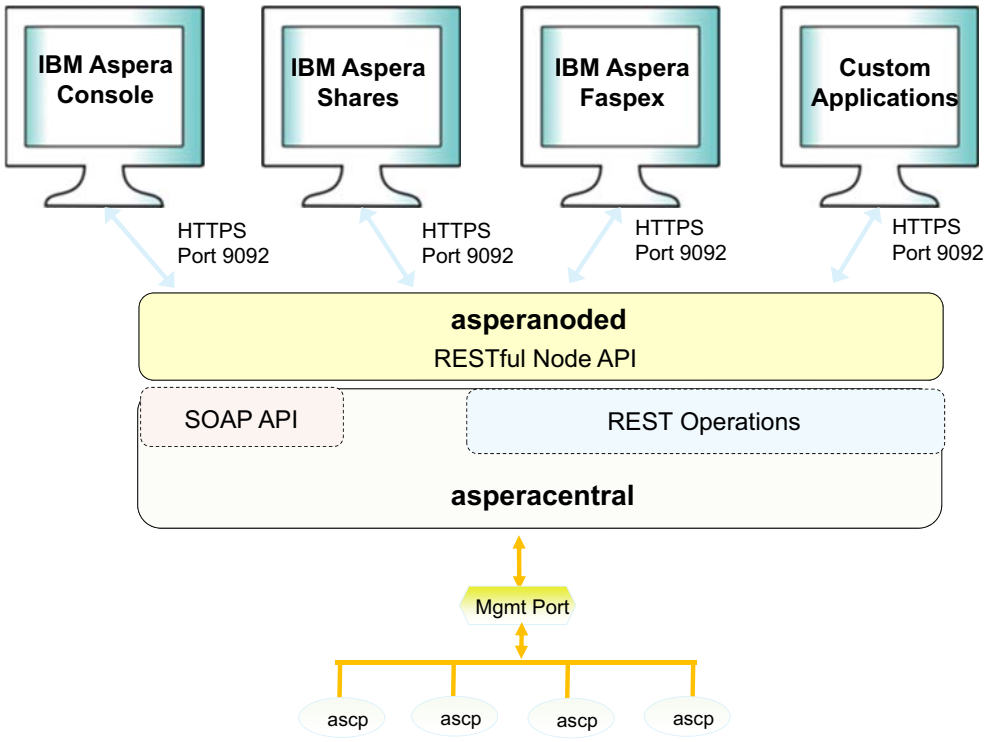
The Aspera Node server provides this method through the Node API. The web applications can use this API for a standard method of connecting to transfer servers without having to create different configurations for different servers.

The Node API can be accessed via HTTP or HTTPS (Aspera suggests only HTTPS is configured). The API is an authenticated API, supporting its own application-level users, which are maintained in an application-specific database. Users and their associated passwords are managed with the `asnodeadmin` command, included with IBM Aspera HSTS software.

Numerous tasks can be performed remotely via the API.

IBM Training

Aspera Central & Node API



Configuring advanced features

Figure 7-13. Aspera Central & Node API

The asperacentral process provides a management port for all ascp processes on the server, making it possible to manage all ascp processes running on the server.

It offers a web-services interface that can be used to start, monitor, and control transfers and query and modify the node's configuration. Information about transfers on the server is available to asperacentral, regardless of whether the transfers are started through the web services or initiated by another application or command-line operation on the local system.

Node API is a Representational State Transfer (REST) API implemented as a daemon process, asperanoded. The asperanoded process provides secure access to various system and file operations on the node to retrieve the needed information. Details about the functions available through the Node API are discussed later in this module.
As an alternative to the REST API, asperacentral can also provide the same information via a Simple Object Access Protocol (SOAP) API. The asperacentral process can log events directly to an external MySQL database by using the `dblogger` routine. The SOAP interface and the use of `dblogger` is still available to support older versions of transfer server software that do not support Node API.

IBM Training

Node API REST endpoints

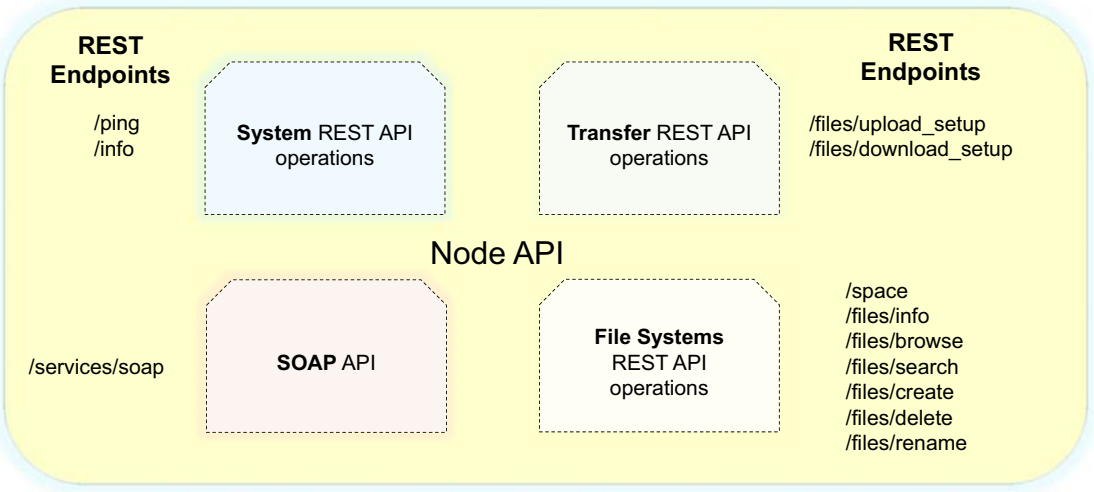


Figure 7-14. Node API REST endpoints

The Node API can be used to ping and query information from Aspera nodes and perform various system and file operations on the node. The Node API provides secure access to a wide range of actions and is RESTful, which means that HTTP requests are used to interact with resources within a targeted Aspera node. Classification of operations available through the Node API, include:

System operations: Endpoints to verify that the transfer server is reachable and provide basic information about the transfer server.

Transfer operations: Endpoints the setup the system for uploads and downloads.

File Systems operations: Endpoints that facilitate browsing and manipulating the file system of the transfer server.

SOAP services: Endpoint that connects with the SOAP interface that supports XML-based messaging.

IBM Training

Typical Node API functions

System REST API operations

/ping
Is the node running?

/info
Software version?
System time on the node?
Node ID?

File Systems REST API operations

/space
Total bytes in directory?
Free space in directory?
System time on the node?

/files/browse
Browse a directory

/files/create
Create files and directories

/files/delete
Delete files and directories

/files/search
Search file system using filters

/files/rename
Rename files and directories

Transfer REST API operations

/files/upload_setup
Get information needed for upload

/files/download_setup
Get information needed for download

/transfers
List node-to-node transfers
Create node-to-node transfer
Resume or stop transfer

SOAP API

Initiate transfers
Monitor transfers
Control current transfers

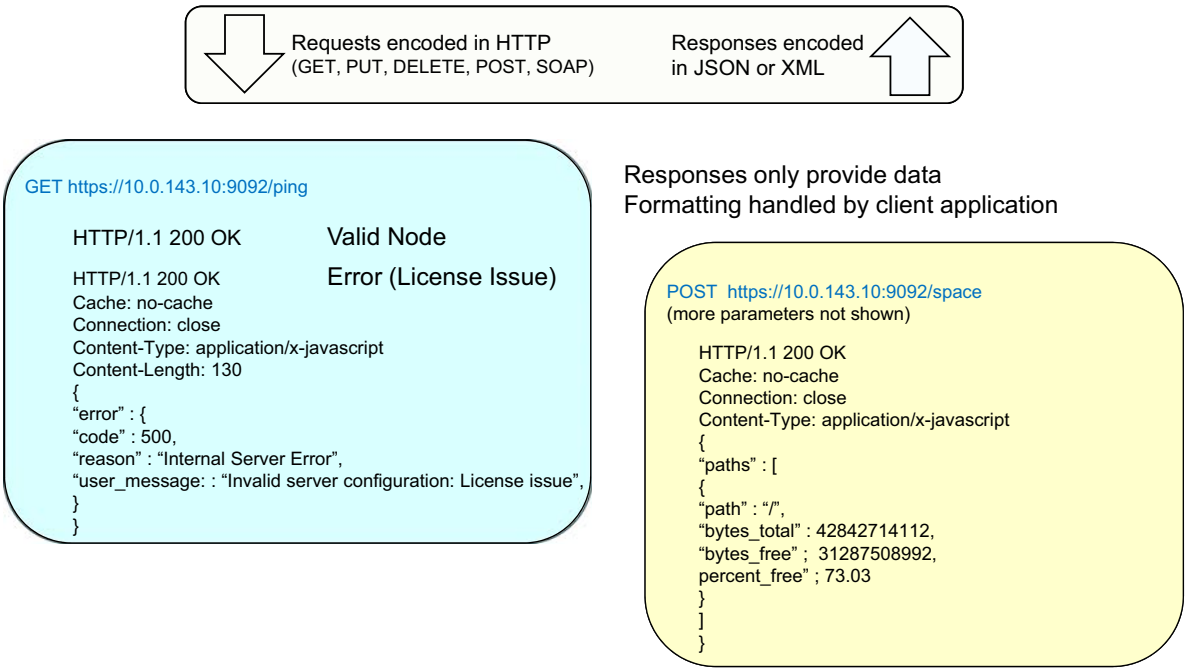
Figure 7-15. Typical Node API functions

The Node API is used by developers and system administrators to query nodes and manage transfers. The example that is shown indicates the kinds of tasks that can be performed by referencing these endpoints. Only the endpoint references are shown, not the actual requests. But those requests typically include HTTP verbs (GET, POST, DELETE, PUT), login credentials, filters, and possibly arguments. Details about fully developed requests are presented in a separate module of this training course.

The included SOAP API is not RESTful, but does support SOAP requests to the transfer server. The response to SOAP requests are provided in XML format.

IBM Training

Node API requests and responses



Configuring advanced features

Figure 7-16. Node API requests and responses

Basic authentication is implemented in Node API, in which the calling application authenticates itself using an encoded username and password, which is typically included within the HTTP request, as defined by RFC 1945.

The Node API supports standard HTTP requests (GET, POST, DELETE, PUT); GET is the recommended method to retrieve information from a resource and POST is recommended for updating or modifying an entity. Details about using these requests are addressed in a separate module of this training course.

Responses are always in JavaScript Object Notation (JSON) format, unless the request was made by using SOAP, in which case the response is in XML format. The response does not define how the data is displayed to the requesting user. The web application that generated the request is responsible for formatting of the response. If a request is successful, Node API responds with an HTTP status code of 200. If errors are found when handling the request, the server returns a formatted JSON error message that contains the error code and a detailed explanation of the error.

Node API calls can be run from the command line with the `curl` command. `Curl` is free, open software that runs under various operating systems, providing a command-line tool and a library for transferring data that uses URL syntax. `Curl` supports standard HTTP or HTTPS GET, POST, DELETE, and PUT functions.

A complete description of the IBM Aspera Node API and how to use it for application development or from the command line is available on the Aspera Developer Network.

IBM Training

Enable Node API in aspera.conf

- Must be configured for web applications (Shares/Faspex/Console)
- Must be configured when tethering on-premises node to Aspera on Cloud
- Special permissions can be assigned with "– - acl-set" option
- Configured by FASPex installation script (if installed on transfer server system)

```
<central_server>
    <port>40001</port>
    <address>127.0.0.1</address>
    <persistent_store>enable</persistent_store>
</central_server>
<server>
    <server_name>server_IP_or_name</server_name>
    <http_port>9091</http_port>
    <https_port>9092</https_port>
    <enable_http>false</enable_http>
    <enable_https>true</enable_https>
</server>
```

service asperacentral restart
or
/etc/init.d/asperacentral restart

service asperanoded restart
or
/etc/init.d/asperanoded restart

Value must be "true"

Ports shown represent default values – can be set to any value

asconfigurator –x "set_server_data;server_name,<FQDN or IP address>
asconfigurator –x "set_server_data;http_port,9091;https_port,9092"
asconfigurator –x "set_server_data;enable_https,true;enable_http,false"

Configuring advanced features

Figure 7-17. Enable Node API in aspera.conf

Aspera web applications use the Node API to communicate with the transfer servers, even if the Aspera application is installed on the transfer server. Entries in the `aspera.conf` file enable the Node API service. Transfer user accounts that are used by the applications require more entries, which are discussed in the other training modules that are related to the specific applications.

In the `<central_server>` section, create the `<persistent_store>` entry and set its value to enable. This setting causes the retention of historical transfer data that is used by the stats collector.

In the `<server>` section, locate 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.

Continuing in the `<server>` section, configure the http and https ports as shown.

After the `aspera.conf` file is updated, it is necessary to restart both the asperacentral and asperanoded services:

`sudo service asperacentral restart` ***OR*** `sudo /etc/init.d/asperacentral restart`

IBM Training

Manage Node API (each server)

Must be configured for Shares and Console deployments

Automatically created during FASPex installation (if on Connect Server system)

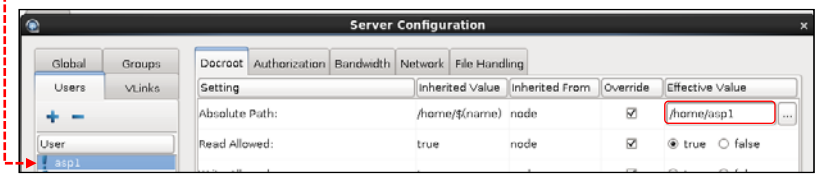
Create Node API user

 # opt/aspera/bin/asnodeadmin –a –u node_user – p aspera –x asp1

 # /opt/aspera/bin/asnodeadmin -l

Transfer user is local – operating system user

```
List of node user(s):
                user    system/transfer user    acls
==================  ======================  ==========
           node_user                    asp1  []
```



List options
asnodeadmin -h

Backup redis database
asnodeadmin -b /filepath/database.backup

Restore redis database
asnodeadmin -r /filepath/database.backup

See “Node Admin Tool” section of Admin Guide for more details

Figure 7-18. Manage Node API (each server)

Aspera web applications authenticate with the remote node service by using a Node API username and password.

Setting up each transfer server’s Node API is a process of creating a node user. The node user is mapped to a local transfer user that is configured in the Aspera database. The transfer user must be a local operating system account, which is used to authenticate the connection for the actual transfers. You can select an existing user account, or you can create an account unique for the web application. This transfer user account is used to authenticate the actual ascp transfer, and to set the node user’s docroot directory.

Aspera provides the `asnodeadmin` utility for managing the servers node users. You can use `asnodeadmin` utility to add, modify, delete, and list node users. For each node user you must indicate the following information:

- Node username
- Node user password
- Transfer username

The example creates a node user named `node_user` with a password of aspera, and the associated transfer user is asp1.

Running asnodeadmin with the –l argument lists the currently defined node user accounts and their associated transfer user. It also indicates any ACLs (Access Control Lists) or other local file system attributes that are specified.

NOTE: You can use the `--acl-add` option to add an ACL for a user. Use the `--acl-set` to set ACLs for a user. You can use the `--acl-del` option to delete ACLs for a user when running asnodeadmin to modify a node user.

The transfer user that you associate with the node user account must be entered into the Aspera configuration as an authorized Aspera user on the server. You can add a user from within the Aspera GUI, or by editing the `aspera.conf` file. It is also required that you specify a docroot for the transfer user account.

The asnodeadmin command is used to perform several other functions, including deleting or modifying node users, changing passwords, and more. You can list the supported options by running the asnodeadmin command with the –h option. For more information about asnodeadmin, see the Node Admin Tool section of the *IBM Aspera HSTS Administration Guide*.

Node user accounts are not defined in the aspera.conf file like transfer user accounts. Instead, node user data is stored in a redis database, which can be backed up and restored by using the asnodeadmin command:

Windows Systems

```
asnodeadmin –b C:\filepath\database.backup
asnodeadmin –r C:\filepath\database.backup
```

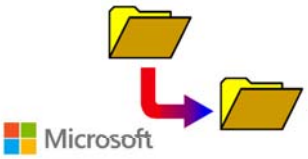
Linux Systems

```
/opt/aspera/bin/asnodeadmin –b /filepath/database.backup
/opt/aspera/bin/asnodeadmin –r /filepath/database.backup
```

IBM Training

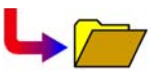
Automating transfers: Overview

Hot Folders (Windows only)
One-way replication or forwarding files in workflow
Scans file system for changes



Aspera Watch Service
File system change and detection service
Optimized for speed, scale, and distributed sources

Aspera Watch folders
Large-scale automated transfers
Ultra-large directories w over 10 million items and “growing” files



Configuring advanced features

Figure 7-19. Automating transfers: Overview

Hot Folders

An automatic file delivery feature called Hot Folders is available when the IBM Aspera HSTS software is running on a Windows system. Hot Folders are used to monitor local or remote folders for changes and automatically transfer new or modified files. Hot Folders can be used for one-way replication between two locations, or as a way of forwarding files in your workflow.

A Hot Folder can download from a server (pull) or upload to a server (push). Files and folders added to or modified within a Hot Folder on the source are automatically sent to the destination folder. Files that are deleted from the source are NOT deleted on the destination.

The Hot Folder feature does have some limitations, discussed later in this module, that might impact the decision to use the feature.

Watch Folders and Aspera Watch Service

All IBM Aspera HSTS systems include features that provide the ability for monitoring file system changes and automatically transferring new and modified files. Watch Folders and the Aspera Watch Service are designed to enable large scales automated file and directory transfers, including ultra-large directories with over 10 million items and directories with growing files.

Watch Folders use input from the Aspera Watch Service to automate transfers of files that are added or modified in a source folder. They can be configured to push from the local server or to pull

from a remote server. Remote servers can be an IBM Aspera HST Server, IBM Aspera HST Endpoint, or IBM Aspera Shares servers. Push Watch Folders can use IBM Aspera on Cloud and IBM Aspera Transfer Cluster Manager nodes for a destination.

The Aspera Watch Service creates and compares snapshots of the file system to determine changes rather than scanning the entire file system, thus significantly improving response time. Details about Aspera Watch Service are discussed later in this module.

IBM Training

Configuring Hot Folders

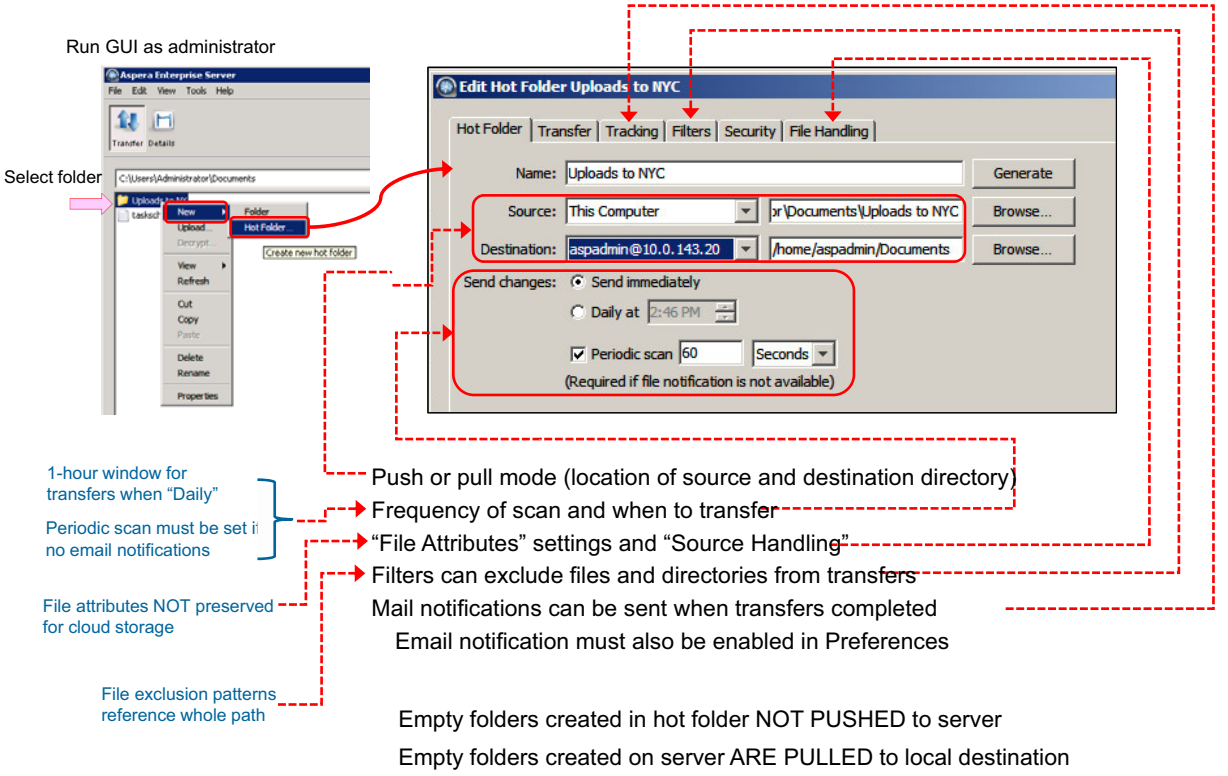


Figure 7-20. Configuring Hot Folders

Hot Folders can be implemented to monitor specific directories for changes, and then automatically transfer new or modified files to a designated destination. This feature can be used for one-way replication between two systems, or as an automated function for forwarding files in a workflow.

Administrators configure Hot folders with the Aspera GUI. Right mouse click the local directory and select the Hot Folder option to open the New Hot Folder page. Tabs located across the top of the page open more pages where configuration parameters are located.

Hot Folder Tab

This page is where the details of the hot folder functions are defined. The Source and Destination fields indicate whether files are pushed from the local system to the remote system, or pulled from the remote system to the local system. If the Source field is set to This Computer, then the Hot Folder is in push mode. If the Destination field is set to This Computer, then the folder is in pull mode.

NOTE: Only Aspera systems configured under Connections are available for either the Source or Destination fields

You can specify when files should be transferred in the Send Changes section:

If the hot folder is in Push mode, the Send immediately option can be selected to send new files as soon as they are placed in the folder, or existing files are modified. This setting keeps the

destination directory up to date with any new or modified files that are placed in the local source hot folder.

You can also configure transfers to occur at a specific time of day, regardless of the hot folder's mode of transfer (Push or Pull) by selecting the Daily at field and specifying a time. When the specified time is reached, transfers from the hot folder are allowed for 1 hour, including any new files that are added to the directory or retries required during that 1 hour window.

If the hot folder is in Pull mode, select the Every field and specify the time interval that is used for scanning the source and receiving files.

The Periodic scan field can be configured to specify how often the local source directory is scanned (in seconds, minutes, or hours) for new or modified files, and the identified files transferred.

NOTE: If file notification is not available, you must select the Periodic scan option to detect file changes.

Transfer Tab

As with all FASP transfers, you can specify your preferences for Transfer Policy and speed settings (Target Rate) by using the page that is displayed from the Transfer tab. You can also specify a Minimum Rate value, but this value should usually be left at the default value of zero.

Tracking Tab

The Tracking tab opens a page where you can enable and then configure email notifications that are sent when a transfer is completed.

IMPORTANT: The GUI must remain open in order for Hot Folder email notifications to be sent.

NOTE: Even though you enable email notifications on this page, you also need to enable them in the Preferences setting in order for the feature to function.

Filters Tab

The Filters tab opens a page that where you can specify patterns that prevent matching files or directories from being transferred. This feature is useful to avoid transferring files that you want to keep local, but store in the same directory as other hot files.

The excluded patterns are compared with the whole path, not just the file, or directory name.

The asterisk (*****) can be used to match zero or more characters in a string:

Directories: *mydir matches any directory with the name mydir; for example, /home/bob/mydir or /home/sally/mydir.

Characters: *2017 matches any file or directory name that ends with the characters 2017; for example, `/home/bob/sales_2017` or `/home/bob/lost_2017`.

Files: **update* matches the entire path name of the file named update; for example, `/home/abc/update`) and /update.

NOTE: Files used by FASP to resume incomplete transfers are automatically ignored based on the resume suffix (`.aspx` by default, but can be any value that is assigned for this purpose).

Security Tab

This page is where you enable file encryption: in transit, at rest, or both.

File Handling

The File Handling page can be used to configure several features that affect transferred data.

Resume Section

The first section of the page is Resume, which defines whether and how transfers are to be resumed if disrupted. The `Resume incomplete files` checkbox must be marked to enable this function. If this feature is not enabled, interrupted transfers are not completed, if restarted, they are transferred in entirety, rather than starting at the point where the transfer was stopped.

This section also provides a place to indicate how you want the system to determine the integrity of the partially completed transfer file before resuming: The `When checking files for differences` value can be set to the following values:

Compare file attributes: Compares sizes of the existing and original file and if the same, resumes the transfer, but if different, transfers the original file again.

Compare sparse file Checksums: performs a “sparse checksum” calculation on the received file and resumes the transfer if the results match the original file.

Compare full file checksums: Performs a full file checksum on the transferred file and resumes the transfer only if the calculation on the received file matches the checksum of the original file.

NOTE: Selecting the Compare full file checksums value does affect the transfer rate.

The `When a complete file exists at the destination` field can be configured to use an overwrite rule when a source file has the same name as a file in the destination directory. The default behavior is to always overwrite existing files, but you can select other options:

Always overwrite (default): no comparison is considered and exiting files or directories are always overwritten.

- *Overwrite if the source is different:* Overwrite the existing file if the source file is different than the existing file, even if the source file is older than the existing one.
- *Overwrite if the source is newer:* Overwrite the existing file if the source file is newer than the existing one, regardless if the file is different or not.
- *Overwrite if the source is both different and newer:* Overwrite the existing file only when the source file is different and newer than the existing file.
- *Never overwrite:* Keep the existing file and create a new file.

File Attributes

Configures how file attributes of access time, modification time, and source access time are preserved on the destination file.

NOTE: These times cannot be preserved for node or Shares connections that are using cloud storage.

Source Handling

This section is where you can indicate how you want to deal with source files that are successfully transferred. The options available are different for a hot folder that is configured for Push mode (local source file) or Pull mode (source directory is on the remote system).

Push Mode Options:

- A*utomatically delete source file after transfer:* Deletes the source files from the source directory.

- *Automatically move uploaded source files to a directory after transfer*: Do not delete the transferred files. Move them to an existing directory on the local system (you are prompted for the name of the directory where the files are placed).
- *Delete empty source subdirectories*: Remove any empty directories after their files are deleted or moved.

Pull Mode Options:

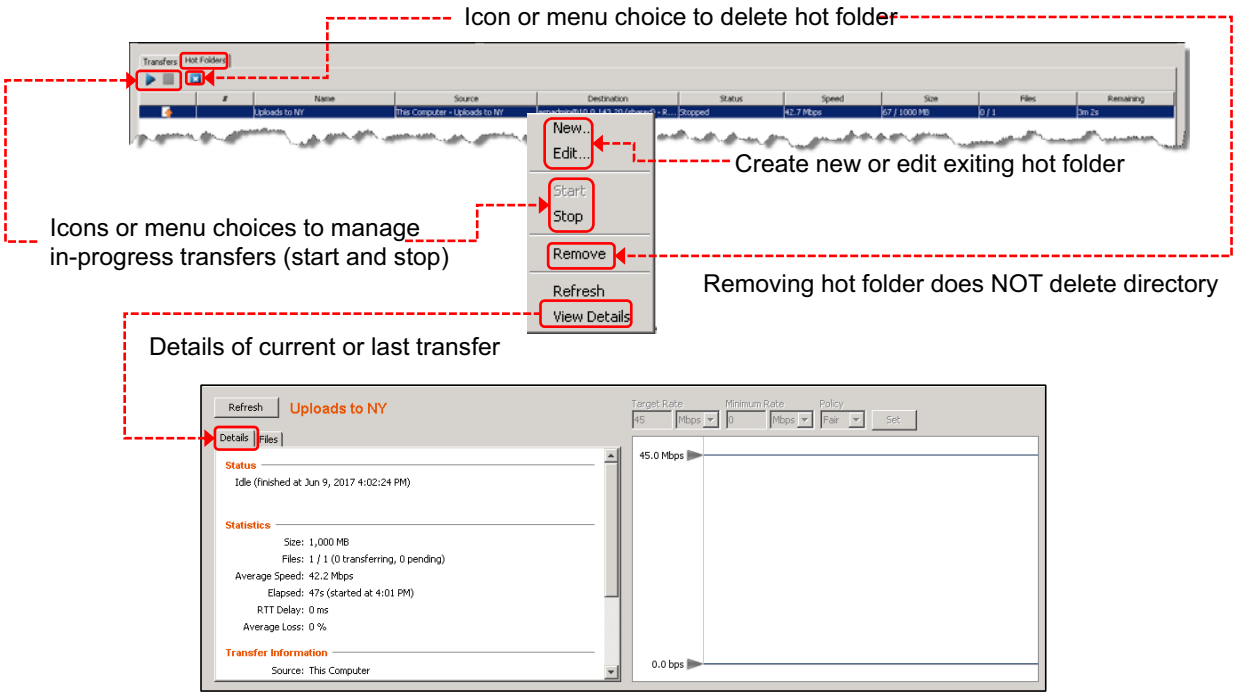
- *Automatically delete source file after* transfer: Same as this option for Push mode hot folders.
- *Transfer source directory contents only:* Transfer only the contents of the source directory, not the directory itself. If this option is NOT selected, the source destination directory is included in the transfer and the destination directory contains a copy of the source directory and its contents.
- *Delete empty source subdirectories:* Same as this option in Push mode.

It is important to remember that any empty folders in a hot folder are NOT PUSHED to the server. However, any empty folders created on the server ARE PULLED to the local destination.

IBM Training

Managing Hot Folders

Hot Folder tab in Transfer Panel



Configuring advanced features

Figure 7-21. Managing Hot Folders

The Transfer pane of the Aspera GUI includes a Hot Folders tab that shows all configured hot folders. This section provides information about current or previous transfers to or from those hot folders, just like the normal transfer information. In addition to viewing transfer progress, this panel provides you the ability create new hot folders, manage existing hot folder configurations and any transfers currently in-progress to or from the selected hot folder.

Three icons above the hot folder listing that can be used to stop or resume in-progress transfers and remove a selected hot folder configuration.

NOTE: Deleting a hot folder removes the hot folder configuration that is associated with a directory. However, the directory itself is NOT removed, and any files that are contained within the directory remain intact.

You can right mouse click a hot folder name to open a menu that where you can create a new hot folder (it opens the same New Hot Folder page previously discussed).

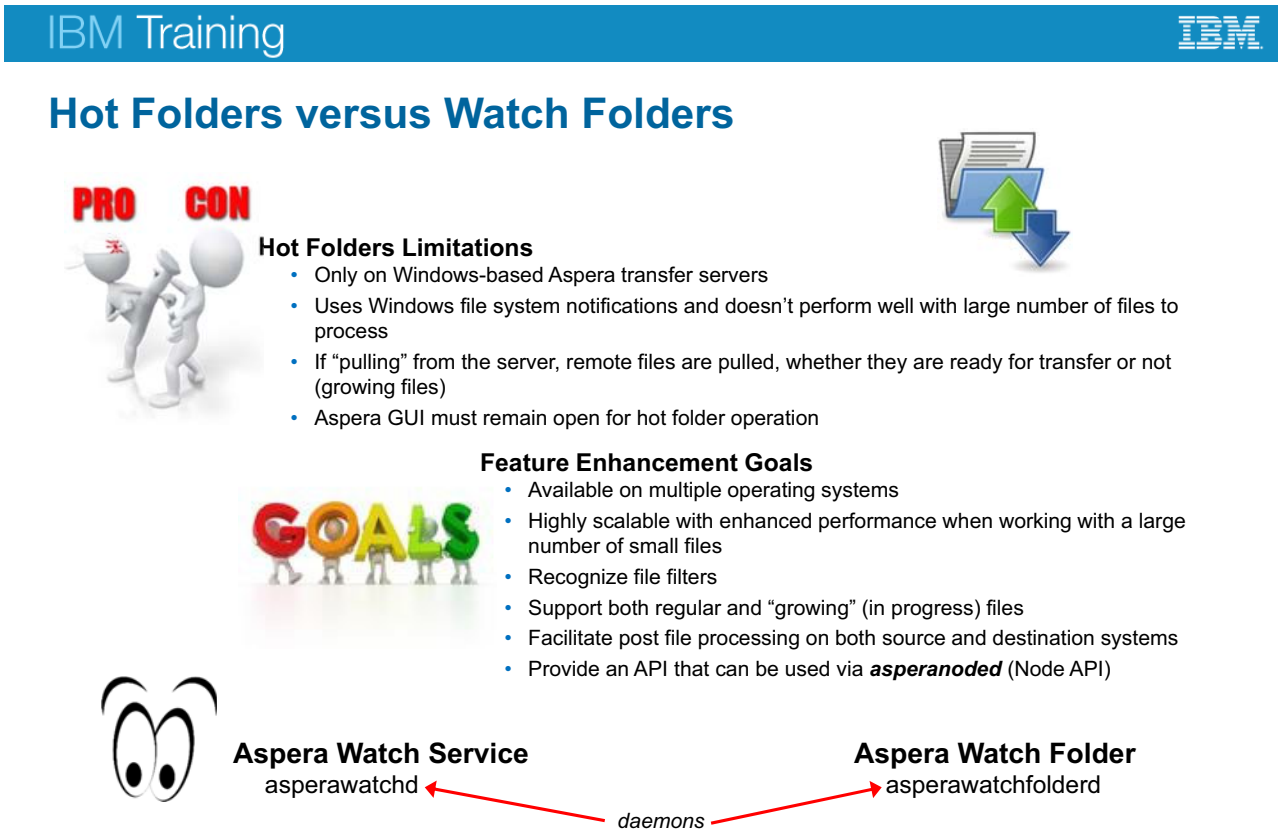


Figure 7-22. Hot Folders versus Watch Folders

As discussed, the hot folders feature provides an easy-to-use graphic interface that facilitates one-way replication between two locations, by using either a push (client to server) or pull (server to client) operation. However, the hot folder feature has some limitations that might or might not be a factor in deciding what to use in your Aspera environment.

Hot Folders Limitations

Only supported in the Windows version of Aspera Transfer Server software

Uses Windows OS file system notifications, which can degrade with many files

- When configured to pull from the server, remote files are pulled whether they are ready for transfer or not (growing files).
- Aspera GUI must remain open for hot folders to work

Watch Folder Design

Recognizing the limitations of hot folders as implemented on Windows-based transfer servers, Aspera developed a new file delivery feature that provides similar function to hot folders, but with several enhancements:

- Available on multiple operating systems

- Highly scalable and high performance, even when working with many small files
- Implements file filters
- Supports both regular and growing files
- Works with post file processing on both source and destination systems
- Includes an API that can be used via asperanoded (Node API)

Aspera Watch Services & Watch Folders

The function of the Aspera Watch Folder feature is similar to that of hot folders; the automation of file transfers from a source to a destination system. When new files are placed into a configured source directory (or existing files are modified or deleted), the changes are automatically transferred to a specified destination directory.

Aspera transfer servers (regardless of the operating system used) now include this type of function through services that are implemented as daemon processes, the Aspera Watch Service (aperawatchd) and Aspera Watch Folders (asperawatchfolderd).

IBM Training

Aspera Watch Folder: Overview

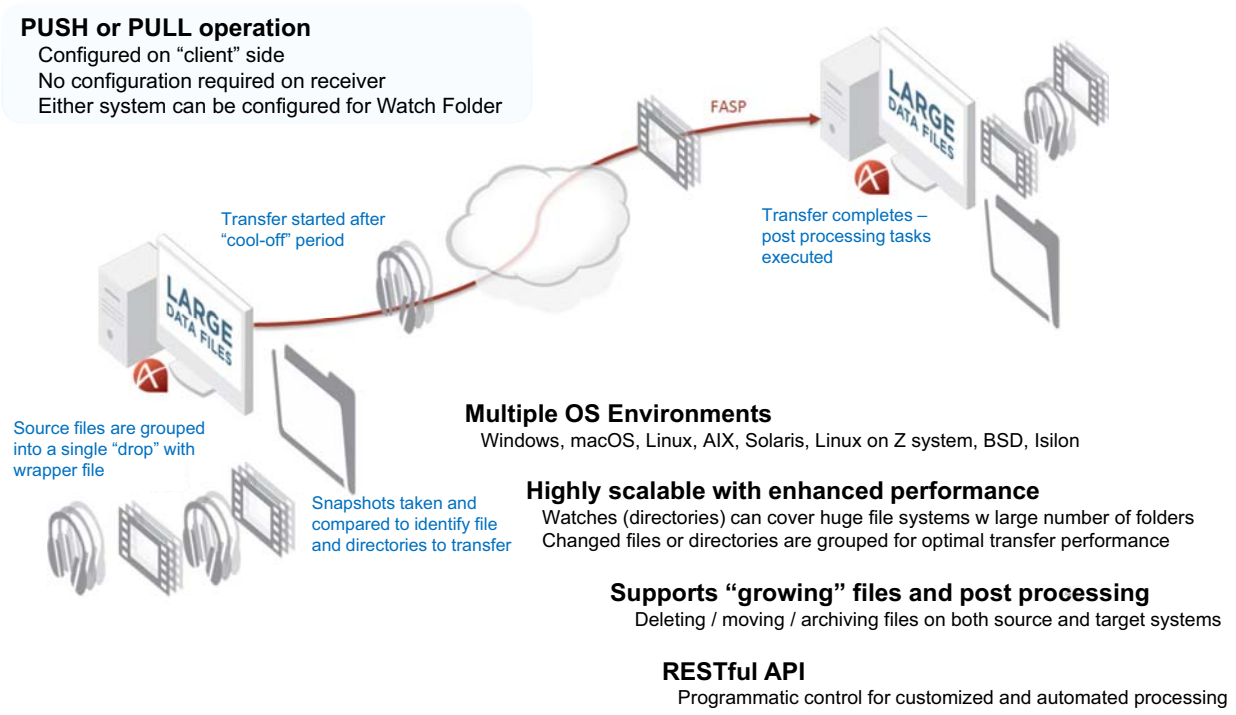


Figure 7-23. Aspera Watch Folder: Overview

Watch Folders are configured on the system that is acting as the managing side of the connection. It can be configured to push files from the local system or pull files from a remote system. Watches are implemented as the source directory and no additional configuration is required on the destination directory.

Multiple OS Environments

Aspera Watch Folders function is included with all IBM Aspera High-Speed Transfer Server software, which is available for most commonly deployed operating systems: Windows, macOS, Linux, AIX, Solaris, Linux on IBM Z, BSD, Isilon.

Highly scalable with enhanced performance

Watches (directories to be monitored) can be embedded in huge file systems, and each watch can include a large number of subdirectories. Files that changed and need to be transferred are identified by taking snapshots of the file system and determining the differences between the most current snapshot and the previous snapshot. This approach minimizes the need to perform file system scans (which can take a great deal of time and effort), which would ultimately limit the maximum size of the file system that can be monitored.

Additionally, the files that are identified as changed and needing to be transferred are grouped into a drop, which is treated as a single logical unit for transfer. Organizing files into a drop significantly

reduces the amount of network overhead that is associated with establishing the transfer session, which minimizes the performance impact of transferring a very large number of small files.

The Aspera Watch Service is used to perform this essential function. Details about exactly what tasks the Aspera Watch Service performs are provided later in this module.

Supports growing files and post processing

Growing files are files that are still being written and can have no end-of-file indicator. Growing files are usually associated with streaming data, and as such, frequently use file name extensions that can be easily recognized, for example, `*.mov`. Watch Folders can support growing files by using a file filter to identify the existence of growing files in the watched directory. Growing files are discovered by identifying files with a specified extension. Growing files are transferred with special parameter values that modify the details of how the file is read and transferred.

Watch Folders also support post processing on both the remote and local directories. Remote post processing is conducted first. After all the files of a drop are transferred, the remote system applies post processing rules that define how the files are stored; for example, overwriting existing files or merging the new files with existing folders.

After remote post processing is completed, files on the local system are manipulated according to file handling rules that are defined locally.

RESTful API

Aspera Node API uses asperanoded to receive requests to perform some `watchfolderd` tasks or provide information about watch folder functions. The transfer server can send data to `asperanoded` when certain events occur, such as a file or drop's state changes. For example, Aspera Console can send a command to create a new watch folder instance, or Console can listen for status updates that the transfer server sends via `asperanoded` to monitor watch folder transfers.

IBM Training

Watch Service & Watch Folder daemons

Configure PUSH or PULL watch folder
Configure with GUI, command line, or Watch Folder API

Aspera Watch Service Manager (*asperarund*)
- Stores Aspera Watch Service and Watch Folder configuration
- Manages ***asperawatchd*** and ***asperawatchfolderd*** services
- Starts services under different users without switching contexts

Aspera Watch Service (*asperawatchd*)
- Compares snapshots of file directory and generates file list of changes
- Eliminates need to scan file system each time a transfer occurs
- Does NOT perform the transfer

Aspera Watchfolders (*asperawatchfolderd*)
- Automates transfer of changed files
- Uses output of asperawatchd
- Can be created and managed from the GUI

Configuring advanced features

Figure 7-24. Watch Service & Watch Folder daemons

Watch folders can be configured as either a push (local-to-remote) or a pull (remote-to-local) folder. While some initial steps must be performed from the command line, watch folders can be configured from the Aspera GUI, from the command line, or by using the Watch Folder API.

The Aspera Watch Folder and Watch Service functions consists of three separate components, each running as a daemon process: the Watch Service Manager (asperarund), the Watch Service (asperawatchd), and Watch Folder (watchfolderd). All three of these daemons run on the system where the source directory is located.

Aspera Watch Service Manager

Management of both the Aspera Watch Service and Aspera Watch Folders is provided by the IBM Aspera Run Service (asperarund). This daemon process stores the Aspera Watch Service and the Aspera Watch Folders Service configurations in its database. The asperarund process automatically starts services when they are added and restarts services if they fail. A major benefit of asperarund is that it enables administrators to start services under different users without switching between accounts, and apply logging and database configurations to all services.

Aspera Watch Service

The Aspera Watch Service (asperawatchd) is a file system change detection and snapshot service that is optimized for speed, scale, and distributed sources. On file systems that have file system notifications, changes in source file systems (new files and directories, deleted items, and

renames) are detected immediately, eliminating the need to scan the file system. On file systems without file notifications, such as object storage, Solaris, AIX, and Isilon, file system scans are automatically triggered.

The Aspera Watch Service is tasked with monitoring configured directories (called watches) by creating file system snapshots, analyzing them for differences, and generating a list of any changes. The list of changes can then be used to transfer the identified files to a configured destination system.

A snapshot represents the structure of the file system at a particular point in time. After the snapshots are created, `asperawatchd` compares the snapshot with the previously created snapshot to detect file system changes at a granular level, and generates a file list of those changes. This approach eliminates the need for scanning the file system each time transfers are to be made, significantly enhancing the performance and scalability of transferring directories that contain a large number of files.

The Aspera Watch Service can also significantly improve the performance of Aspera Sync, when Sync requests snapshots from `asperawatchd` (a process that takes longer depending on the number of files in the directory). Additionally, the asperawatchd scans the directories that are specified in the aspera.conf file, reducing time spent waiting for a new scan.

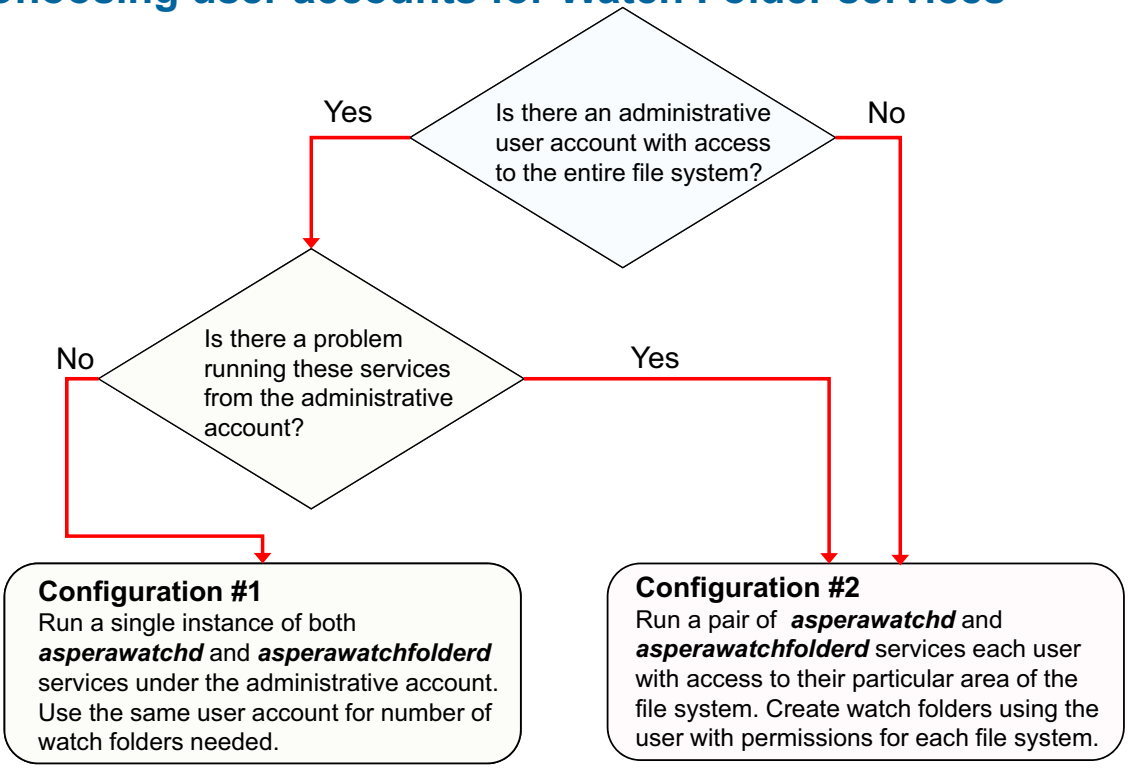
Aspera Watch Folders

Aspera Watch Folders (`asperawatchfolderd`) enables large-scale, automated file and directory transfers that include ultra large directories and growing file sources. Watch folders use the output of asperawatchd to automate file transfers from a source folder to a destination system.

Watch Folders can be created and managed in the GUI, which offers all the functions of the command-line set-up and management tools. Only the Node API user must be set up from the command line.

IBM Training

Choosing user accounts for Watch Folder services



Configuring advanced features

Figure 7-25. Choosing user accounts for Watch Folder services

Watch folder services must be run under a user with access to every area of your file system in which you intend to create a watch folder. Watch folders facilitate running multiple instances of these services under different users. Most users run these services under one user. In this case, you must choose a user with permissions to access to the entire file system.

However, you can need to run multiple instances of these services to access every area of your file system. For example, your file system can include mounted storage from the marketing department and another from the release team. You might not have a single user that can access files in both mounted storages. Or your administrative account can access the entire file system, but your policy prohibits running the asperawatchd and asperawatchfolderd services under that user account.

Watch folders can be configured in two ways, depending on your scenario:

Configuration #1

This scenario is the most common configuration of watch folder services. Choose an account with read permissions for all your files and configure both asperawatchd and asperawatchfolderd services under this single user account.

Configuration #2

In this scenario, you cannot run watch folder services under the administrative account or you do not have a single user with permission to access to the entire file system. So, you must run pairs of

asperawatchd and asperawatchfolderd services under enough users to access your entire file system.

For example, your mounted storage from the marketing department is accessible only by user bob. Another storage from the release team, is accessible only by user sally. You need to run a pair of asperawatchd and asperawatchfolderd services under each user.

Aspera suggests you configure services and manage watch folders in a multi-user context by using the Node API. You can interact with the Node API from the IBM Aspera Console or by using curl commands from the command line.

IBM Training

Prepare system for Watch Folders

- Confirm asperarund is active
  # systemctl status asperarund *
  - Select/create transfer user account to run serviceS
    # systemctl status asperarund
  - Verify user permissions to default log directory
    - Configure docroot or restrictions for transfer user
      # asconfigurator -x "set_user_data;user_name,username;file_restriction,|path"

- Add server entries in aspera.conf

```
<server>
  <http_port>9091</http_port>
  <https_port>9092</https_port
  <enable_http>false</enable_http>
  <enable_https>true</enable_https>
</server>
```

- Configure **asperawatchd** and **asperawatchfolderd** settings (optional)
- Configure Linux for many watch folders (optional)

 # echo "fs.inotify.max_user_watches=xxxxxxx" >> /etc/sysctl.conf
 # echo "fs.inotify.max_user_instances=xxxx" >> /etc/sysctl.conf
 # echo "fs.file-max=xxxxxx" >> /etc/sysctl.conf
 # sysctl -p /etc/sysctl.conf

- Create node API user w admin ACL (required to use Aspera GUI for configuration)

 # /opt/aspera/bin/asnodeadmin -a -u node_username -p node_password -x transfer_user --acl-set "admin,impersonation"

* Configuration defined in aspera.conf file

Configuring advanced features

Figure 7-26. Prepare system for Watch Folders

It is necessary to configure the operating system to support Watch Folders before creating the watch folders themselves.

Confirm that asperarund is active

The asperarund daemon must be running for the operation of any Aspera Watch Folders function. This daemon is automatically configured to start when the system boots up during the installation process, but it is a good idea to confirm that it is running before attempting any further configuration.

NOTE: Logging and Redis database configuration that is used by asperarund can be optionally configured with the following entries: (See the Creating, Managing, and Configuring Services section of the *IBM Aspera HSST Administration Guide* for further details):

```
<server>
  . . .
    <rund>
      <log_level>log</log_level>
      <log_directory>AS_NULL</log_directory>
      <db_spec>redis:127.0.0.1:31415</db_spec>
    </rund>
    <watch>
      . . .
```

</watch>
</server>

Select or create user account to run services

As previously discussed, identify the user account that you need to run the services.

Set docroot or restriction for user

Docroots and path restrictions limit the area of a file system or object storage accessible by the user. Users can create watch folders and watch services on directories or objects only within their docroot or restriction.

Note: Users can have a docroot or restriction, but not both or Watch Folder creation fails

Associate user with Node API user

You must configure a Node API user with the admin/impersonation ACLs set as follows:

Cloud

```
/opt/aspera/bin/asnodeadmin –a –u node_user_name –p node_password –x transfer_user
- - acl-set “admin,impersonation”
```

Configure Linux (optional, dependent upon number of watch folders used).

If you plan to watch more than 8,200 directories on a Linux computer, you might need to configure it to support that many processes. See the Configuring Custom Watch Folder Permissions Policies in the GUI section of the *IBM Aspera HSTS Administration Guide* for instructions to modify OS settings.

Configure asperawatchd and asperawatchfolderd (optional):

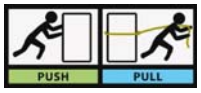
Default configuration values for these services are optimized for most users. You can configure the snapshot database, snapshot frequency, logging, scan threads, and drop handling if needed. For more information, see the Watch Service Configuration and Watch Folder Service Configuration sections of the *IBM Aspera HSTS Administration Guide* for details.

Verify user permissions to default log directory

The user under which the service is running must have permission to access the default log directory.

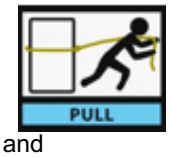
IBM Training

Watch Folder restrictions



Restrictions on ALL watch folders

- Only local-to-remote (push) and remote-to-local (pull) configurations supported
  NO remote-to-remote or local-to-local
- Growing files only supported for local sources (push Watch Folders)
- No source file archiving if Watch Folder source is in object storage.



Restrictions on PULL watch folders

- Remote server must run HSTS or HST End Point (version 3.8.0 or newer)
- Must be authenticated with access key ID and secret, Node API username and password, or IBM Aspera Shares credentials
  - SSH authentication is not supported for remote sources
- Cannot be authenticated by Node API if transfer user is configured with restrictions
- Cannot use IBM Aspera on Cloud or IBM Aspera Transfer Cluster Manager nodes as remote source (only supported as “push” function)
- Do not support growing files

Figure 7-27. Watch Folder restrictions

Watch folders can be configured as either a push or a pull directory. Some global restrictions apply to all watch folders, regardless if they are configured as push or pull, while pull watch folders have extra restrictions.

Restrictions on ALL watch folders

Only local-to-remote (push) and remote-to-local (pull) configurations are supported. Remote-to-remote and local-to-local are not supported.

Growing files are supported only for local sources (push Watch Folders).

If the Watch Folder source is in object storage, source file archiving is not supported.

Restrictions on pull watch folders

The remote server must run IBM Aspera HST Server or IBM Aspera HST Endpoint version 3.8.0 or newer.

Pull Watch Folders must be authenticated with an access key ID and secret, a Node API username and password, or IBM Aspera Shares credentials. SSH authentication is not supported for remote sources.

Pull Watch Folders that use Node API authentication cannot be authenticated with a Node API user whose associated transfer user is configured with a restriction (the Watch Folder status is

reported as impaired). Edit the transfer user's configuration to use a docroot, restart asperanoded, and the Watch Folder recovers automatically.

Pull Watch Folders cannot use IBM Aspera on Cloud (including IBM Aspera on Cloud transfer service nodes) or IBM Aspera Transfer Cluster Manager nodes as the remote source.

Pull Watch Folders do not support growing files.

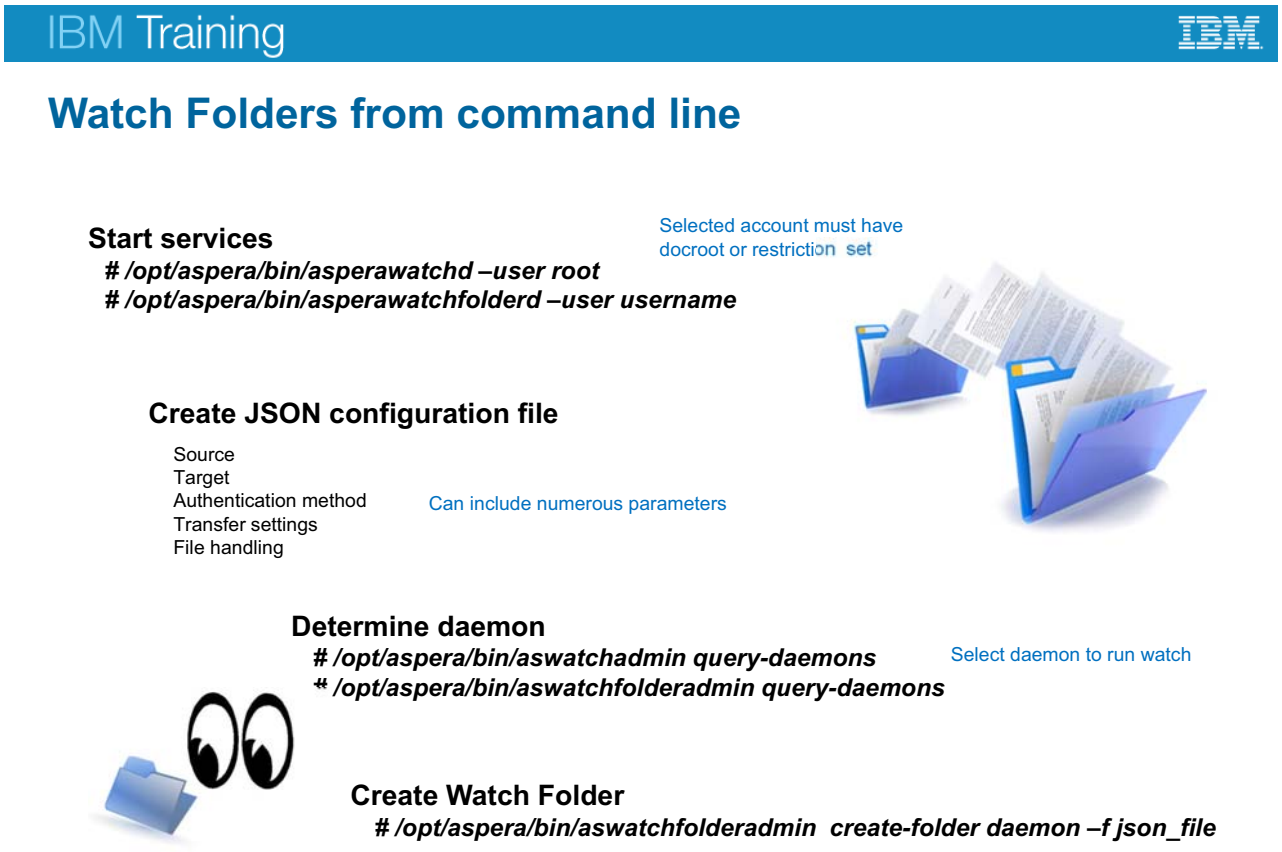


Figure 7-28. Watch Folders from command line

Start Services

Both asperawatchd and asperawatchfolderd are run under system users. A docroot is configured for the user in aspera.conf. If no custom log directory is configured in aspera.conf, then the user needs write permissions to the default log directory. Aspera suggests running asperawatchd under root, and selecting a user to run asperawatchfolderd.

To start asperawatchd and asperawatchfolderd, run the corresponding command:

```
/opt/aspera/sbin/asperawatchd --user username
```

```
/opt/aspera/sbin/asperawatchfolderd --user username
```

A watch service must be running under a user before a Watch Folders service can be created for that user.

Though the default values are already optimized for most users, you can also configure the snapshot database, snapshot frequency, logging, scan threads, and drop handling, among other features. For instructions, see the Watch Service Configuration and Watch Folder Service Configuration sections of the *IBM Aspera HSTS Administration Guide*.

Create a JSON configuration file

Watch Folders are configured by using a JSON configuration file. The Watch Folder JSON file describes the source, target, and authentication to the remote server. This configuration file can

also specify transfer session settings, file handling and post-processing, filters, and growing file handling. (JSON configuration files briefly discussed later in this module).

List running daemons

Use the `aswatchadmin` and `aswatchfolderadmin` utilities to retrieve a list of running daemons. You need the daemon name when you create the watch folder.

```
/opt/aspera/bin/aswatchadmin query-daemons
```

```
/opt/aspera/bin/aswatchfolderadmin query-daemons
```

Create Watch Folder

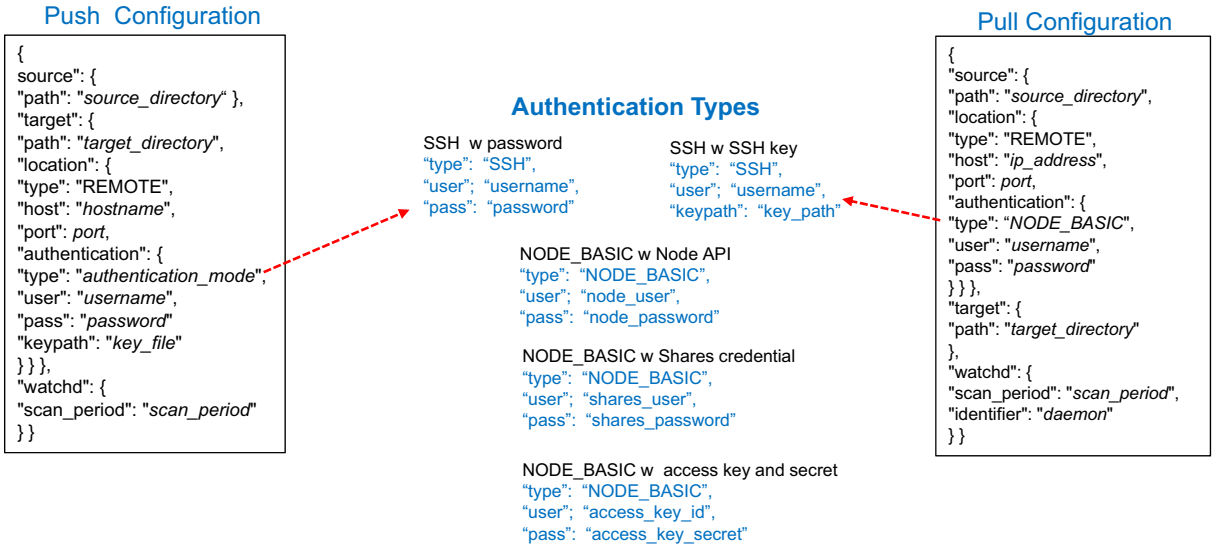
Create the watch folder by running the following command, where daemon is the user that runs the watch folder service, and `json_file` is the path to the configuration file.

```
/opt/aspera/bin/aswatchfolderadmin  create-folder daemon –f json_file
```

NOTE: When you create a watch folder, a watch service subscription is automatically created to monitor the source directory. If the subscription is deleted or impaired, Watch Folders automatically creates a new subscription; however, the new subscription does not retain the file change history and all files in the source directory are retransferred.

IBM Training

JSON configuration file



See the "Watch Folder JSON Configuration File Reference" section of the IBM Aspera HSST Administration Guide for details

Figure 7-29. JSON configuration file

Watch Folders supports transfers between a local server and a remote server. For the local server, Watch Folders requires only the local path, whether it is the source or target. For the remote server, Watch Folders requires the host address, port for authentication, and authentication credentials.

Authentication credentials are required for remote systems, and the details of what values to use for authentication can be defined.

The examples that are shown represent configuration file contents for simple push and pull watch folders, but numerous other parameters can also be configured within the JSON configuration file.

NOTE: See the Watch Folder JSON Configuration File Reference section of the *IBM Aspera HSST Administration Guide* for details about all parameters that can be defined in the JSON configuration file

IBM Training

The Watch Folder GUI

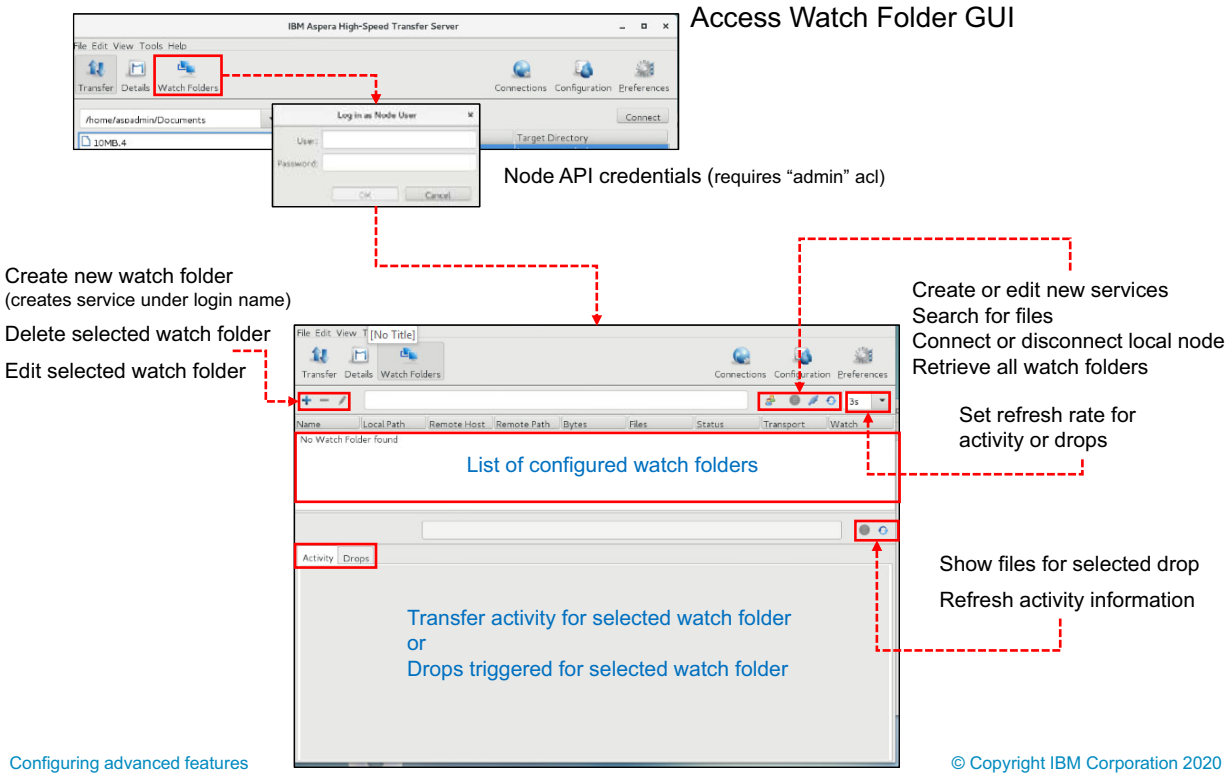


Figure 7-30. The Watch Folder GUI

While Watch Folders can be created and managed from the command line, it is generally easier to configure by using the GUI. For more information about creating watch folder from the command line, see the Creating, Managing, and Configuring Services section of the *IBM Aspera HSTS Administration Guide*.

The Watch Folder GUI can be accessed by selecting the Watch Folder link at the top of the Transfer Server GUI. Use the Node API credentials that you created with the `- - acl-set “admin, impersonation”` option.

In addition to listing watch folders, the Watch Folder GUI also provides numerous links for performing critical tasks that are associated with watch folders.

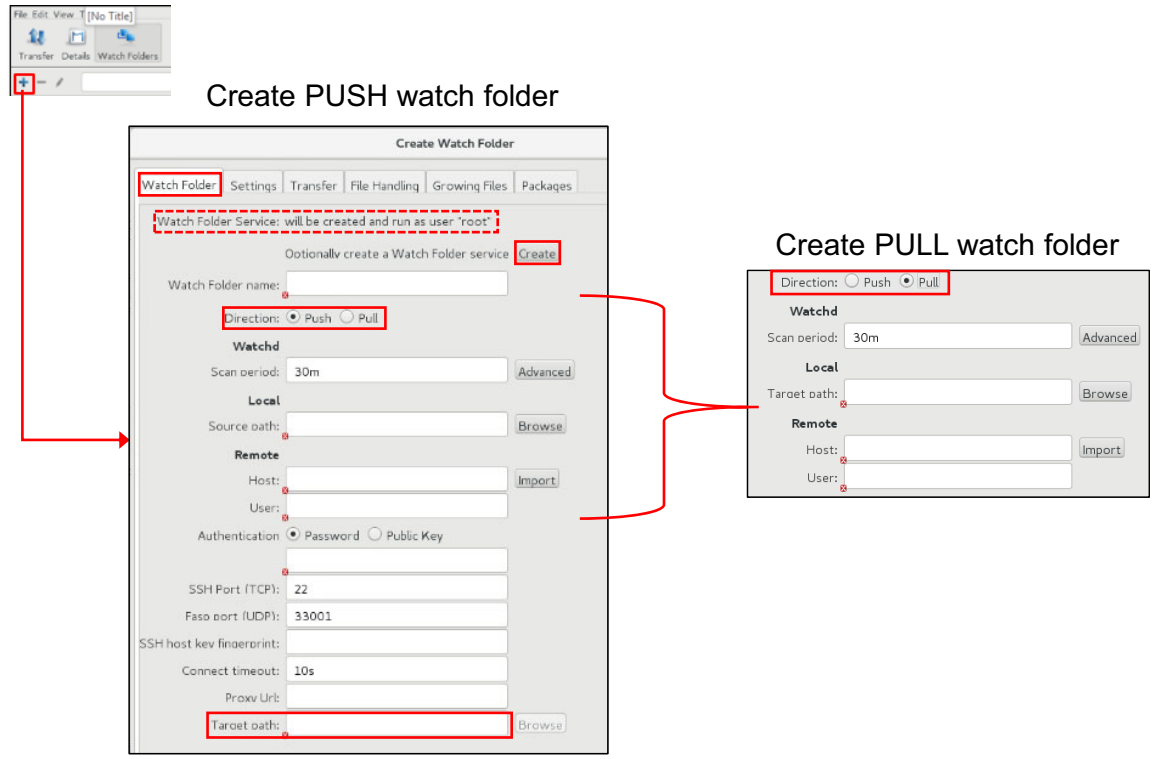
The Watch Folder GUI displays a list of configured watch folders in the center pane of the GUI. You can highlight one of the configured watch folders then select the Edit or Delete link at the left side of the screen to manage the selected watch folder.

Below the listing of configured watch folders (Activity tab) is a summary of activity statistics that are associated with the selected watch folder. Selecting the Drops tab displays the status of drops that are associated with the selected watch folder. The Watch Folder service is responsible for determining changes to the configured watch folder and adds identified files to a Drop, which is used when transferring files. Details about managing watch folders are provided later in this module.

Several other options are provided in the GUI, which are used when managing watch folders (addressed later in this training module).

IBM Training

Configuring Watch Folders from GUI



Configuring advanced features

Figure 7-31. Configuring Watch Folders from GUI

A new watch folder can be created by clicking the + icon that is provided on the opening Watch Folder GUI page.

NOTE: If the error message, “You cannot create Watch Folders. Please contact your Administrator,**”** Is displayed, the Node API user is not configured with the necessary permissions.

Select the appropriate checkbox to create a push or pull watch folder.

The fields associated with the watch folder location (source and target) and relevant login credentials differ, depending upon whether you are configuring a push or pull watch folder.

Host (and authentication): The IP address, DNS, hostname, or URL of the remote server. Click Import to import connection information from the Connections list. The username, authentication, and target path are automatically populated from the connection settings, as are settings under Transfer and File Handling.

If you enter the host manually, use the following syntax based on the type of remote endpoint and authentication method:

- HST Server or HST Endpoint that is authenticated with an SSH user: Enter the IP address or hostname of the endpoint for the host, then enter the SSH user and their password or public key.
- HST Server or HST Endpoint that is authenticated with Node API or access key credentials: Enter the node URL as `ip_address_or_server_url:9092/`. If a different HTTPS port is

configured, replace 9092 with the correct port. Enter the Node API username or access key ID as the User and the Node API user's password or the access key secret as the Password.

• IBM Aspera on Cloud (including IBM Aspera on Cloud transfer service nodes) and IBM Aspera Transfer Cluster Manager nodes: Enter the endpoint URL as:

https://ip_address_or_server_url:443/ and provide the access key ID and secret for the user and password.

• IBM Aspera Shares: Enter the URL of the Shares server as https://ip_address:443 and provide the Shares login credentials.

Watchd scan period: Set the amount of time between assessments of the watch (from end of one to start of the next). The value can be specified with units, such as 30m for 30 minutes, or 24-hour clock, such as 01:00:00 for 1 hour. The asperawatchd process monitors watch folders for changes, independent of the snapshot minimum interval and snapshot minimum changes to ensure that changes are captured.

On file systems without file notifications, such as object storage, mounted storage (NFS), Solaris, AIX, and Isilon, file system scans triggered by the scan period are used to detect file changes. In this case, set the scan period to frequently scan for changes. On operating systems that support file notifications (Linux, Windows, macOS), asperawatchd uses the file notifications as the primary means for detecting changes, and the scan period serves as a backup. In this case, the default value of 30 minutes is usually acceptable and no change is necessary. To never scan, and rely entirely on file notifications, set to infinite.

NOTE: Shorter scan periods detect changes faster but can result in greater resource consumption, particularly for object storage.

IBM Training

Configuring Watch Folders from GUI (Cont.)

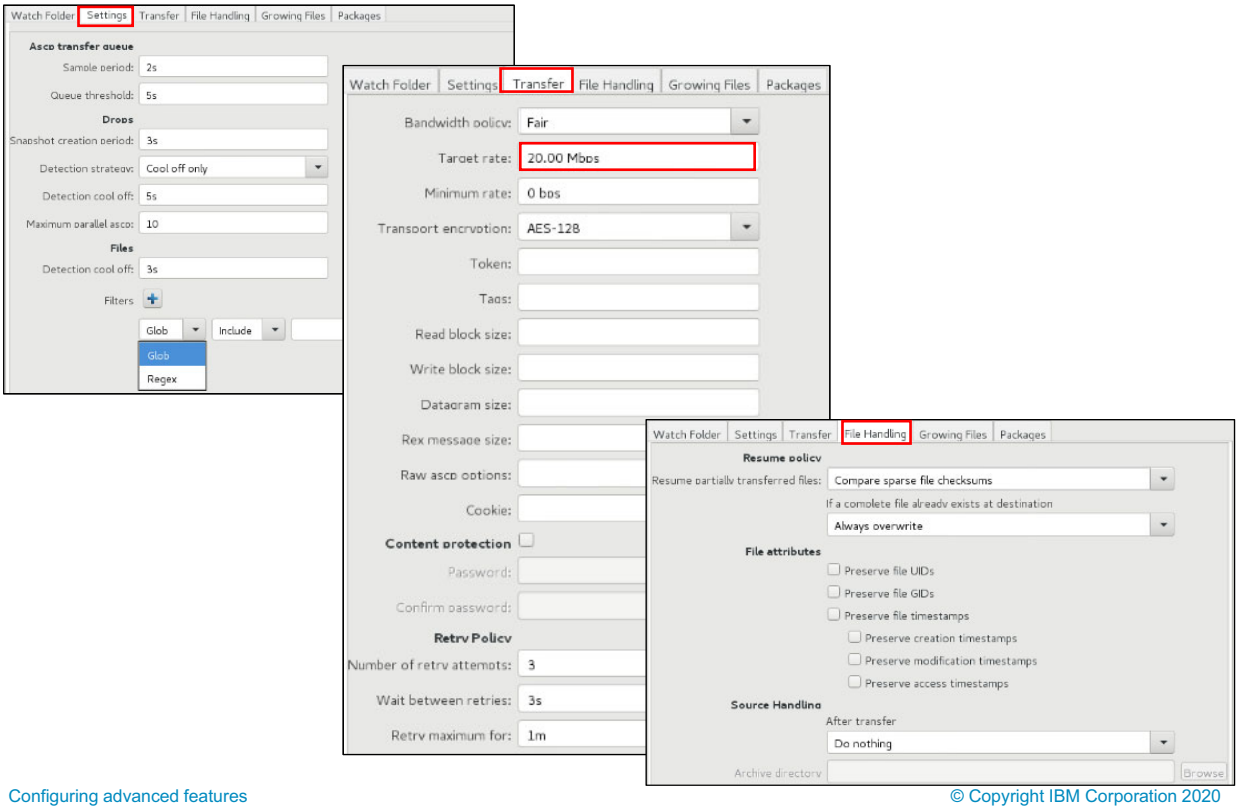


Figure 7-32. Configuring Watch Folders from GUI (Cont.)

Several tabs across the top of the Create New Watch Folder page provide links to various parameters that manage the transfer activity for the watch folder:

Settings:

This page provides fields for modifying default values for parameters that define how frequently the file system is scanned for changes and define how drops are managed for this watch folder.

Transfer

Modify default values about how files are transferred, for example, bandwidth policy, target rate, or encryption

File Handling

Set specific values for how transferred files are managed, for example, how to deal with existing files, file permission, and what to do with source files after a successful transfer.

Growing Files

Configure parameter values when working with growing files.

IBM Training

Managing Watch Services from GUI

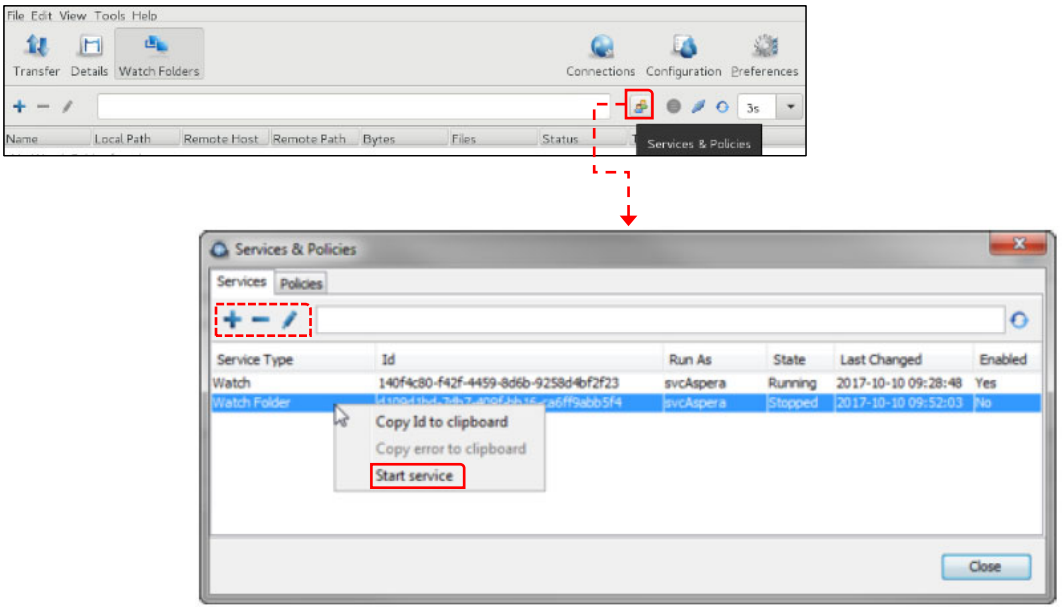


Figure 7-33. Managing Watch Services from GUI

Services can be managed (added, started, restarted, stopped, edited, and deleted) in the Services & Policies dialogue.

Create a new service by clicking the + icon. Select the type of service (Watch or Watch Folder). If you select Watch Folder, a Watch service is automatically created if one does not exist for the user that runs the Watch Folder service.

NOTE: If a service is added outside of the GUI, the services do not appear in the list of services until the list is refreshed. The node API user must have permissions to view all services if the services were created for another user

IBM Training

What you learned

- Aspera supports both SSH and HTTP authentication methods –
  SSH best suited when all systems part of same administrative domain
  HTTP best suited for access with arbitrary clients or from internet at large
- Aspera products all include a self-signed SSL certificate that can be replaced with one recognized by a CA
- HTTP fallback is enabled by adding the required entries in *the <http_server>* section of the **aspera.conf** file
- The pre/post function is implemented by referencing scripts or executables in the **aspera-prepost** file
- Node API provides a means of accessing server data and performing tasks remotely
- Node API users are created and managed using the ***asnodeadmin*** utility
- The Hot Folders feature is only available on Windows servers – offering automatic one-way transfers
- The Aspera Watch Service is available on all platforms and offers high performance by minimizing the time required to scan file systems for changes before performing transfers
- The ***asperawatchd*** daemon detects file system changes, performs snapshots, and generates a list of changes that are used by ***ascp*** or watch folders
- The ***asperawatchfolderd*** daemon works with ***asperawatchd*** to automate transfers of large number of files
- The ***aswatchfolderadmin*** command can be used to manage watch folders

Figure 7-34. What you learned

IBM Training

Unit summary

- Configure IBM Aspera High-Speed Transfer Server to use custom SSL certificates and token authorization
- Outline the process of configuring HTTP Fallback
- Manipulate files using the Aspera Pre/Post feature
- Configure and manage Node API settings
- Distinguish between the hot folders and Aspera Watch Service
- Explain the procedure for implementing hot folders on Windows platforms
- Implement Aspera Watch Folders

Figure 7-35. Unit summary

IBM Training

Review questions (1 of 2)

1. True or False: After HTTP Fallback is configured, FASP-based command-line transfers use HTTP when a problem arises with FASP.

2. Which of the following features facilitate running a virus checking routing on received files, and if no virus is found, moving them to another directory? Select all that apply:
   A. Node API
   B. Aspera Pre/Post
   C. Aspera central
   D. Hot folders

Figure 7-36. Review questions (1 of 2)

Review questions (2 of 2)

3. Which of the following statements are most accurate regarding Watch Folders? Select all that apply:
   A. Watch Folders can be configured to “push” or “pull” files to or from a remote system
   B. The number of Watch Folders that can be configured on an IBM Aspera Transfer Server is 50
   C. Watch Folders cannot be configured to support remote-remote or local-to-local transfers
   D. Implementing Watch Folders requires configuration on both the sending and receiving Aspera systems

4. True or False: The Aspera GUI can monitor the transfer activities of configured Watch Folders

Figure 7-37. Review questions (2 of 2)

IBM Training

Review answers (1 of 2)

1. True or False: True or False: After HTTP Fallback is configured, FASP-based command-line transfers use HTTP when a problem arises with FASP.

   The answer is False. HTTP Fallback is only functional when using IBM Aspera web-based applications. It does not work when using ascp from the command line

2. Which of the following features facilitate running a virus checking routine on received files, and if no virus is found, moving them to another directory? Select all that apply:

   A. Node API
   B. Aspera Pre/Post
   C. Aspera central
   D. Hot folders

   The answer is B

Figure 7-38. Review answers (1 of 2)

IBM Training

Review questions (2 of 2)

3. Which of the following statements are most accurate regarding Watch Folders? Select all that apply:
   A. Watch Folders can be configured to “push” or “pull” files to or from a remote system
   B. The number of Watch Folders that can be configured on an IBM Aspera Transfer Server is 50
   C. Watch Folders cannot be configured to support remote-remote or local-to-local transfers
   D. Implementing Watch Folders requires configuration on both the sending and receiving Aspera systems

   The answer is A and C

4. True or False: The Aspera GUI can monitor the transfer activities of configured Watch Folders

   The answer is True

Figure 7-39. Review questions (2 of 2)

IBM Training

Lab Exercise 6

In this lab, you configure, implement Node API, and verify Node API.
You also configure and test Hot Folder function on a Windows platform
You configure and use Watch Folders on Linux systems.

Figure 7-40. Lab Exercise 6

Unit 8. Routine maintenance tasks

Estimated time

01:30

Overview

This unit identifies common performance bottlenecks, presents common maintenance tasks, and introduces how to interpret some of the Aspera log file entries.

IBM Training

Unit objectives

- Identify common transfer performance bottlenecks
- Describe the process for conducting backups of Aspera files and configurations
- Access log files to identify errors
- Use log files to analyze file transfer performance

Figure 8-1. Unit objectives

IBM Training

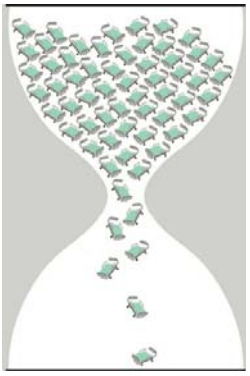
Potential transfer bottlenecks

Aspera license or Bandwidth settings

Disk I/O - Sender or Receiver
Common with high concurrency or high packet loss or if disk accessed by other applications

Disk I/O - Sender or Receiver NAS / SAN
Might be due to physical configuration of disks or access protocol (NFS or CIFS)

Firewall / QoS - Sender or Receiver
Firewalls that inspect packet deeply or implement QoS can cause delays



Physical Bandwidth - Receiver or Sender

CPU - Sender or Receiver
Common if encryption is turned on

Network and Network Devices
VPN and packet fragmentation can cause delays

ISP Controlling UDP Ports
ISPs can limit UDP throughput

Figure 8-2. Potential transfer bottlenecks

IBM Aspera FASP-based transfer server products are engineered to provide you with high-speed data transfers. However, numerous issues can limit the performance of Aspera transfers. If you feel that files are not transferring at the expected speeds, you might want to consider some bottleneck issues. Some of these considerations are configuration issues (Bandwidth configuration settings), while other potential issues require detailed analysis, for example, disk I/O performance, quality of service (QoS) settings, and network issues.

The factors that can affect the end-to-end IBM Aspera transfer rate are as follows:

- Aspera license rate
- Bandwidth settings
- Disk I/O (on the send and receiver)
- Firewall settings
- Physical bandwidth (size of pipe)
- CPU speed on the sender or receiver
- Network devices
- ISP issues

IBM Training

Transfer Server maintenance tasks

- **Backup** - */opt/aspera*
- **Upgrade Software** – *Install new version – configuration retained*
- **Modify license** – *cat new_license >> /opt/aspera/etc/aspera-license*
- **Verify Aspera user configuration** - */opt/aspera/bin/asuserdata –v*
- **Log files** – *identifying performance issues*

Figure 8-3. Transfer Server maintenance tasks

IBM Training

Aspera logging: Overview

Figure 8-4. Aspera logging: Overview

The IBM Aspera High-Speed Transfer Server software maintains extensive historical information through its logging function. The log files include detailed information about events that are associated with every Aspera service and component, and can be useful for review and support requests.

You can view log files by using the Tools menu. Depending upon what operating environment you are using, the HSST application provides slightly different options for log files. On Linux systems, the only option available for logs is the View Log option. If you are running the transfer server in a Windows environment, you see two options that are associated with log files, View Log and Open Logs Folder.

The View Log option opens a window that displays logging information about the IBM Aspera HSTS application. The location of the log file is operating system-dependent, `C:\Program Files (x86)\Aspera\Enterprise Server\var\log` on Windows systems and `/var/log/messages` on Linux systems.

While viewed from within the IBM Aspera HSTS GUI, the log file is named `log.html`, and is located in the `/root/.aspera/enterprise server` directory in Linux environments or `C:\Users\Administrator\AppData\Roaming\Aspera\Enterprise Server` directory on Windows systems. The contents of these log files that are displayed from within the transfer server application extract relevant information from other files.

In Linux environments, output of all Aspera utilities and services are stored in a single file, by default `/var/log/messages`. You can extract the specific log information that you want by piping the file contents to grep. You can also redirect Aspera logging data to a different file (for example, `/var/log/aspera.log`) by modifying the `/etc/syslog.conf` file.

If you choose to separate the Aspera log information from the default, you should also modify the `/etc/logrotate.d/syslog` file to include the log files you configured.

The general log file shows entries based on date and time. Entries can be identified with any one of three levels of severity: INFO provides neutral information; WARN indicates a non-fatal event; and ERROR indicates an unrecoverable event, which cannot be completed. The file also indicates which routine generated the entry for the log file (Thread). The Messages field provides a brief description of the event.

For example, the highlighted INFO entry provides basic information that is recorded as the application starts. The message indicates that the system found the language folder. The ERROR entry indicates an unrecoverable problem when starting (the system encountered a problem when loading the license file and cannot continue. The WARN entry indicates that a specific parameter (Vlinks) is configured incorrectly.

While the information provided by the log file can seem cryptic, the entries provide critical details when resolving a problem in IBM Aspera HSTS operation.

The Windows version of the IBM Aspera HSTS application provides an extra option under the Tools menu, the Open Logs Folder option. Selecting this option opens a window with multiple log files, each associated with a specific service or component of IBM Aspera HSTS. Depending upon how the system was implemented, different log files for various utilities and services, such as asconfigurator, ascmd, asperacentral, and ascp might be displayed.

The name of the log file indicates the service or component that is generating the data that is written to the file. The `asconfigurator` routine is called by the GUI when configuration changes are made, and is responsible for updating the aspera.conf file with those changes.

The `ascmd` service is responsible for file browsing and manipulation from within the Aspera user interface. The `asperacentral` service handles server-side transfers.

The ascp program is tasked with managing the actual transfer functions. So, knowing what these services and component do helps to understand the information stored in their log file.

asconfigurator.log: Stores event information about the configuration of the IBM Aspera HSTS application

ascmd.log: Stores information about events associated with Aspera file browsing

asperacentral.log: Events associated with file transfers between servers

aspera-scp-transfer.log: Record of all FASP-based transfer statistics and events

IBM Training

Reading transfer logs

Session Start	Initializing FASP version x.x.x.xxxxx, . . . ….. FASP Session Start
Transfer Start	FASP Session uuid=xxxxxx.xxxxx …… Sender bl . . . (if sending files) Receiver bl . . . (If receiving files)
Session Stop	FASP Session Stop id=xxxxxx.xxxxx . . .
Summary	FASP Session Statistics (Receiver) . . . ====== File Transfer Statistics ======

Figure 8-5. Reading transfer logs

Every transfer that uses the FASP protocol is logged. While a detailed explanation of the Aspera logs is outside the scope of this training course, it is useful to understand the general organization of the data collected about each transfer. Every transfer occurs within a session. The logs show a session start and a session stop, with the individual transfer data located between the session start and session stop.

A session summary is provided which reports the actual statistics that are associated with the session.

IBM Training

Transfer Server log: Performance

t m c n vl vr r

rate t/m/c/n/vl/vr/r=200000000/0/4241355/101242456/190512576/160039440/4241355

t = target rate
m = minimum rate
c = calculated rate
n = network rate
vl = vlink rate
vr = remote vlink rate
r = storage controller rate

c = lowest value from n, vl, vr, r

rate_rtt b/l/h/s/r/f=10/70/93/89/1

b = base rtt
l = lowest rtt seen in 20 sec period
h = highest rtt seen in 20 sec period
s = smoothed rtt (average)

s much higher than b = ??

Figure 8-6. Transfer Server log: Performance

You can view performance settings (target rate, minimum rate, Vlink rate, storage I/O rate) for a transfer within the Aspera log files by locating the lines that start with “rate” and “rate_rtt”. The values that are displayed on the line have the following meanings:

t = target rate

m = minimum rate

c = calculated rate

r = network rate

vl = Vlink rate

vr = remote Vlink rate

r = storage controller rate

The calculated rate is determined by selecting the lowest rate from the following factors: network rate, Vlink rate, and remote Vlink rate and storage controller rate.

The calculated rate shown (42 Mbps) is the result of comparing the values that are reported for those factors. The storage controller rate is the lowest, which is what is set at the calculated rate.

The rate_rtt values identify the rtt times observed that are reported in a 20 sec interval, and are used by the Aspera algorithm to manage the adaptive rate control. The following values are identified:

b = base rtt

l = lowest rtt seen

h = highest rtt seen

s = smoothed rtt rate

The smoothed rtt rate is used in the algorithm to adjust the rate that the sending system used for its transmissions.

The example that is shown indicates a difference between the base rtt and the smoothed rtt. This difference indicates that network congestion or slow routing queues somewhere between the sender and the receiver are impacting transfer performance.

What you learned

- Several factors, including, licensed maximum rate, configuration parameters, physical bandwidth, CPU/Memory (sender or receiver), disk I/O (sender or receiver), and network devices (routers and firewalls) can limit transfer rates.
- Backing up **/opt/aspera** captures all configuration data
- Software upgrades can be performed by installing the new software (make sure /**opt/aspera** is backed up before starting an upgrade
- Access log files to identify errors:

  Windows *C:\Program Files (x86)\Aspera\Enterprise Server\var\log*

  Linux */var/log/messages*
- The Aspera transfer log files report transfer statistics every 20 seconds of the transfer and can be used as an indicator of where transfer performance is limited

Figure 8-7. What you learned

IBM Training

Unit summary

- Identify common transfer performance bottlenecks
- Describe the process for conducting backups of Aspera files and configurations
- Access log files to identify errors
- Use log files to analyze file transfer performance

Figure 8-8. Unit summary

Review questions

1. Which of the following statement is most accurate regarding upgrading IBM Aspera High-Speed Transfer Server software? Select all that apply:
   A. It is recommended to remove the */opt/aspera* directory on Linux systems and *C:\program files\aspera* on Windows systems before starting an upgrade to ensure that all essential files are properly replaced.
   B. It is necessary to renew the Aspera license file after an upgrade.
   C. IBM Aspera High-Speed Transfer Server upgrades are implemented by running a new version of the installation software.
   D. Specific patches can be applied to upgrade specific components of the application.

2. True or False:
   Aspera log entries contain data about disk I/O performance during transfer sessions.

Figure 8-9. Review questions

IBM Training

Review answers

1. Which of the following statement is most accurate regarding upgrading IBM Aspera High-Speed Transfer Server software? Select all that apply:
   A. It is recommended to remove the */opt/aspera* directory on Linux systems and *C:\program files\aspera* on Windows systems before starting an upgrade to ensure that all essential files are properly replaced.
   B. It is necessary to renew the Aspera license file after an upgrade.
   C. IBM Aspera High-Speed Transfer Server upgrades are implemented by running a new version of the installation software.
   D. Specific patches can be applied to upgrade specific components of the application.

   The answer is C

2. True or False:
   Aspera log entries contain data about disk I/O performance during transfer sessions
   The answer is True

Figure 8-10. Review answers

Unit 9. Course summary

Estimated time

00:30

Overview

This unit summarizes the course and provides information for future study.

IBM Training

Unit objectives

- Explain how the course met its learning objectives
- Access the IBM Training website
- Identify other IBM Training courses that are related to this topic
- Locate appropriate resources for further study

Figure 9-1. Unit objectives

IBM Training

Course objectives

- Describe the operation of the FASP protocol
- Outline the functions of various Aspera software products
- Explain Aspera configuration parameters and assign their values
- Create and manage Aspera users and groups
- Perform file transfers using the Aspera GUI and from the command line
- Implement support for Aspera Node API
- Configure Hot Folders and Aspera Watch Service
- Execute basic troubleshooting tasks for common problems

Figure 9-2. Course objectives

IBM Training

IBM badge

- Earn a Skills badge for this course by passing a quiz
- To earn the badge for this course: https://ibm-learning-skills-dev.github.io/ibm-learning-skills-dev.github.io/badges/IBM_Aspera.html
- Other IBM Cloud badges: https://ibm-learning-skills-dev.github.io/ibm-learning-skills-dev.github.io/badges/badgemain.html

Figure 9-3. IBM badge

IBM Training

IBM Professional Certifications

- By achieving an IBM Professional Certification, you can demonstrate your IBM Cloud product mastery to your employer or clients
- Certifications are a higher level of credential than a Skills badge for a single education course
- Product certifications demonstrate a strong knowledge of the product and typically require several months of work with the product
- IBM Cloud certifications are available for several roles, including developers, administrators, and business analysts
- For information on specific certifications and their requirements, see http://www.ibm.com/certify

Figure 9-4. IBM Professional Certifications

IBM Training

Other learning resources (1 of 4)

- **IBM Skills Gateway**
  - Search the new IBM Training and Skills website (formerly IBM Authorized Training website) to find and access the content you want.
  - https://www-03.ibm.com/services/learning/ites.wss/zz-en?pageType=page&c=a0011023

- **IBM Cloud Education Wiki Home**
  - Go to the wiki to find course abstracts, course correction documents, and curriculum development plans for IBM Cloud offerings.
  - https://www.ibm.com/developerworks

- **Role-based Learning Journeys**
  - Learning Journeys describe the appropriate courses, in the recommended order, for specific products and roles.
  - https://www-03.ibm.com/services/learning/ites.wss/zz/en?pageType=page&c=a000306

Figure 9-5. Other learning resources (1 of 4)

IBM Training

Other learning resources (2 of 4)

- **IBM Professional Certification Program**
  - IBM Professional Certification enables skilled IT professionals to demonstrate their expertise to the world. It validates skills and proficiency in the latest IBM technology and solutions.
  - https://www.ibm.com/certify

- **IBM Training blog, Twitter, and Facebook**
  - These official IBM Training and Skills accounts provide information about IBM course offerings, industry information, conference events, and other education-related topics.
  - https://www.ibm.com/blogs/ibm-training
  - https://twitter.com/IBMTraining
  - https://www.facebook.com/ibmtraining

Figure 9-6. Other learning resources (2 of 4)

IBM Training

Other learning resources (3 of 4)

- **Business Partner Technical Enablement Portal**
  - https://ibm.box.com/s/695khv9nyzekaorykqmsjrematz3v9xh
  - This program provides technical training content modules to IBM software partners (via PartnerWorld) and IBM Business Partners.

- **IBM Developer**
  - IBM's official developer program offers access to software trials and downloads, how-to information, and expert practitioners.
  - https://developer.ibm.com

- **IBM Education Assistant**
  - These multimedia educational modules help users gain a better understanding of IBM Software products and use them more effectively to meet business requirements.
  - https://www.ibm.com/products/software

Figure 9-7. Other learning resources (3 of 4)

IBM Training

Other learning resources (4 of 4)

- **IBM Knowledge Center**
  - The IBM Knowledge Center is the primary home for IBM product documentation.
  - https://www.ibm.com/support/knowledgecenter
- **IBM Marketplace**
  - IBM Marketplace is the landing page for all IBM Cloud products. Go to the Marketplace to learn about IBM offerings for Cloud, Cognitive, Data and Analytics, Mobile, Security, IT Infrastructure, and Enterprise and Business Solutions.
  - https://www.ibm.com/products
- **IBM Redbooks**
  - IBM Redbooks are developed and published by the IBM International Technical Support Organization (ITSO). Redbooks typically provide positioning and value guidance, installation and implementation experiences, typical solution scenarios, and step-by-step "how-to" guidelines.
  - http://www.redbooks.ibm.com

Figure 9-8. Other learning resources (4 of 4)

IBM Training

Unit summary

- Explain how the course met its learning objectives
- Access the IBM Training website
- Identify other IBM Training courses that are related to this topic
- Locate appropriate resources for further study

Figure 9-9. Unit summary

IBM Training

Course completion

You have completed this course:

IBM Aspera High-Speed Transfer Server Administration

Do you have any questions?

Figure 9-10. Course completion

IBM Training