

Thru Node for Distributed Hybrid Architecture

TECHNICAL INFORMATION

Version 1.3

Thru.[®]

Contents

Introduction	3
Using Thru Node in Your Network	4
Understanding Thru Flows	4
Installation	4
Upgrading Thru Nodes.....	4
Authentication & Network Firewall Permissions	5
Monitoring & Alerting	5
Thru Node Processing Features	5
Use Cases	5
Between Systems on a Network.....	5
Between Remote Networks and Central Network	5
System Requirements.....	6
Tested Operating Systems	6
Windows	6
Linux.....	6
Minimum Hardware Requirements to Run Thru Node.....	6
Processor	6
RAM	6
Hard Disk Space.....	6
Additional Notes.....	6

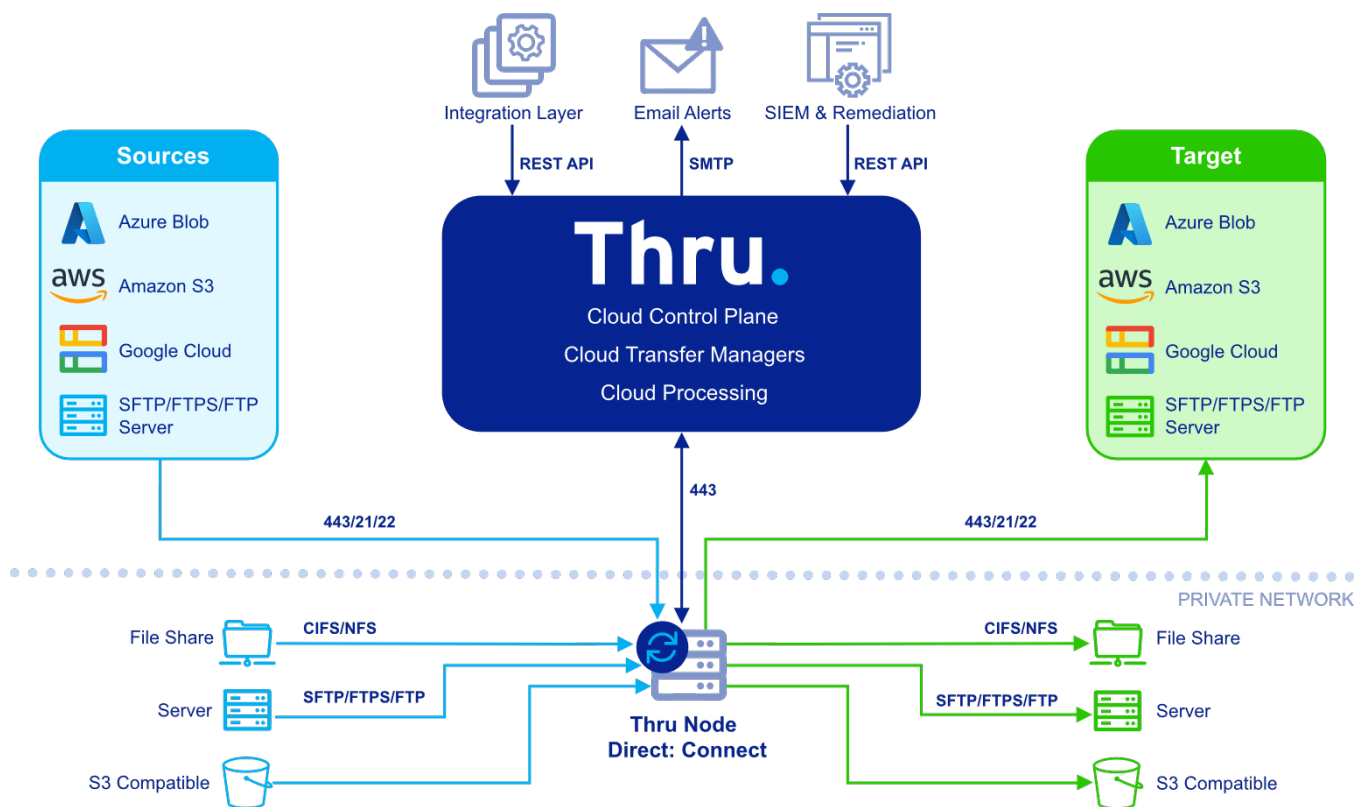
Introduction

Thru is a cloud-native managed file transfer (MFT) service that operates as the exchange for all file transfers as well as corresponding metadata. Thru is the control plane where all processes are configured. Furthermore, Thru records all file transfer transactions and activity to provide data governance and monitoring.

The Thru Node, also referred to as an MFT agent, is a key component in creating a distributed hybrid architecture for Thru's cloud-native managed file transfer workflow. The node or nodes, which are runtimes installed on a server or a cluster of servers, connect to Thru in the cloud.

The on-premises node complements the cloud solution by providing an instance on the network to orchestrate file transfers. Nodes can be used for files traversing a network and for file transfers between systems on a network or between the network and cloud. The primary benefit of this hybrid architecture is to maximize the use of a SaaS model and minimize the on-premises footprint for file transfers across the enterprise landscape.

Note: Thru Node only uses an outbound HTTPS connection for the data and control channels which is easier to manage than SFTP connections.



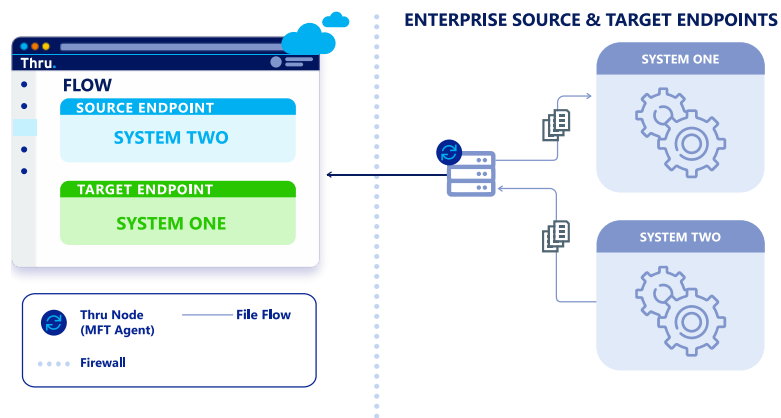
Using Thru Node in Your Network

Thru Node is a runtime installed on a machine within a network to connect to Thru over an outbound HTTPS connection, usually over port 443. In accordance with its workflow, simply called a *flow*, the node picks up and drops off files from any path or directory address that it has access rights to on the network.

Note: Thru Node does not need to be installed on the system from which it is picking up or dropping off files. As long as the node has access to the folder path from the machine it is installed on, the node can function.

Understanding Thru Flows

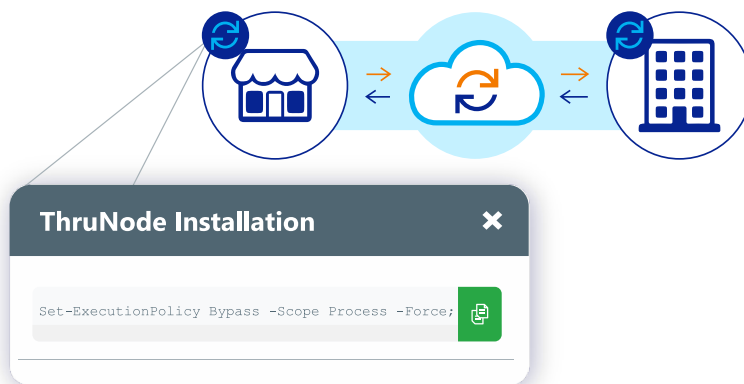
Flows are created using the graphical user interface, known as Thru Flow Studio, or via APIs. A flow is a directional workflow consisting of any number of source and target endpoints. A node can be an endpoint of a flow as a source, target or both. Each endpoint in a flow can have a set of specific processing and file routing rules.



When a flow's source and target locations are on the same network, the file remains on the network and is not exposed to the internet. The file is moved from the source to target location over the network's file sharing protocol such as NFS, SMB, etc. The metadata associated with the file transfer is recorded in Thru Audit and transmitted via an outbound connection over HTTPS.

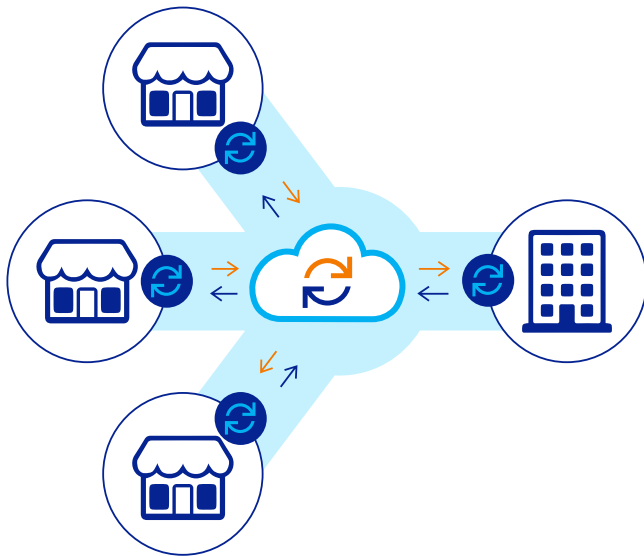
Installation

When a node is created in Thru, an installation script (Windows PowerShell or Linux Script) is made accessible via the GUI. The runtime package downloads and installs when the script is executed.



Upgrading Thru Nodes

Thru Nodes follow a continuous deployment pipeline and are automatically updated when a new release is pushed by Thru.



Authentication & Network Firewall Permissions

Thru Node connects to Thru in the cloud over HTTPS with token-based authentication. Connection direction is always from Thru Node to Thru in the cloud. This means no inbound ports are opened in the corporate firewall when the node is installed on an internal network.

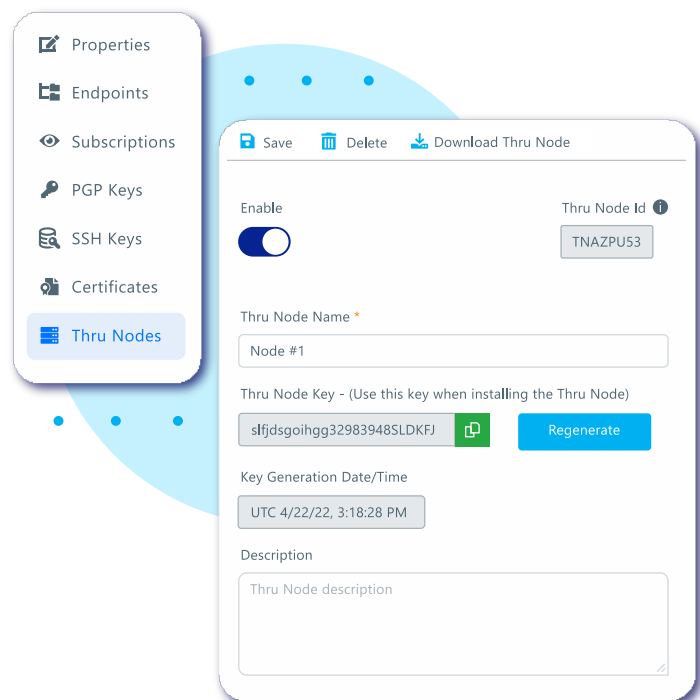
Monitoring & Alerting

Thru Node(s) connects with Thru in the cloud via an API and each node has a unique ID. Each time the node requests an update from Thru, it also reports its health in terms of being responsive or not. This status is available via an API that can be consumed by an event or IT service management tool for alerts and remediation.

Thru Node Processing Features

Thru node, in addition to transferring files, also has the capability to apply a range of rules based on the flow endpoint:

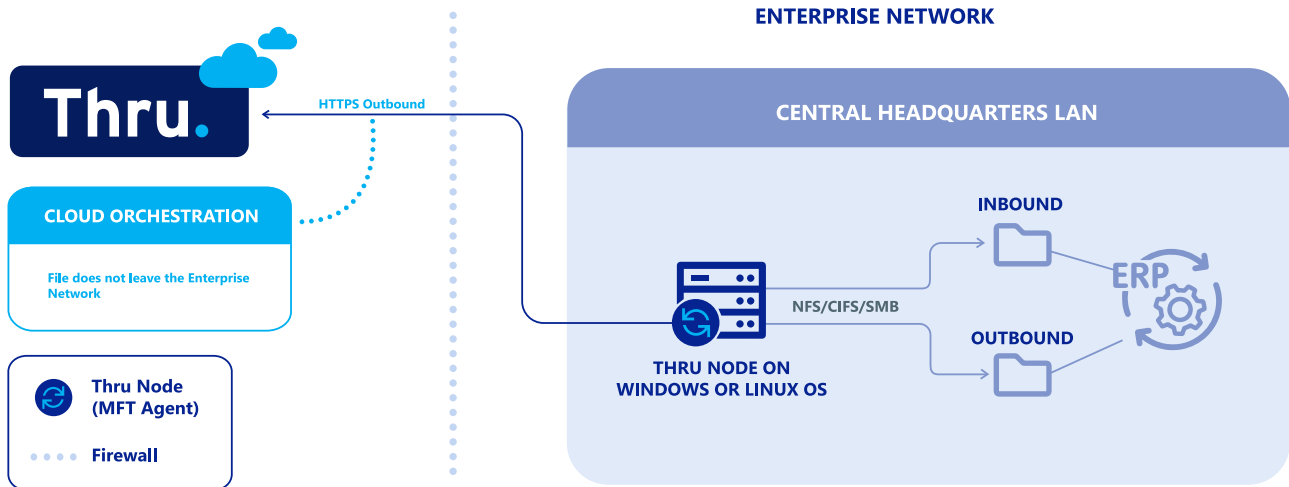
- Zip file
- Compress/decompress file
- Archive file at source
- Rename file at source or target
- Trigger files
- File filters
- Date filters



Use Cases

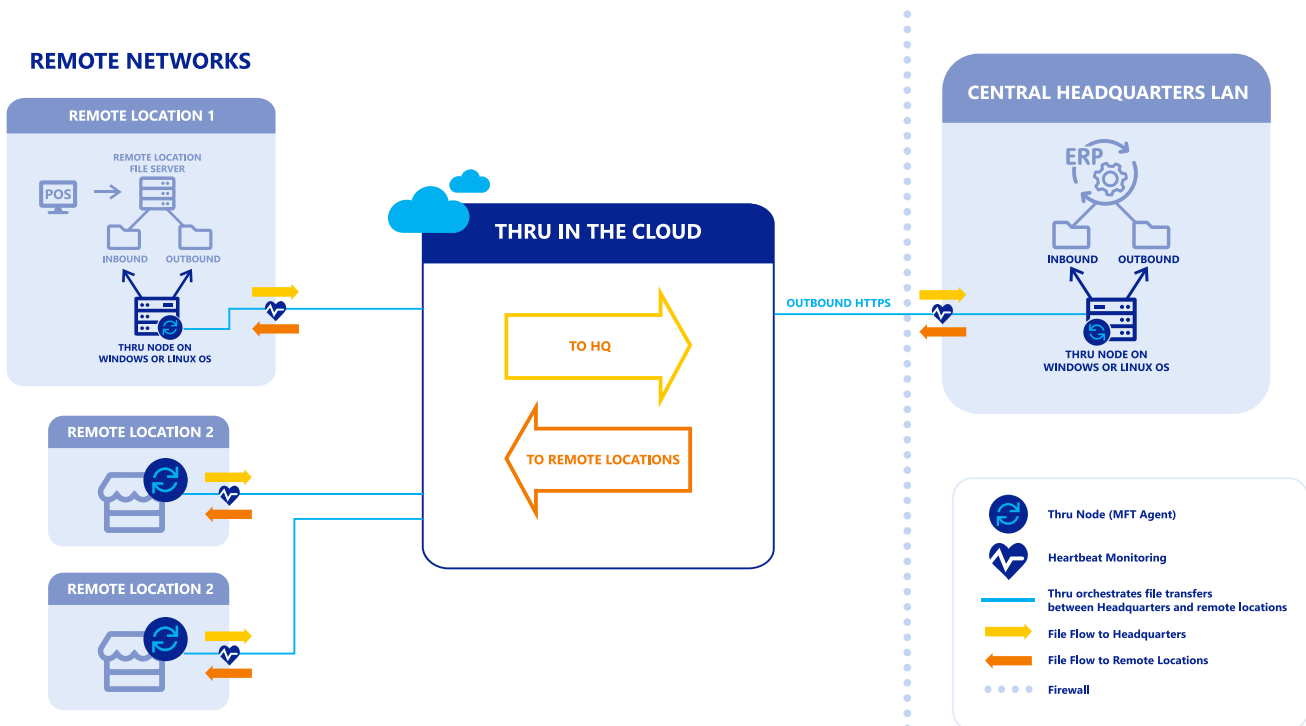
Between Systems on a Network

One of the common use cases when using Thru Node in the distributed hybrid architecture is to transfer files between systems in a local or wide area network (LAN or WAN). As long as the node is installed on a machine with access permissions to directories on the network, the node can orchestrate file transfers between the systems.



Between Remote Networks and Central Network

Use of Thru Node is also common in retail and distribution network landscapes where remote locations transfer daily business critical data files between core systems on the central network and many remote locations. In some cases, the remote locations number in the thousands.



System Requirements

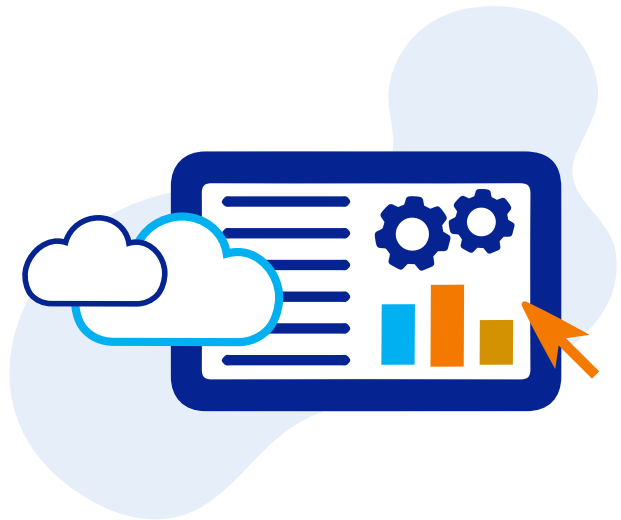
Tested Operating Systems

Windows

- Windows 10 / 11
- Windows Server 2012 (which is close to end of life)
- Windows Server 2019
- PowerShell 5.1 (or greater)
- Windows Task Scheduler

Linux

- Ubuntu 18.04 / 20.04 / 20.04.4
- RedHat 7.9 / 8.6 / 9.2



Minimum Hardware Requirements to Run Thru Node

Processor

- 1.4 GHz 64-bit processor or faster processor
- CPU cores - (2) cores for few flows, little processing, (4) cores for more flows and/or more processing

RAM

- 8 gigabytes (GB) with no file processing
- 16 gigabytes (GB) with file processing (encryption/decryption/compression/decompression)

Hard Disk Space

- Varies depending on the type of flow configurations
- For LAN transfers only, a minimum of 100GB is recommended
- For high file volumes and large files, storage may need to be as much as 1 TB
- Retention of files and configuration of retention policies will also affect storage requirements

Additional Notes

Thru Nodes currently do not communicate with each other, which means you cannot have one node monitoring a file path that then passes the file to another node to a different file path.

Nodes work independently accessing source and target filesystems.

Currently, multiple Thru Nodes cannot monitor a single file path. If two nodes are configured to pull files from the same path, they will end up in a race condition which could cause errors in file transfers. Thru will be adding a load balancing function in a future release.

Thru Nodes have built-in support to run on machines configured with proxies.