

User Guide

# Nimble Studio File Transfer



# Nimble Studio File Transfer: User Guide

Copyright © 2022 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

---

# Table of Contents

.....	<b>vi</b>
<b>What Is Nimble Studio File Transfer?</b> .....	<b>1</b>
Features of File Transfer .....	1
How to get started with File Transfer .....	3
Related services .....	3
Accessing File Transfer .....	3
Pricing for File Transfer .....	4
<b>Concepts and terminology</b> .....	<b>5</b>
Key concepts and terminology .....	5
<b>Setting up</b> .....	<b>8</b>
Sign up for an AWS account .....	8
Create a user with administrative access .....	9
Create a member account .....	10
Set up a studio in Nimble Studio .....	11
Create an S3 bucket .....	12
Create an IAM Access Policy .....	12
Set up the AWS CLI .....	16
<b>Getting started</b> .....	<b>18</b>
Prerequisites .....	18
Hardware configuration .....	18
Step 1: Install File Transfer .....	19
Step 2: Configure File Transfer .....	19
<b>Using the GUI</b> .....	<b>26</b>
File Transfer GUI overview .....	26
Upload files .....	27
Configuring hot folders .....	29
Jobs, Logs, and Report tabs .....	31
Download files .....	32
<b>Using the CLI</b> .....	<b>36</b>
Upload files .....	36
Flags .....	37
Configuring hot folders .....	42
Download files .....	44
Flags .....	45

---

<b>Checksums</b> .....	<b>52</b>
<b>Remote daemon</b> .....	<b>54</b>
Configure the remote daemon .....	54
Run a remote daemon .....	59
<b>Best practices</b> .....	<b>60</b>
Amazon Simple Storage Service (Amazon S3) .....	60
AWS Key Management Service (AWS KMS) .....	60
Hardware .....	60
Configuration .....	61
Autotuning configuration .....	61
Threads .....	61
Chunk size .....	62
Max active transfers .....	62
Checksums .....	61
Performance optimization .....	63
Network bandwidth .....	63
Disk throughput .....	63
Latency .....	64
Throttling .....	64
Maximum limit of open files .....	64
Bucket visibility .....	64
Optimize uploads (when not autotuning) .....	64
Configuration and database file location .....	65
Turning off the API server .....	65
<b>Monitoring</b> .....	<b>66</b>
Logging .....	66
Bucket report .....	67
<b>Troubleshooting</b> .....	<b>68</b>
Generate a support file .....	68
Troubleshooting the GUI .....	69
File Transfer is unable to connect after upgrading from v1.x to v2.0 .....	69
File Transfer is unable to connect .....	70
Troubleshooting the CLI .....	71
Expired or invalid credentials .....	71
Invalid transfer profile .....	71
TCP I/O .....	72

---

Absolute path .....	73
Unable to open connection .....	73
<b>Security .....</b>	<b>75</b>
<b>Support .....</b>	<b>76</b>
Amazon Nimble Studio Support .....	76
AWS Premium Support plans .....	76
AWS Support Center .....	76
<b>Release notes .....</b>	<b>77</b>
2.5.0 release notes .....	79
Major updates .....	79
Bug fixes .....	80
Known issues .....	80
2.1.0 release notes .....	80
Major updates .....	80
Bug fixes and minor updates .....	81
Known issues .....	81
2.0 release notes .....	81
Major updates .....	81
Bug fixes and minor updates .....	81
1.1.0 release notes .....	82
Major updates .....	82
Bug fixes and minor updates .....	82
<b>Document History .....</b>	<b>83</b>
<b>AWS Glossary .....</b>	<b>84</b>

**End of support notice:** On October 22, 2024, AWS will discontinue support for Nimble Studio File Transfer. After October 22, 2024, you will no longer be able to access the Nimble Studio File Transfer console or Nimble Studio File Transfer resources.

# What Is Nimble Studio File Transfer?

Nimble Studio File Transfer is a file transfer feature provided by Amazon Nimble Studio. File Transfer accelerates media asset transfer workflows into and out of Amazon Simple Storage Service (Amazon S3).

Digital imaging technicians (DIT) and content creators can use File Transfer without needing AWS expertise. With File Transfer, you can transfer on-set camera data or final production archive data directly into Amazon Simple Storage Service (Amazon S3) buckets. File Transfer can move thousands of files, including large media files, while preserving hierarchy structure. File Transfer provides a native graphical user interface (GUI) for digital creatives.

With File Transfer, you can transfer digital media between Amazon S3 and local storage over public and private network connections. You can move thousands of files from on-premises to AWS, and move them to different AWS Regions. File Transfer works for any file system to Amazon S3 transfer. This means that you can use File Transfer on an Amazon Elastic Compute Cloud (Amazon EC2) instance to move data from Amazon Elastic Block Store (Amazon EBS) to Amazon S3. This is useful for maintaining consistency throughout your pipeline.

Studios can use File Transfer for camera to cloud, work in-progress synchronization, final media delivery, and archival workflows. All file transfers are encrypted at rest and in-transit based on how you define your Amazon S3 encryption policies. You can choose to use Amazon S3 managed keys (SSE-S3) or your own keys stored in AWS Key Management Service (AWS KMS). If you choose your own S3 buckets, you can set up your own bucket policies and encryption with Amazon S3 Standard AES-256 encryption or with a custom KMS key.

## Topics

- [Features of File Transfer](#)
- [How to get started with File Transfer](#)
- [Related services](#)
- [Accessing File Transfer](#)
- [Pricing for File Transfer](#)

## Features of File Transfer

File Transfer has the following features:

- **Upload and Download files to and from Amazon S3** – End users can select files and folders to upload and download to and from their local file systems to and from Amazon S3. Uploads traverse the files and folders on the on-premise network and select the Amazon S3 bucket destination. Downloads traverse the files and folders on the selected Amazon S3 bucket and select the on-premise network destination.
- **Drag and Drop Graphical User Interface (GUI)** – With the GUI, you can drag and drop files while uploading to and downloading from Amazon S3.
- **Command Line Interface (CLI)** – The File Transfer CLI gives users more control over File Transfer configuration parameters, adjustments, flags, and more.
- **High speed file transfers** – File Transfer offers parallelization and autotuning for maximum performance. Autotuning automates the process of tuning your chunk size and max active transfers based on the file sizes.
- **Jobs control table** – Monitor active transfer jobs and control them through the Jobs table. File Transfer supports Cancel/Pause/Resume controls, the Rename and Generate report functions, as well as a more detailed view of your transfer jobs via Job Details.
- **Checksum** – Verifies the integrity of files transferred to Amazon S3.
- **Bucket reports** – This feature lets users export a report of the files and folders in the Amazon S3 bucket without needing to log into the Amazon S3 console.
- **Upload hot folder** – Designate a folder or folders on local storage for File Transfer to monitor. Whenever you add new content to that folder, File Transfer automatically uploads that content to Amazon S3.
- **Remote daemon** – You can use the remote daemon to start a daemon that a GUI running on a different machine can connect to. This is useful if you have multiple people working on the same File Transfer application.
- **Bookmarks** – Connect to a different computer that is running the remote daemon. This means that a GUI user can connect to a remote machine, access their file system, and start transfers from that remote machine.
- **Multiple AWS Regions** – Amazon S3 is a global resource and isn't bound by AWS Region availability. You can use File Transfer anywhere that Amazon S3 is available. For more information about Amazon S3 endpoints, see [Amazon Simple Storage Service endpoints and quotas](#) in the AWS General Reference.



**Note**

You must have a working Nimble Studio to access File Transfer. Nimble Studio is only supported in the AWS Regions listed in [Availability Zones for Amazon Nimble Studio](#). After you create a studio, you can use File Transfer in any AWS Region that Amazon S3 is supported in. There is no cost impact for using File Transfer in a different Region than your Nimble Studio studio.

## How to get started with File Transfer

After you familiarize yourself with the [Concepts and terminology for Nimble Studio File Transfer](#) page, proceed to the [Getting started with Nimble Studio File Transfer](#) page. That page contains helpful information and step-by-step instructions about how to set up File Transfer and how to configure File Transfer for your team. The tutorials show how to start uploads and downloads by using File Transfer.

## Related services

- [Nimble Studio](#)
  - File Transfer is a feature of Nimble Studio that provides high speed file transfers and data management.
- [Amazon S3](#)
  - File Transfer uses Amazon S3 as its cloud storage. You can choose your own S3 buckets and set your own bucket policies and encryption with either Amazon S3 Standard AES-256 encryption, or with a custom KMS key.
- [IAM](#)
  - File Transfer uses AWS Identity and Access Management (IAM) to authorize who has permission to access the S3 bucket. To use File Transfer, you are required to create an IAM access policy.

## Accessing File Transfer

The File Transfer installer can be accessed from the Nimble Studio console or from [Step 1: Install File Transfer](#) in the *Getting started* documentation.

You can interact with File Transfer by using the command line interface (CLI) or the graphical user interface (GUI). The File Transfer CLI gives you more control over File Transfer configuration parameters, adjustments, flags, and more. The GUI displays reporting on all transfers. You can also start uploads and downloads from the GUI.

## Pricing for File Transfer

File Transfer is provided at no additional cost for customers. Amazon S3 standard rates for data transfers and storage still apply. For information about pricing, see the [Amazon S3 pricing](#) page.

# Concepts and terminology for Nimble Studio File Transfer

This guide introduces you to key concepts and terminology for understanding and using Nimble Studio File Transfer.

## Contents

- [Key concepts and terminology](#)

## Key concepts and terminology

**Nimble Studio File Transfer** – File Transfer is a file transfer tool for accelerating media asset transfer workflows into and out of Amazon Simple Storage Service (Amazon S3).

**Amazon Nimble Studio console** – The [Nimble Studio](#) console is a portion of the AWS Management Console that is devoted to our admin IT customers. This console is where admins create their cloud studio and manage many settings.

**File Transfer GUI** – With the File Transfer GUI, you can transfer files to and from Amazon S3 and view data about your transfers.

**Session** – A session is a period of time in which you can upload files or download files from Amazon S3 by using File Transfer. Your session status is indicated by the check mark icon next to the remote configuration for your Amazon S3 Bucket. You must have an active session to transfer files.

**Job queue** — When you start a transfer, File Transfer displays a list of transfer jobs. This list corresponds to the individual files that you selected for transfer. You can find the following information in the **Job queue** section for both uploads and downloads.

- **Filter:** Filter by transfer status to adjust which files are displayed in the upload and download queues.
- **File name:** File name of individual file being uploaded. Selecting this will toggle where an individual file's name or file path is displayed.
- **Checksum:** Validates that the file is still unmodified at a future date.
- **Active:** Reports the current amount of data uploaded and downloaded across all jobs in your session.

- **Avg. Speed:** Reports the average speed of all file uploads and downloads in your session.
- **Session Total:** Reports the total amount of all planned data uploaded and downloaded for all jobs in your session.
- **Size:** Reports the total size of the job.
- **ETA:** Reports the estimated completion time of a job.
- **Start time:** Reports when a job was started.
- **Progress:** Reports the status of a given job.

**Remote configuration** – Remote configurations are different configurations available to transfer files to different buckets or directories. Use remote configurations to differentiate between different destinations and different teams for the same, or for different, productions.

**Amazon Simple Storage Service** – [Amazon Simple Storage Service \(Amazon S3\)](#) is an object storage service that offers scalability, data availability, security, and performance. File Transfer uploads files to Amazon S3 by using Amazon S3 APIs. All CloudTrail, CloudWatch, and CloudFormation information about File Transfer is logged as Amazon S3 usage.

File Transfer is like an improved Amazon S3 transfer experience. File Transfer provides better performance than the AWS Command Line Interface (AWS CLI) and performs checksumming on your uploads.

**AWS Identity and Access Management** – [AWS Identity and Access Management \(IAM\)](#) is a web service that helps you securely control access to AWS resources. With IAM, you can centrally manage permissions that control which AWS resources users can access. You use IAM to control who is authenticated (signed in) and authorized (has permissions) to use resources.

File Transfer relies on IAM to limit who has access to your Amazon S3 bucket.

**AWS managed policies** – An AWS managed policy is a standalone policy that is created and administered by AWS. Standalone policy means that the policy has its own Amazon Resource Name (ARN) that includes the policy name. For example, `arn:aws:iam::aws:policy/IAMReadOnlyAccess` is an AWS managed policy. For more information about ARNs, see [IAM ARNs](#) in the *IAM User Guide*.

AWS managed policies are used for granting permissions to common job functions. Job function policies are maintained and updated by AWS when new services and API operations are introduced. For example, the `AdministratorAccess` job function provides full access and permissions

delegation to every service and resource in AWS. Partial-access AWS managed policies such as `AmazonMobileAnalyticsWriteOnlyAccess` and `AmazonEC2ReadOnlyAccess` can provide specific levels of access to AWS services without allowing full access. To learn more about access policies, see [Understanding access level summaries within policy summaries](#) in the *IAM User Guide*.

**AWS Regions** – File Transfer is available in all global Regions. Users close to the Region where your S3 bucket is located will experience faster upload and download speeds. For more information, see [Amazon Simple Storage Service endpoints and quotas](#) in the AWS General Reference. To see the mapping of IDs to Availability Zones in your account, see [AZ IDs for Your Resources](#) in the *AWS RAM User Guide*.

### Note

You must have a working Nimble Studio to access File Transfer. Nimble Studio is only supported in the AWS Regions listed in [Availability Zones for Amazon Nimble Studio](#). After you create a studio, you can use File Transfer in any AWS Region that Amazon S3 is supported in. There is no cost impact for using File Transfer in a different Region than your Nimble Studio studio.

**Availability Zone (AZ)** – Availability Zones are multiple, isolated locations within each AWS Region. An AZ is represented by an AWS Region code followed by a letter identifier. For example: us-east-1a

# Setting up Nimble Studio File Transfer

Before you use Nimble Studio File Transfer for the first time, complete the following tasks.

## Topics

- [Sign up for an AWS account](#)
- [Create a user with administrative access](#)
- [Create a member account](#)
- [Set up a studio in Nimble Studio](#)
- [Create an S3 bucket](#)
- [Create an IAM Access Policy](#)
- [Set up the AWS CLI](#)

## Sign up for an AWS account

If you do not have an AWS account, complete the following steps to create one.

### To sign up for an AWS account

1. Open <https://portal.aws.amazon.com/billing/signup>.
2. Follow the online instructions.

Part of the sign-up procedure involves receiving a phone call and entering a verification code on the phone keypad.

When you sign up for an AWS account, an *AWS account root user* is created. The root user has access to all AWS services and resources in the account. As a security best practice, assign administrative access to a user, and use only the root user to perform [tasks that require root user access](#).

AWS sends you a confirmation email after the sign-up process is complete. At any time, you can view your current account activity and manage your account by going to <https://aws.amazon.com/> and choosing **My Account**.

## Create a user with administrative access

After you sign up for an AWS account, secure your AWS account root user, enable AWS IAM Identity Center, and create an administrative user so that you don't use the root user for everyday tasks.

### Secure your AWS account root user

1. Sign in to the [AWS Management Console](#) as the account owner by choosing **Root user** and entering your AWS account email address. On the next page, enter your password.

For help signing in by using root user, see [Signing in as the root user](#) in the *AWS Sign-In User Guide*.

2. Turn on multi-factor authentication (MFA) for your root user.

For instructions, see [Enable a virtual MFA device for your AWS account root user \(console\)](#) in the *IAM User Guide*.

### Create a user with administrative access

1. Enable IAM Identity Center.

For instructions, see [Enabling AWS IAM Identity Center](#) in the *AWS IAM Identity Center User Guide*.

2. In IAM Identity Center, grant administrative access to a user.

For a tutorial about using the IAM Identity Center directory as your identity source, see [Configure user access with the default IAM Identity Center directory](#) in the *AWS IAM Identity Center User Guide*.

### Sign in as the user with administrative access

- To sign in with your IAM Identity Center user, use the sign-in URL that was sent to your email address when you created the IAM Identity Center user.

For help signing in using an IAM Identity Center user, see [Signing in to the AWS access portal](#) in the *AWS Sign-In User Guide*.

## Assign access to additional users

1. In IAM Identity Center, create a permission set that follows the best practice of applying least-privilege permissions.

For instructions, see [Create a permission set](#) in the *AWS IAM Identity Center User Guide*.

2. Assign users to a group, and then assign single sign-on access to the group.

For instructions, see [Add groups](#) in the *AWS IAM Identity Center User Guide*.

## Create a member account

### Note

Skip this step if you're setting up Nimble Studio in your management account.

If you're an IT administrator with an AWS member account and you're trying to set up Nimble Studio, your administrative user must first grant the correct access and permissions to that member account.

You can set up Nimble Studio in a management account or a member account as long as that account is in an organization from AWS Organizations. An organization has a single *management account*. The central features of the organization are configured and enforced by the management account. *Member accounts* set up and use different services. For more information about management accounts and member accounts, see [AWS Organizations terminology and concepts](#).

In addition, AWS IAM Identity Center must be enabled in the organization. IAM Identity Center can only be enabled in the management account, and the studio must be in the same AWS Region as IAM Identity Center. To enable IAM Identity Center in your organization, follow the instructions in [Enable IAM Identity Center](#).

### Note

If you attempt to set up a studio in a member account without IAM Identity Center enabled, the member account won't be able to enable IAM Identity Center themselves. In that case, the member account must ask their enterprise IT to configure IAM Identity Center in their AWS Organization.



## To create a member account with permission to create a studio

1. Use an existing member account or create a new one by following the instructions in [Add users](#) in the AWS IAM Identity Center User Guide.
  - This member account must belong to the organization that is setting up your studio in Nimble Studio.
2. Delegate administrator access to the member account by following the instructions in [Register a member account](#).
  - Delegated administrator access is an IAM Identity Center feature. Delegated administrator access is unrelated to IAM administrator access. Someone can have full administrator permissions to access their account but not have delegated administrator access from the management account.

Your IT administrator can now complete the following steps in the next sections.

## Set up a studio in Nimble Studio

If you already have a Nimble Studio cloud studio, skip this step.

### Note

File Transfer does not require your Amazon S3 buckets to be associated with a Nimble Studio. File Transfer only requires a Nimble Studio as the tool is available to use at not additional cost for Nimble Studio customers only.

To create a studio, follow the instructions in [Setting up Nimble Studio](#). Make sure that the following information is true when you set up your studio.

- Set up your Nimble Studio in a management account, or in a member account with IAM Identity Center delegated admin access.
- In *step 5* of [Step 1: Configure studio infrastructure](#), choose the AWS Region that you enabled IAM Identity Center in.

## Create an S3 bucket

Before you can use File Transfer, you must complete the [Setting up Amazon S3](#) tutorial. If Amazon S3 isn't properly configured, the security of the contents in your bucket could become compromised.

You must also complete the [Create your first S3 bucket](#) tutorial. This creates an S3 bucket for you to upload and download files from.

- (Recommended) In *step 8*, **Enable Bucket Versioning**.
  - This makes sure that your data isn't lost if you accidentally overwrite a file in Amazon S3 with a new version.
  - Enabling bucket versioning accrues additional cost. For more information about Amazon S3 pricing, see the [Amazon S3 pricing](#) page.
- (Recommended) In *step 11*, for **Encryption key type**, choose **AWS Key Management Service key (SSE-KMS)**.
  - If you don't have an SSE-KMS key, create one by following the instructions in the [Creating symmetric encryption KMS key](#) tutorial.
  - For more information about the difference types of keys, see the [Customer keys and AWS keys page](#) in the AWS Key Management Service Developer Guide. To allow someone to use the bucket from another AWS account, you must use a customer managed key. It's difficult to change the key after you create the bucket, so make sure that you create your bucket with the correct keys.
- Leave all other settings and user preferences at their defaults.

## Create an IAM Access Policy

Next, you must create an IAM access policy that gives permission to the Amazon S3 bucket that you created in the [Create an S3 bucket](#) section. After that, you'll attach this IAM policy to an IAM user. This IAM user will generate the credentials that File Transfer needs to access the Amazon S3 bucket.

Follow the [Creating policies on the JSON tab](#) tutorial in the *IAM User Guide* and use the following JSON policy document. The policy that you need to use depends on what type of AWS KMS key that you chose.

## Using an AWS KMS key (SSE-KMS)

- Enter the following text in the JSON template to provide the required access for Amazon S3 uploads and downloads.
- To allow deleting objects in the S3 bucket, include the actions listed in the statement with Sid "OptionalActions" from the following text. You don't need to include those actions if you don't want to allow deleting S3 objects.

```
{
  "Statement": [
    {
      "Sid": "ListBucketContents",
      "Action": [
        "s3:ListBucket",
        "s3:GetBucketLocation",
        "s3:PutObject",
        "s3:GetObject",
        "s3:GetObjectTagging"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3::bucket-name",
        "arn:aws:s3:::bucket-name/*"
      ],
    },
    {
      "Sid": "KMSKeyAccess",
      "Action": [
        "kms:GenerateDataKey*",
        "kms:Encrypt",
        "kms:Decrypt"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:kms:key-region:account-number:key/key-id"
    },
    {
      "Sid": "OptionalActions",
      "Action": [
        "s3:DeleteObject",
        "s3:DeleteObjectVersion",
        "s3:ListBucketVersions",
        "s3:AbortMultipartUpload"
      ]
    }
  ]
}
```

```

    ],
    "Effect": "Allow",
    "Resource": [
        "arn:aws:s3:::bucket-name",
        "arn:aws:s3:::bucket-name/*"
    ],
  }
],
"Version": "2012-10-17"
}

```

- Replace *bucket-name* with the name of the bucket that you created in [Create an S3 bucket](#).
- Replace *key-region* with the AWS Region that you created your key in.
- Replace *account-number* with your AWS account number.
- Replace *key-id* with the ID of the KMS key that you chose in *step 2* of [Create an S3 bucket](#).
  - To find the KMS key ID, follow the instructions in **To view the S3 Bucket Key setting for your bucket** in the [Viewing the settings for an S3 Bucket Key](#) tutorial.
  - Choose the bucket that you created in [Create an S3 bucket](#).
  - Find the **AWS KMS key ARN** in the **Default encryption** section. The KMS key ID is the last portion of the ARN.

### Using an Amazon Managed KMS key (SSE-S3)

- Check if your bucket has an active KMS key.
  - Follow the instructions in **To view the S3 Bucket Key setting for your bucket** in the [Viewing settings for an S3 Bucket Key](#) tutorial.
  - If you aren't using a KMS key, you can proceed to *step 2*.
  - If you do have a KMS key attached to the bucket, follow the instructions in **To use an AWS KMS key (SSE-KMS)**.
- Enter the following text in the JSON template to provide the required access for Amazon S3 uploads and downloads.
- To allow deleting objects in the S3 bucket, include the actions listed in the statement with Sid "OptionalActions" from the following text. You don't need to include those actions if you don't want to allow deleting S3 objects.

```

{
  "Statement": [

```

```
{
  "Sid": "ListBucketContents",
  "Action": [
    "s3:ListBucket",
    "s3:GetBucketLocation",
    "s3:PutObject",
    "s3:GetObject",
    "s3:GetObjectTagging"
  ],
  "Effect": "Allow",
  "Resource": [
    "arn:aws:s3::bucket-name",
    "arn:aws:s3:::bucket-name/*"
  ],
},
{
  "Sid": "OptionalActions",
  "Action": [
    "s3:DeleteObject",
    "s3:DeleteObjectVersion",
    "s3:ListBucketVersions",
    "s3:AbortMultipartUpload"
  ],
  "Effect": "Allow",
  "Resource": [
    "arn:aws:s3::bucket-name",
    "arn:aws:s3:::bucket-name/*"
  ],
}
],
"Version": "2012-10-17"
}
```

- Replace *bucket-name* with the name of the bucket that you created in [Create an S3 bucket](#).

You've now created an IAM policy that grants permission to the S3 bucket that you created in [Create an S3 bucket](#).

# Set up the AWS CLI

Install and configure the AWS CLI if you haven't already. File Transfer uses the AWS Command Line Interface (AWS CLI) named profiles only to handle and store IAM credentials. For more information, see [Getting started with the AWS CLI](#).

1. To install or upgrade the AWS CLI on your local machine, follow the instructions in [Installing the AWS Command Line Interface version 2](#) in the *AWS Command Line Interface User Guide*.
2. Configure the AWS CLI by following the instructions in [Setting up new configuration and credentials](#).
3. Verify the installation or upgrade by running `aws nimble help`. This command displays a list of available Nimble Studio commands.
4. Create a named profile by following the instructions in [Using named profiles](#). This named profile will be used to configure File Transfer in the [Getting started with Nimble Studio File Transfer](#) section.
  - a. To create an access key and a secret key, follow the instructions in [Creating an IAM user in your AWS account](#). After you create a user, the console generates an access key and secret key value.
  - b. In *step 4*, choose **Command Line Interface (CLI)** for the type of access this user will have.
  - c. In *step 6*, select **Attach existing policies directly**. Select the check the box for the policy that you made in [Create an IAM Access Policy](#).
5. Verify that you created a named profile by running the following command: `aws --profile [name of profile you created in step 4] sts get-caller-identity`
  - This command should generate an output similar to the following output example. In this example, the profile is named `filetransfer`.

```
$ aws --profile filetransfer sts get-caller-identity
  "UserId": "ARXXXXXXXXXXXXXXXXXXXX:username",
  "Account": "123456789012",
  "Arn": "arn:aws:sts::123456789012:XXXXXXXXXXXXXXXXX..."
}
```

We recommend that you read about additional AWS CLI security controls that are available in the [AWS Command Line Interface User Guide](#).

# Getting started with Nimble Studio File Transfer

This tutorial shows how to install and configure File Transfer. Before you begin the following steps, make sure that the [Setting up Nimble Studio File Transfer](#) tutorial has been completed by your administrator.

If you encounter any problems while following this tutorial, see the [Support for Nimble Studio File Transfer](#) page.

## Topics

- [Prerequisites](#)
- [Step 1: Install File Transfer](#)
- [Step 2: Configure File Transfer](#)

## Prerequisites

Complete the [Setting up Nimble Studio File Transfer](#) section before installing and configuring File Transfer.

## Hardware configuration

We recommend that your computer meets the following requirements for you to use File Transfer. For more information about how to increase the speed of your transfers, see [Performance optimization](#).

- 8 logical CPU cores
- 8 GB RAM

Your transfer speeds depend on your hardware, network configuration, and bandwidth. File Transfer can transfer files as your network and hardware permits. For example, if your machine has been allocated a network bandwidth of 500 Mbps, the fastest that File Transfer can try to complete transfers is 500 Mbps.



## Step 1: Install File Transfer

The File Transfer installer will guide you through the setup, and it will set the correct threads and chunk size based on your machines.

Download and install the Nimble Studio File Transfer client from [File Transfer clients](#).

## Step 2: Configure File Transfer

With File Transfer, you can either use the graphical user interface (GUI) or the command line interface (CLI). If you use an operating system (OS) with a desktop, you can configure File Transfer in the GUI. If you aren't using an OS with a desktop, you can use the File Transfer CLI to configure and run File Transfer. The CLI provides more flexibility for how you use File Transfer.

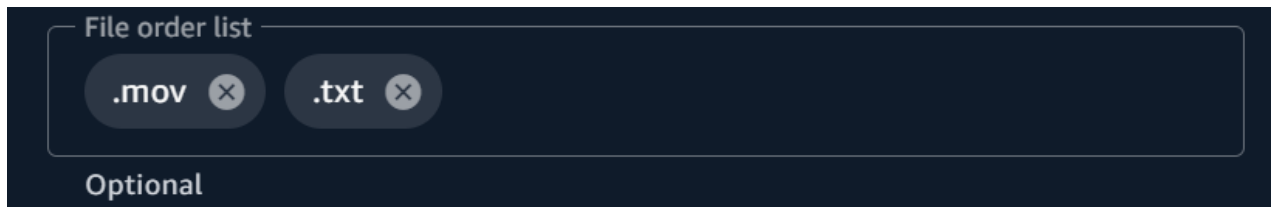
### GUI

Before you can transfer files with File Transfer, you must add a *remote configuration*. Remote configurations are treated as different configurations that you can use to transfer files to different buckets and directories.

#### To add a remote configuration

1. Choose **Create a Remote Configuration** on the S3 Bucket file browser window of the GUI.
2. In the **Add Remote Configuration** section, enter the following information.
  - a. In **Remote Configuration name**, enter a name for the remote configuration. The name can't be modified once it is set.
  - b. In **S3 Bucket**, add the name of the S3 bucket you want to use for File Transfer. To create an S3 bucket, see [Create an S3 bucket](#).
  - c. In **AWS Region**, enter the region where your S3 bucket is located.
  - d. In **AWS named profile**, enter an AWS named profile to use for S3 access. If you choose to use an AWS Access Key or an AWS Secret Key instead, leave this field blank. If you do not have an AWS named profile, see the [Create an IAM Access Policy](#) and [Set up the AWS CLI](#) topics.
3. Choose **Add** to continue.
4. In the **Advanced** section, enter the following information.

- a. **Storage Class** specifies the storage class of the S3 Bucket. For information on storage classes, see [Amazon S3 Storage Classes](#)
- b. **Checksum Algorithm** specifies which checksum algorithm is used for the checksum computations.
- c. (Optional) **Local directory** specifies the default directory on your local machine.
- d. (Optional) **S3 bucket prefix** specifies the default directory of the S3 bucket.
- e. **Filter** specifies a filter for files based on format. This field accepts valid regular expressions. For example, "`^\.*\.(mov)$`" only uploads files ending in `.mov`.
- f. **Max Age** Limits File Transfer to files created or modified after the Max Age time value and before the current time. Units are expressed as **m** (minutes), **h** (hours), **d** (days), and **w** (weeks). If no units are supplied, the value is specified in seconds. For example, an input of **5d** transfers files created or modified in the last five days. An input of **350** transfers files created or modified in the last 350 seconds.
- g. (Optional) **File order list** sets transfer priority by file extension. Enter an extension to add it to the list. Files with extensions that aren't in the list are the lowest priority and are transferred in the order they are listed in the file system.



- h. **Transfer autotuning** dynamically sets the number of threads and chunk size for file transfers when enabled. This is enabled by default. We recommend that you keep this option enabled.

### Number of threads

Number of individual threads that are used to transfer each individual file.

### Chunk size

Size of the chunk (in megabytes) that is delivered by each thread.


- i. Enabling **Use S3 Transfer Acceleration** improves the speed of transfer for large files when enabled. The best practice is to keep S3 Transfer Acceleration off because it isn't required to achieve high speed transfers in File Transfer. If the bucket is geographically far from you, turn on this feature. For more information, see [S3 Transfer Acceleration](#).

If this is enabled, additional fees might apply. For more information, see [Amazon S3 Pricing](#).

- j. **Enable metadata filter** automatically filters system metadata files when enabled. These files include files that start with `._` as well as `thumbs.db` and `.DS_Store` files.

## 5. Choose **Save**

### To add a studio ID

1. Open File Transfer.
  - a. Go to the **Start Menu** and search for **File Transfer**.
  - b. Select **Nimble Studio File Transfer** from the list.
2. From the dropdown menu , choose **Settings**.
3. Enter the studio ID that you found in the [Set up a studio in Nimble Studio](#) tutorial.
4. Choose **Save**.

## CLI

Now that File Transfer is installed, edit the configuration file.

### To edit the configuration file


1. Open the configuration file with any text editing software on your computer.
  - a. Windows: Navigate to the `User/<your username>` folder on your computer. Open the `.filetransfer` folder and open the `filetransfer.yaml` file with a text editor.
  - b. macOS: Enter **Cmd+Shift+G**. Then enter `~/filetransfer`. Open the `filetransfer.yaml` file with a text editor.
  - c. Linux: Open the `filetransfer.yaml` file using any text editor. The file is located in `~/filetransfer/configuration.yaml`.
2. Define the following variables in the configuration file. The required values are populated. You must provide the optional values.

- a. `studio_id`: Enter the studio ID that you found in the [Set up a studio in Nimble Studio](#) tutorial.
  - b. `max_active_checksums`: Enter the number of individual checksums that are processed at the same time. If this value exceeds the number of available CPU cores in the system, a warning message displays and the value is capped to the number of CPU cores. For more information, see [File Transfer checksums](#).
  - c. `max_active_transfers`: Enter the number of individual files that are processed at the same time.
3. Define at least one remote configuration in the configuration file. The required values for the remote configuration aren't populated, so you must manually enter them. Define the following variables for each remote configuration.

```
protocols:
  s3:
    remote_configuration:
      demo:
        name: demo
        bucket: my-bucket
        region: us-west-2
        profile: my-profile
        storage_class: standard
        auto_tuning: true
        chunk_size: 25
        threads: 10
        checksum_algorithm: md5_hex
        max_age: ""
        accelerated: false
        file_order: []
        filter: ""
        upload_hot_folder:
          enabled: false
          local_source_folder: ""
          s3_destination_folder: ""
        enable_metadata_filter: true
        paths:
          local: ""
          remote: ""
```

4. `name`: Enter a name for your transfer profile.

5. **bucket:** Enter the name of the S3 bucket that you want to upload to and download from. Your administrator should have created the bucket in [Create an S3 bucket](#).
6. **region:** Enter the AWS Region that your bucket is located in.
7. **profile:** Enter the name of the profile that your admin created in *step 4* of [Set up the AWS CLI](#). This profile allows you to access Amazon Simple Storage Service (Amazon S3). To get a list of configured profiles, run the following command in a terminal window: `aws configure list`
8. **storage\_class:** By default, this is set to `standard`. The accepted values for this variable are as follows:
  - `reduced_redundancy`
  - `standard_ia`
  - `onezone_ia`
  - `intelligent_tiering`
  - `glacier`
  - `deep_archive`
  - `glacier_ir`

 **Note**

You can upload directly to any storage class in Amazon S3. If you use S3 Glacier Deep Archive or S3 Glacier Flexible Retrieval, you can't download objects you uploaded with File Transfer from File Transfer directly. For more information about storage classes, see [Amazon S3 Storage Classes](#).

9. **auto\_tuning:** Automatically adjusts chunk size and max active transfers based on the size of the file. By default, this is set to `true`.
10. **chunk\_size:** Enter the size of the chunk (in MB) that's delivered by each thread. This field isn't required if `auto_tuning` is set to `true`.
11. **threads:** Enter the number of individual threads that is used to transfer each individual file. This field isn't required if `auto_tuning` is set to `true`.
12. **checksum\_algorithm:** This is the checksum algorithm that will be used when uploading your files to S3. You can choose `md5-hex` (the default value), `xxhash`, `xxhash64`, and `xxh3` checksum algorithms.

13. (Optional) `Max_Age`: Limits File Transfer to files created or modified after the `Max_Age` time value and before the current time. Units are expressed as **m** (minutes), **h** (hours), **d** (days), and **w** (weeks). If no units are supplied, the value is specified in seconds. For example, an input of **5d** transfers files created or modified in the last five days. An input of **350** transfers files created or modified in the last 350 seconds.
14. `accelerated`: Enables S3 Transfer Acceleration. By default, this is `false`. The best practice is to keep S3 Transfer Acceleration off because it isn't required to achieve high speed transfers in File Transfer. If the bucket is geographically far from you, turn on this feature. For more information, see [S3 Transfer Acceleration](#). If this is enabled, additional fees might apply. For more information, see [Amazon S3 Pricing](#).
15. (Optional) `file_order`: Enter a list of comma-separated file extensions to define a priority of files to transfer. Any file extensions that you don't list are transferred last in the order that the file system lists them. For example, `".mov, .txt"` prioritizes transferring `.mov` files over `.txt` files.
16. (Optional) `filter`: Filter files being transferred based on format. The `filter` field accepts valid regular expressions. For example, `"^.*\.(mov)$"` only uploads files ending in `.mov`
17. (Optional) `upload_hot_folder_enabled`: Turns on the upload hot folder functionality.
18. (Optional) `upload_hot_folder_local_source_folder`: This is required when `upload_hot_folder` is enabled. File Transfer recursively monitors all file system events in the specified folder and initiates uploads when files are added or modified.
19. (Optional) `upload_hot_folder_s3_destination_folder`: This is required when `upload_hot_folder` is enabled. Hot folder files that are uploaded are added to this folder.
20. `enable_metadata_filter`: When `true`, File Transfer filters system metadata files automatically. These files include `Thumbs.db`, `.DS_Store`, and files that start with `._`.
21. (Optional) `paths_local`: The root folder for File Transfer files in the local file system.
22. (Optional) `paths_remote`: The root folder for File Transfer files in the S3 bucket.
23. (Optional) `paths_local`: Enter a file path. Your transfers will begin at that local path in the file browser for the local file system.
24. (Optional) `paths_remote`: Enter a file path. Your transfers will begin at that path in the file browser for the S3 bucket.
25. Save the configuration file.
26. Run the following command to validate your AWS credentials: `filetransfer validate-credentials remote configuration`

- a. Replace *remote configuration* with the name of the configuration that you created.
- b. This command checks the AWS credentials that you provided, such as your IAM key. This command checks that File Transfer is able to connect to Amazon S3, and it lists objects in the bucket that's specified in the configuration file.

Having completed installation and configuration, you're now ready to use File Transfer.

# Transfer files using the File Transfer GUI

Learn how to browse and transfer files between your local computer and Amazon S3 with the File Transfer graphical user interface (GUI). You can also set up a *hot folder* to automatically upload new and updated files to an Amazon S3 bucket.

## Topics

- [File Transfer GUI overview](#)
- [Upload files](#)
- [Configuring hot folders](#)
- [Jobs, Logs, and Report tabs](#)
- [Download files](#)

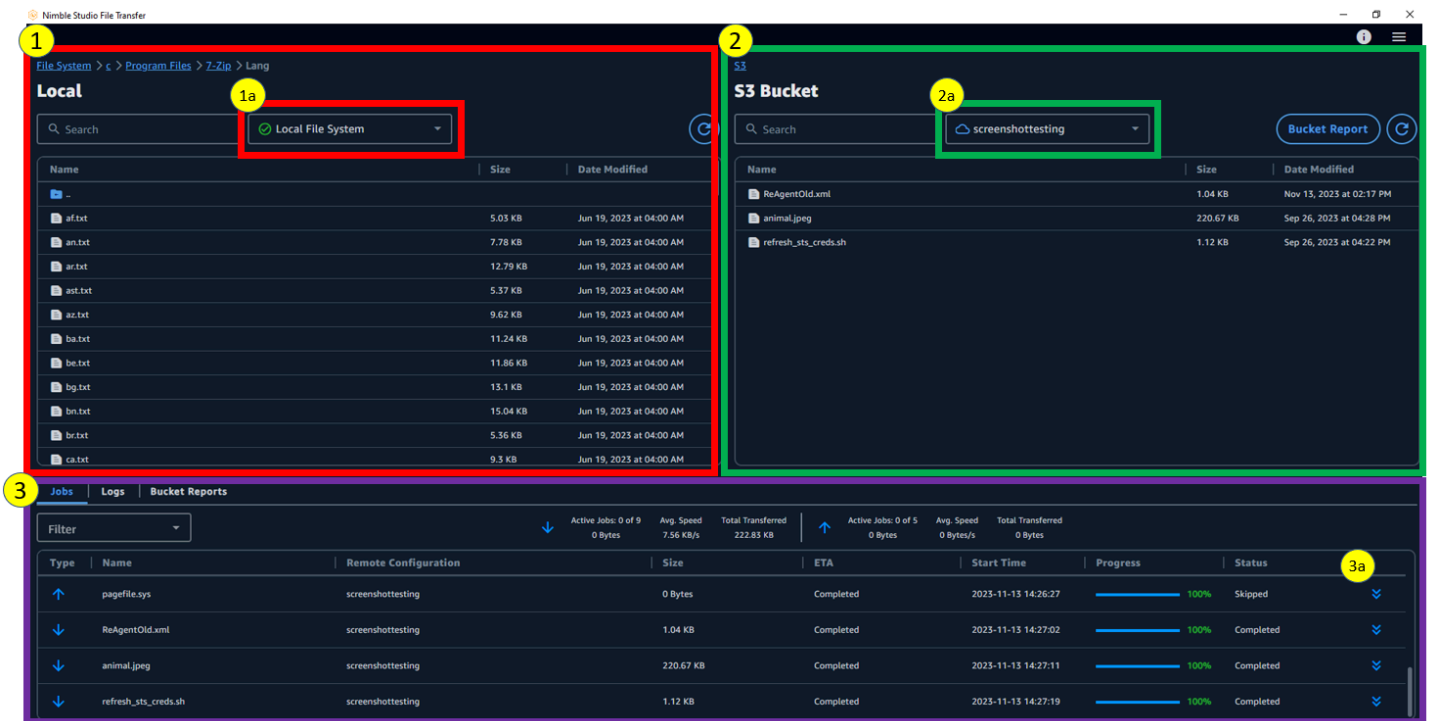
## File Transfer GUI overview

### Note

We recommend that you disable sleep mode on your local computer. If your computer activates sleep mode, ongoing transfers might be interrupted. In **Settings**, toggle **Disable Sleep (macOS only)**.

The following image outlines the various sections of the File Transfer GUI.





## 1. Local file browser

- a. File System dropdown

## 2. S3 Bucket file browser

- a. Remote Configuration dropdown

## 3. Jobs, Logs, and Bucket Reports tabs

- a. **Action** button in **Jobs** tab

# Upload files

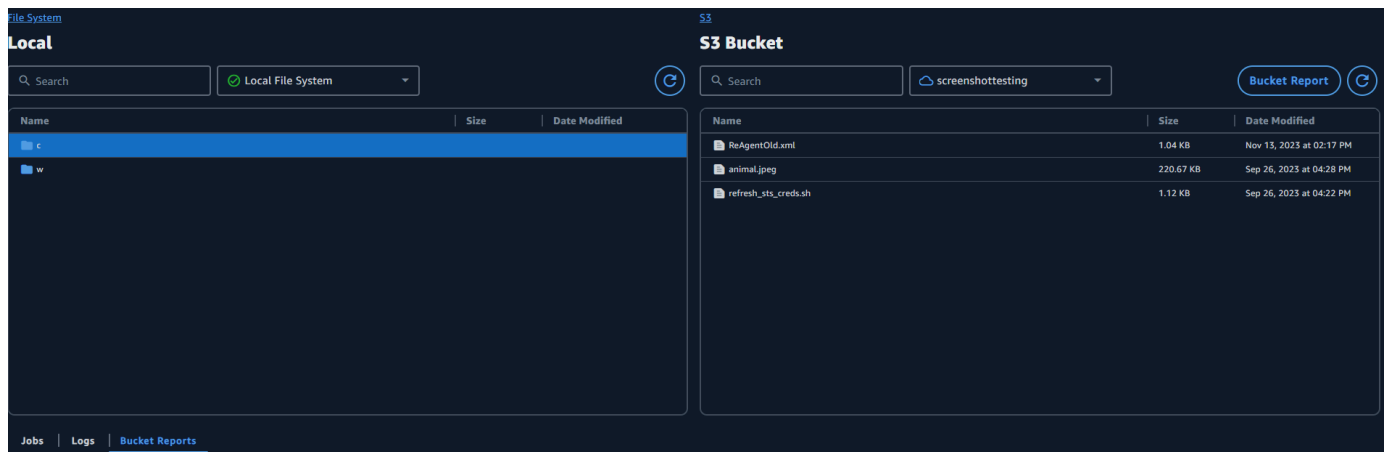
File Transfer uploads to all S3 storage classes. For more information about storage classes, see [Amazon S3 Storage Classes](#).

## To start uploads using the GUI

1. Open File Transfer.
  - a. Go to the **Start Menu** and search for **File Transfer**.
  - b. Select **Nimble Studio File Transfer** from the list.

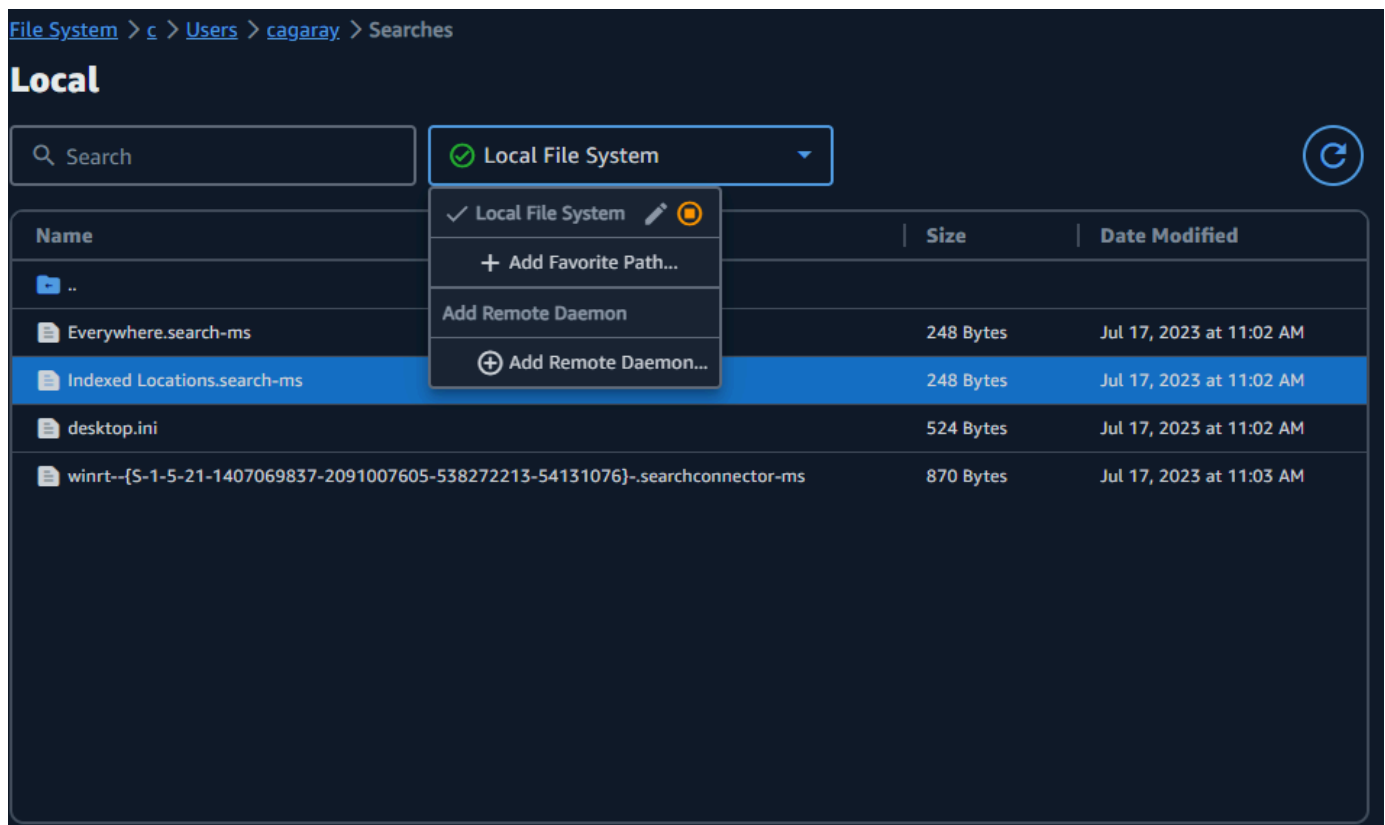
- On the landing page, the **Local** file browser displays on the left, and the **S3 Bucket** file browser displays on the right.

Example:



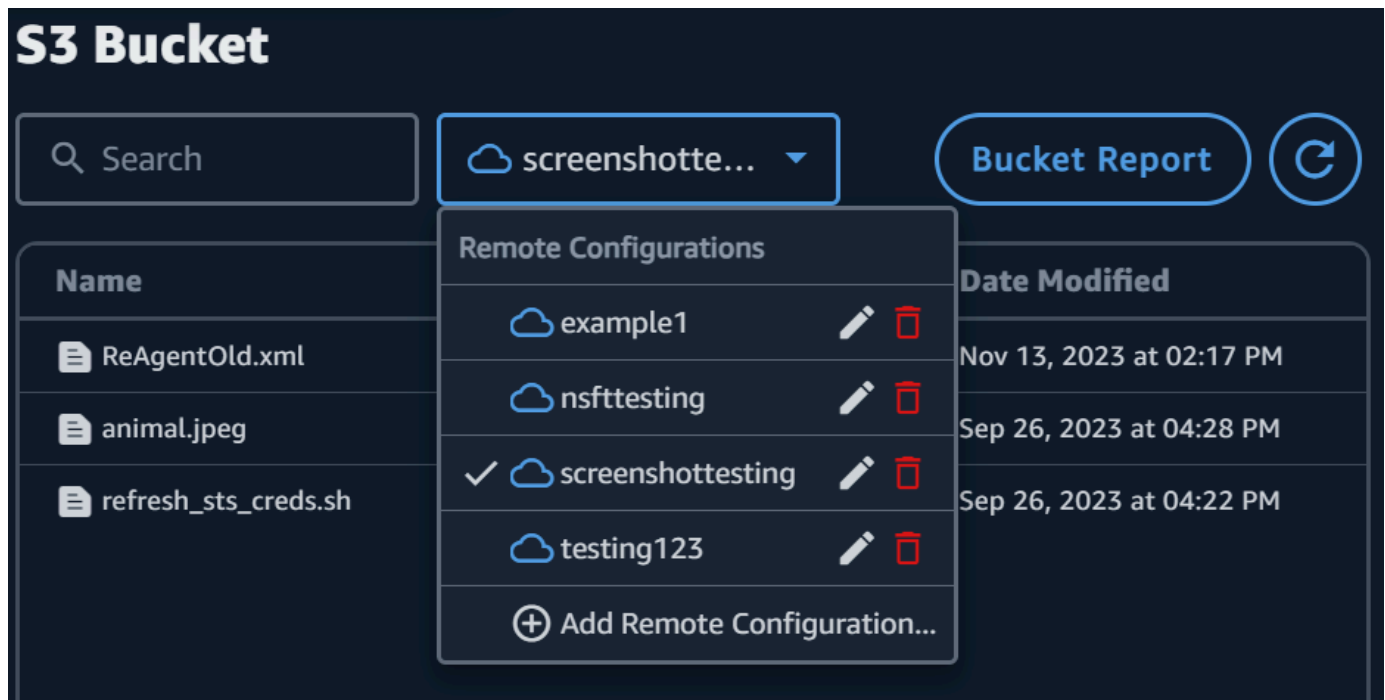
- In the Local file browser, choose the **File system** dropdown to select the Local file system, favorite path, or remote daemon from which you want to upload files. Alternatively, you can navigate through the directory to locate your desired path.

Example:



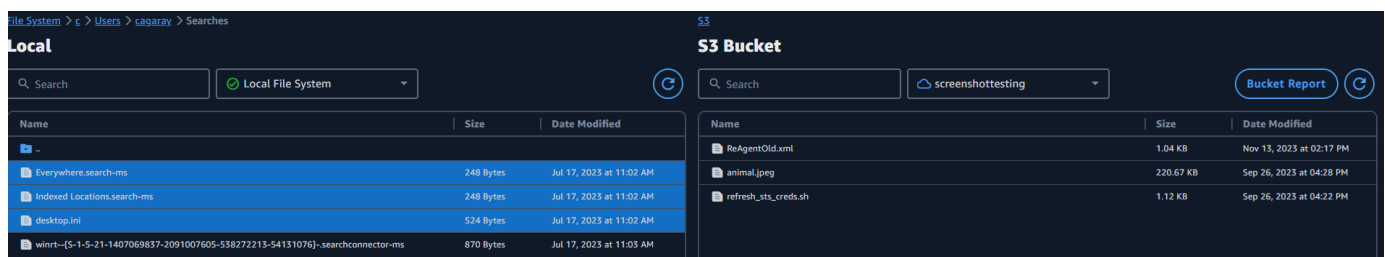
- In the S3 Bucket file browser, select the **Remote Configurations** dropdown. Select the Remote Configuration that contains the Amazon S3 bucket to which you want to upload files.

Example:



- Select the check box next to the files and folders that you want to transfer. The selected files and folders are highlighted.

Example:



- Drag and drop to move the selected files and folder from the left-side Local file browser to the right-side S3 Bucket file browser. You can drop the files and folders into the root, or into a specific folder in the S3 bucket.

## Configuring hot folders


With the upload hot folder, you can set File Transfer to continuously monitor a folder for changes. When you add a new file to the hot folder, File Transfer automatically uploads it to Amazon S3.

The upload hot folder recursively monitors all file system events within the directory that you specify in the GUI. Any new files that you add to this folder are automatically uploaded to the defined bucket(s).

The upload hot folder only works on file systems that support notifying clients of new file system events. The upload hot folder might not work on some remote mounted file systems, such as Network File System (NFS) and Server Message Block (SMB). Whether the upload hot folder works depends on the configuration of the file server.

Before you can use the upload hot folder, you must configure it.

### To configure the upload hot folder using the GUI

1. Open File Transfer.
  - a. Go to the **Start Menu** and search for **File Transfer**.
  - b. Select **Nimble Studio File Transfer** from the list.
2. Select the dropdown menu  and then choose **Settings**.
3. In the **Hot Folders** section, choose **Add Hot Folder**.
4. Give your new Hot Folder configuration a **Name**. It must be unique from other Hot Folder names.
5. Select the **Remote Configuration Name** from the dropdown that contains the S3 bucket and configuration you want to upload to.
6. Enter a **Local Source Folder** and an **S3 Destination Folder**. Alternatively, you can leave the S3 Destination Folder blank to upload to the root directory.
  - a. The **Local Source Folder** must contain the full path.  
  
Example directory: `/media/drive`
  - b. You can configure the same **Local Source Folder** to upload to multiple S3 buckets by selecting the (+) icon and choosing additional Remote Configurations.
7. Choose **Save** for your changes to take effect.

File Transfer will now recursively monitor all file system events in the Local source folder. It will also start uploads when files are added or modified in that folder.

## Jobs, Logs, and Report tabs

The **Jobs** tab at the bottom of the File Transfer landing page has the following functionality:

- View real-time progress of your transfer jobs by percentage and the progress bar, as well as remote configuration (job destination), job size, ETA, Start Time, and overall Status (displaying transfer speed if the job is in progress).
- **Pause, resume, cancel, rename your job, generate job reports** or view more **Job Details** about the job like the status of individual folder transfers by selecting the **Action** arrow button. (See screenshot below.)

Type	Name	Remote Configuration	Size	ETA	Start Time	Progress
↑	swapfile.sys	screenshottesting	0 Bytes	Completed	2023-11-13 14:16:02	100%
↑	DumpStack.log.tmp	screenshottesting	0 Bytes	Completed	2023-11-13 14:16:54	100%
↑	sub1.txt	screenshottesting	0 Bytes	Completed	2023-11-13 14:17:03	100%
↓	refresh_sts_creds.sh	screenshottesting	1.12 KB	Unknown	2023-11-13 14:22:52	0%

### Note

When you pause a transfer job, any files that were actively transferring will need to be restarted.

- Select the **Rename** feature to give a more descriptive name to your transfer job.
- Select the **Generate Report** feature to get a detailed report of all the files and folders that were part of your transfer job.
- The queue displays a list of jobs corresponding to the individual files that you selected to upload and download. Jobs at the beginning of the queue will be displayed at the top of the **Jobs** tab.

The **Logs** tab at the bottom of the File Transfer landing page has the following functionality:

- See more detailed information about your file transfers and your interactions with the File Transfer app.

Jobs   <u>Logs</u>   Bucket Reports		
Levels	Search	
Timestamp	Level	Message
2023-11-13 14:27:22	info	Completed job refresh_sts_creds.sh
2023-11-13 14:27:19	info	Completed download of refresh_sts_creds.sh
2023-11-13 14:27:19	info	Download progress for refresh_sts_creds.sh, transferred 0 Bytes of 1.1 KiB
2023-11-13 14:27:19	info	Started download for refresh_sts_creds.sh
2023-11-13 14:27:19	info	Job progress for refresh_sts_creds.sh: 0 Bytes/1.1 KiB transferred
2023-11-13 14:27:19	info	Job progress for refresh_sts_creds.sh: 0 Bytes/1.1 KiB transferred

The **Buckets Reports** tab at the bottom of the File Transfer landing page has the following functionality:

- Check the progress of the generation of an in-progress bucket report.
- View your generated bucket reports to get more detailed information about the files in your bucket.
- Using the **S3 Bucket** file browser, you can select **Bucket Report** to generate a bucket report. You can select the preferred remote configuration and output format in `.xlsx`, `.json`, or `.csv`. See **How to create a Bucket report by using the GUI** in the **Monitoring** page for more detail.

Jobs   <u>Logs</u>   Bucket Reports					
Search					
Remote Configuration	S3 Bucket	Status	Started	Completed	Report
screenshottesting	nsft-permission-set	Completed	2023-11-13 14:29	2023-11-13 14:29	C:\Users\cagaray\filetransfer\reports\bucket-report-20231113-142916.json

## Download files

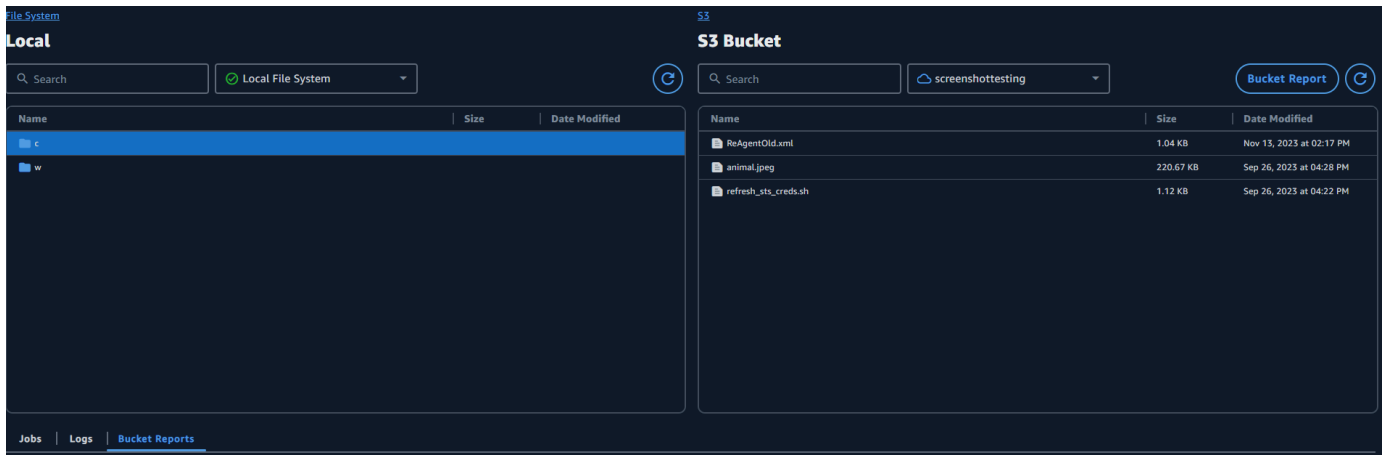
### Note

We recommend that you disable sleep mode on your local computer. If your computer activates sleep mode, ongoing transfers might be interrupted. In **Settings**, toggle **Disable Sleep (macOS only)**.

## To start downloads by using the GUI

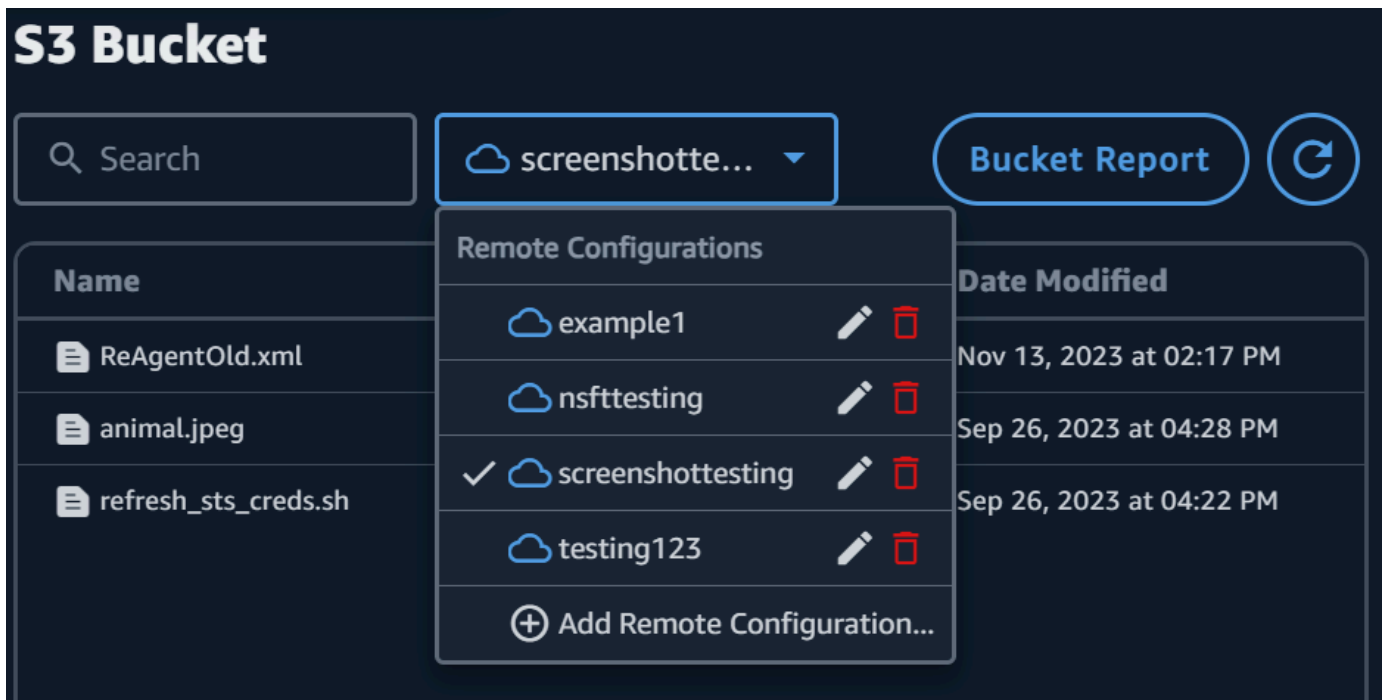
1. Open File Transfer.
  - a. Go to the **Start Menu** and search for **File Transfer**.
  - b. Select **Nimble Studio File Transfer** from the list.
2. On the landing page, the **Local** file browser displays on the left, and the **S3 Bucket** file browser displays on the right.

Example:



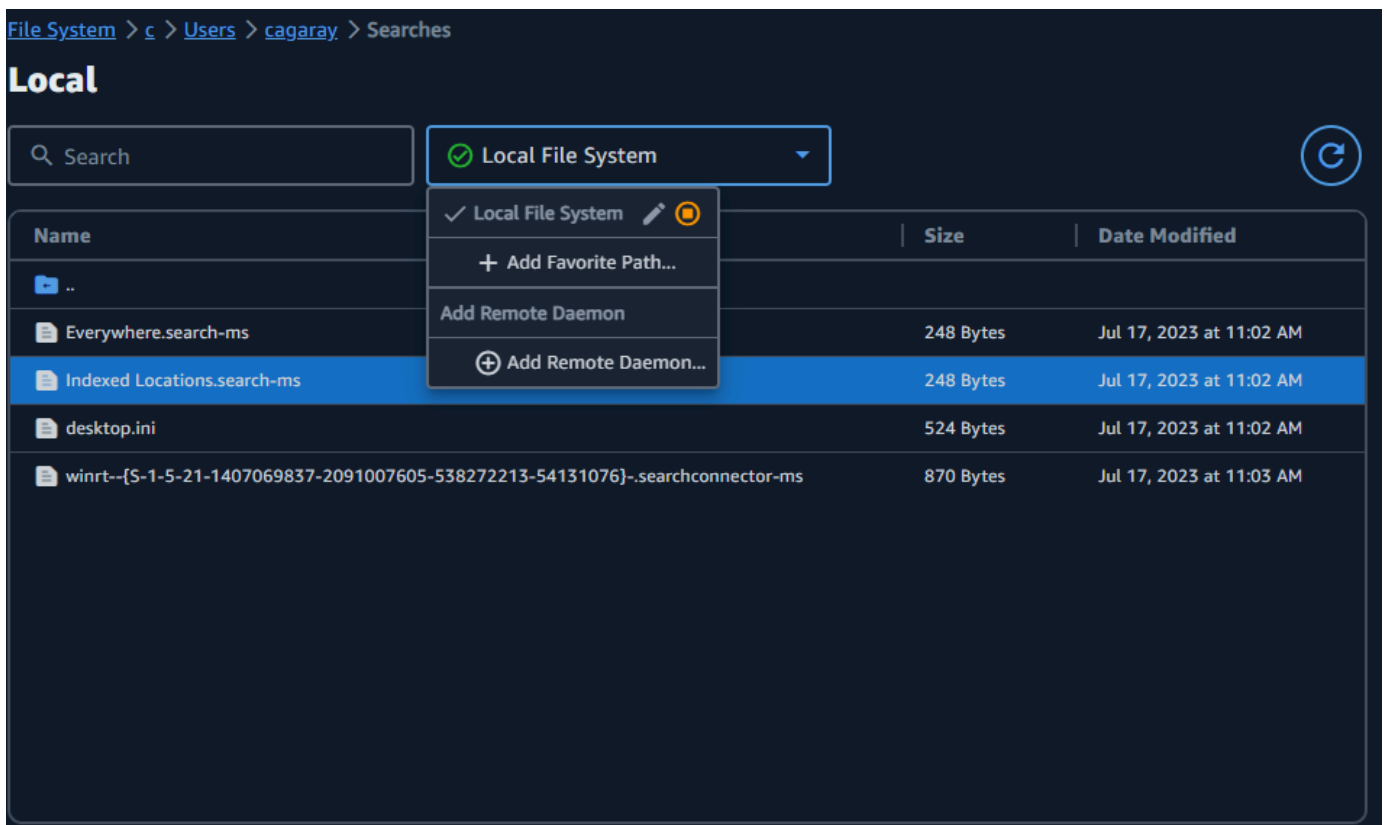
3. In the **S3 Bucket** file browser, select the **Remote Configuration** dropdown. Select the remote configuration that you want to download from.

Example:



4. In the **Local** file browser, select the **File system** dropdown to select the file system, favorite path, or remote daemon that you want to download to.

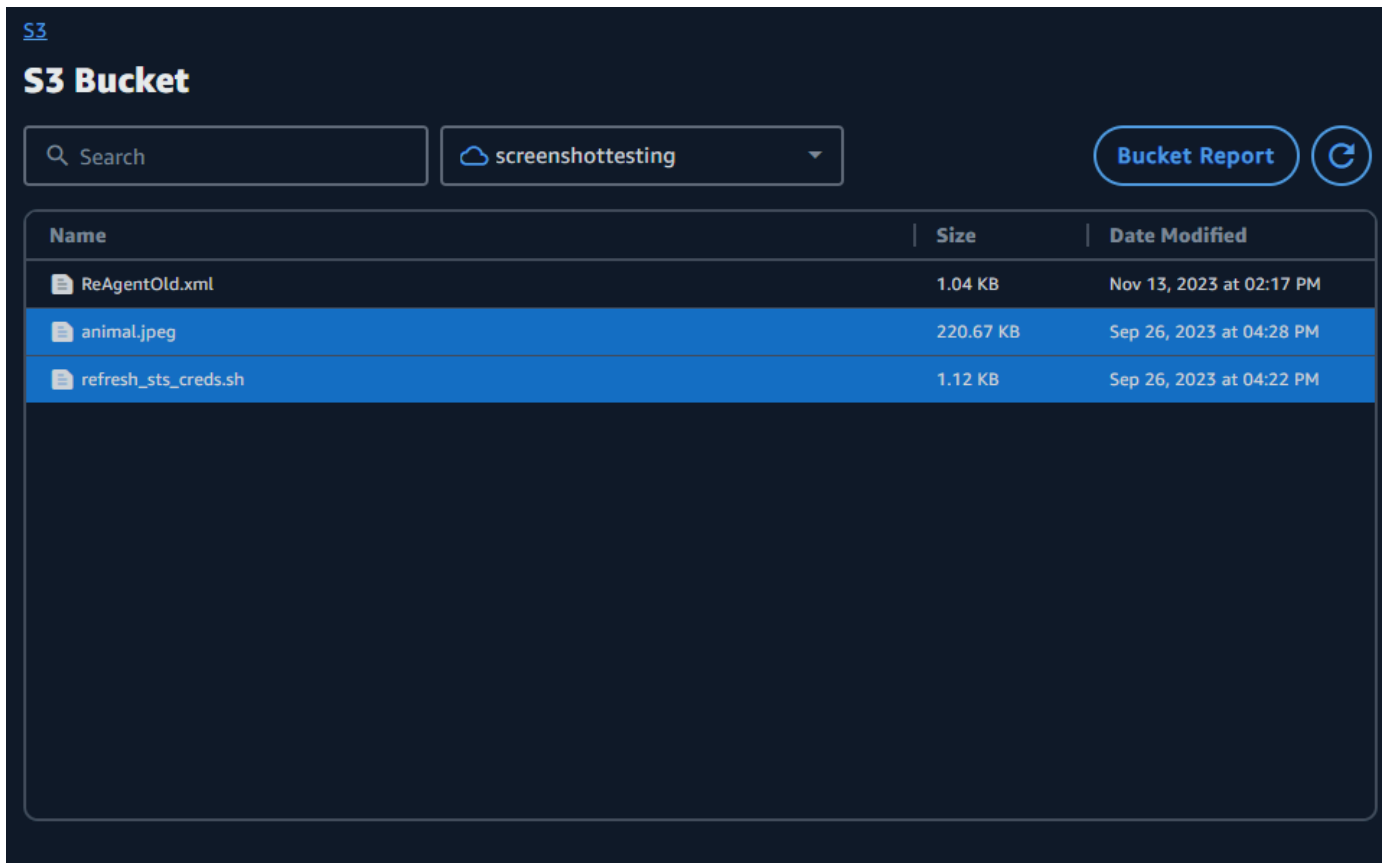
Example:





5. In the **S3 Bucket** file browser, select the file(s)/folder(s) that you want to download. The selected file(s)/folder(s) will appear highlighted.

Example:



6. Move the selected folder(s)/file(s) from the right-side **S3 Bucket** file browser to the left-side **Local** file browser. You can drop the file(s)/folder(s) into a specific folder in the Local file system or drop file(s)/folder(s) into the Local file system loosely.

# Transfer files using the command line interface (CLI)

The following sections detail how to upload, download, and configure hot folders for File Transfer with the command line interface (CLI).

## Topics

- [Upload files](#)
- [Configuring hot folders](#)
- [Download files](#)

## Upload files

The following commands are for the command line interface (CLI). With these commands, you can download files from the corresponding S3 bucket that you set up in your configuration file. You can run these commands from anywhere on your computer. However, we recommend running these commands from root.

File Transfer preserves the folder structure that you give it. This means that the folder structure remains the same after it reaches Amazon S3. However, you can't give File Transfer an absolute path. That means that the file path can't start with a / (forward slash).

File Transfer uploads to all S3 storage classes. For more information about storage classes, see [Amazon S3 Storage Classes](#).

### Note

We recommend that you disable sleep mode on your local computer. If your computer activates sleep mode, ongoing transfers might be interrupted. In **Settings**, toggle **Disable Sleep (macOS only)**.

## To start uploads by using the CLI

1. Open a terminal.
2. (Recommended) Navigate to the folder that contains the files and folders that you want to upload.

3. Run the following command from anywhere on your computer to upload files from your local machine to Amazon S3.

```
filetransfer upload [transfer profile] [relative path]
```

- a. Replace *[transfer profile]* with the transfer profile that you want to use.
- b. Replace *[relative path]* with the path of the file or directory that you want to upload.
  - i. A relative path doesn't start with a / (forward slash). Example: Users/username/Desktop/folder1
  - ii. Don't use an absolute path. An absolute file path starts with a / (forward slash). Example: /Users/username/Desktop/folder1
- c. Example command: `filetransfer upload [transfer profile name] /Users/username/Desktop/folder1`

File Transfer performs [File Transfer checksums](#) on your files. The checksum is used for additional file integrity verification, for you to detect if a file was unexpectedly modified. After those checksums are complete, the transfers start. If you provide your own *Media Hash List (MHL)*, File Transfer can verify that your files match the checksum defined in the MHL. The MHL is an XML file that often comes with raw camera footage.

## Flags

To perform certain actions, you can add flags to the end of a command. It's optional to use flags.

Flag	Description	Example
<code>--help</code>	List the available flags and commands.	<code>filetransfer --help</code>
<code>--checksum-algorithm</code>	Define which checksum computation File Transfer performs. Supported values: md5, XXHash, XXHash64, and XXH3.	<code>filetransfer upload <i>[remote configuration]</i> <i>[relative path]</i> --</code>

Flag	Description	Example
		checksum- algorithm xxhash64
--enable-metadata-filter	When used, File Transfer filters system metadata files automatically. These files include Thumbs.db , .DS_Store , and files that start with ._.	filetransfer upload <i>[remote configuration]</i> <i>[relative path]</i> --enable-metadata-filter
--filter	Filter files that are being transferred based on their format. The filter field accepts valid regular expressions. Example: <code>^.*\.(mov)\$</code> only transfers files ending in .mov. Filters can be added to the configuration file to perform certain actions. When you add filters to the configuration instead of using them as flags on specific commands, File Transfer applies the action to all transfers.	filetransfer upload <i>[remote configuration]</i> <i>[relative path]</i> --filter " <code>^.*\.(mov)\$</code> "  This command will only upload files that are in .mov format.

Flag	Description	Example
--force	<p>Force a transfer regardless of filters or conflicts. By default, File Transfer won't transfer previously transferred files.</p> <p>The force flag makes File Transfer ignore any filters or conflicts. This function is useful if you accidentally deleted a file which you need to download.</p>	<pre>filetransfer upload [remote configuration] [relative path] -- force</pre>
--max-age	<p>Only transfer files that were created or modified within a time window ending at the current time. If no units are supplied, the max age will be in seconds. For example, 2d will download files from the last two days, and 3500 will upload files from the last 3500 seconds.</p>	<pre>filetransfer upload [remote configuration] [relative path] --max-age "2d"</pre> <p>The command above will only upload files to the S3 bucket from the last 2 days.</p>
--retry-count	<p>If an error occurs, File Transfer will attempt to transfer your files as many times as the retry count.</p>	<pre>filetransfer upload [remote configuration] [relative path] -- retry-count 4</pre>

Flag	Description	Example
<code>--chunk-size</code>	Chunk size in MB (default 25)	<code>filetransfer upload [remote configuration] [relative path] -- chunk-size 50</code>
<code>--auto-tuning</code>	Allow tool to decide the best configuration values to optimize transfer	<code>filetransfer upload [remote configuration] [relative path] -- auto-tuning true</code>
<code>--max-active-checksums</code>	Max number of active checksums (default 1). This flag is only valid for uploads.	<code>filetransfer upload [remote configuration] [relative path] -- max-active- checksums 5</code>

Flag	Description	Example
<code>--max-active-transfers</code>	Max number of transfers	<code>filetransfer upload [remote configuration] [relative path] -- max-active- transfers 10</code>
<code>--prefix</code>	S3 prefix path for uploads and downloads	<code>filetransfer upload [remote configuration] [relative path] -- prefix my/ s3/path</code>
<code>--profile</code>	AWS named profile	<code>filetransfer upload [remote configuration] [relative path] -- profile my_named_ profile</code>

Flag	Description	Example
<code>--threads</code>	Number of threads per upload (default 10)	<code>filetransfer upload [remote configuration] [relative path] -- threads 10</code>
<code>FILETRANSFER_CONFIG_DIR</code>	<p>Overrides the default <code>.filetransfer</code> folder.</p> <p>This variable can define any directory to store the configuration file and the database file. If <code>FILETRANSFER_CONFIG_DIR</code> isn't set, the default value, <code>~/.filetransfer</code>, is used. The files must still be named <code>configuration.yaml</code> or <code>configuration.yaml</code>, and <code>checksum-cache.db</code>. If they don't exist, they are created.</p>	<code>FILETRANSFER_CONFIG_DIR=Desktop/ config1 filetransfer upload [remote configuration] [relative path]</code>

## Configuring hot folders

### To configure the upload hot folder by using the CLI.

1. Open the configuration file with any text editing software on your computer.
  - a. Windows: Navigate to the `User\<your username>` folder on your computer. Open the `.filetransfer` folder and open the `filetransfer.yaml` file with a text editor.
  - b. macOS: Enter **Cmd+Shift+G**. Then enter `~/.filetransfer`. Open the `filetransfer.yaml` file with a text editor.
  - c. Linux: Open the `filetransfer.yaml` file using any text editor. The file is located in `~/.filetransfer/configuration.yaml`.



## 2. Create a new `hot_folder` section with the following fields

```
hot_folders:
  - enabled: true
    local_source_folder: /Users/user/myhotfolder
    name: my_hot_folder
    remote_configurations:
      - remote_configuration_name: example_configuration
        s3_destination_folder: my/s3/prefix
```

3. The hot folder will only be active when `enabled` is set to `true`.
4. Replace *local\_source\_folder* with the folder location that you want File Transfer to monitor. The file path must contain the full path: `C:\path\to\upload-hot-folder` (Windows) or `/path/to/upload-hot-folder` (Linux & macOS).
5. Under `remote_configurations`:
  - a. Replace *remote\_configuration\_name* with the name of the remote configuration you want to use. You can find your remote configurations listed under `protocols.s3.transfer_profiles`.
  - b. (Optional) Replace *s3\_destination\_folder* with the S3 destination folder you want files to be uploaded to. File Transfer will create the folder in Amazon S3 if it does not already exist. Files will be placed in the root of the bucket if the value is not set.
  - c. (Optional) You can add as many remote configuration entries here as you would like, and the hot folder will start an upload using each remote configuration. This allows you to set a single hot folder that will upload to multiple buckets.
6. The following example is an upload hot folder configuration with multiple hot folders. In this example, there are two hot folders.
  - a. The first hot folder is set to watch the `Media/drive` folder. When files are added to this folder, it will start two uploads, one using the `example1` remote configuration, and the other using the `example2` remote configuration. Each upload will use its respective `s3_destination_folder`.
  - b. The second hot folder is set to watch the `/Users/user1/myhotfolder` folder. When files are added to this folder, it will start a single upload, using the `another_configuration` remote configuration. All files will be uploaded to the `example_folder` folder in Amazon S3.

```
hot folders:
  - enabled: true
    local_source_folder: /Media/drive
    name: my_hot_folder
    remote_configurations:
      - remote_configuration_name: example1
        s3_destination_folder: my/s3/folder
      - remote_configuration_name: example2
        s3_destination_folder: second/folder
  - enabled: true
    local_source_folder: /Users/user1/my_hot_folder
    name: another_hot_folder
    remote_configurations:
      - remote_configuration_name: another_configuration
        s3_destination_folder: example/folder
```

## 7. Save the configuration file.

### Note

When you start the daemon, or when hot folders are added/updated, an automatic forced upload will be started for the entire folder. If you do not prefer this behavior, you can open the GUI and cancel the job.

## Download files

With the following commands, you can download files from the corresponding S3 bucket that you set up in your configuration file. You can run these commands from anywhere on your computer. We recommend running these commands from root.

File Transfer can't download directly from the Deep Archive or Glacier storage classes. This is because they're stored in a different system. These types of objects require a different method of retrieving the files. For more information about retrieving objects from different systems, see [Restoring an archived object](#). For more information about storage classes, see [Amazon S3 Storage Classes](#).

**Note**

We recommend that you disable sleep mode on your local computer. If your computer activates sleep mode, ongoing transfers might be interrupted. In **Settings**, toggle **Disable Sleep (macOS only)**.

**To start downloads by using the CLI**

1. Open a terminal.
2. (Recommended) Navigate to the folder that you want to download your files to.
3. Run the following command from anywhere on your computer to download files from Amazon S3 onto your computer.

```
filetransfer download [transfer profile]  
                    [relative path]
```

- a. Replace *[transfer profile]* with the transfer profile that you want to use.
- b. Replace *[relative path]* with the path that you want to download your files to.
  - i. A relative path doesn't start with a / (forward slash). Example: Users/username/Desktop/folder1
  - ii. Don't use an absolute path. An absolute file path starts with a / (forward slash). Example: /Users/username/Desktop/folder1
- c. Example command: `filetransfer download [transfer profile name] /Users/username/Desktop/folder1`

File Transfer checks the local File Transfer database to verify if the file that you selected has already been downloaded.

**Flags**

Flags can be added to the end of a command in order to perform certain actions. It's optional to use flags.

Flag	Description	Example
<code>--help</code>	List the available flags and commands.	<code>filetransfer --help</code>
<code>--checksum-algorithm</code>	Define which checksum computation File Transfer performs. Supported values: md5, XXHash, XXHash64, and XXH3.	<code>filetransfer download [remote configuration] [relative path] --checksum-algorithm xxhash64</code>
<code>--enable-metadata-filter</code>	When used, File Transfer filters system metadata files automatically. These files include <code>Thumbs.db</code> , <code>.DS_Store</code> , and files that start with <code>._</code> .	<code>filetransfer download [remote configuration] [relative path] --enable-metadata-filter</code>
<code>--filter</code>	Filter files that are being transferred based on their format. The filter field accepts valid regular expressions. Example: <code>^.*\.(mov)\$</code> only transfers files ending in <code>.mov</code> . Filters can be added to the configuration file to perform certain actions. When you add filters to the configuration instead of using them as flags on specific commands, File Transfer applies the action to all transfers.	<code>filetransfer download [remote configuration] [relative path] --</code>

Flag	Description	Example
		<pre>filter "^.* \.(mov)\$"</pre> <p>This command will only download files that are in .mov format.</p>
--force	<p>Force a transfer regardless of filters or conflicts. By default, File Transfer won't transfer previously transferred files.</p> <p>The force flag makes File Transfer ignore any filters or conflicts. This function is useful if you accidentally deleted a file which you need to download.</p>	<pre>filetransfer download [remote configuration] [relative path] -- force</pre>
--max-age	<p>Only transfer files that were created or modified within a time window ending at the current time. If no units are supplied, the max age will be in seconds. For example, 2d will download files from the last two days, and 3500 will upload files from the last 3500 seconds.</p>	<pre>filetransfer download [remote configuration] [relative path] --max- age "2d"</pre> <p>The command above will only download files to the S3 bucket from the last 2 days.</p>

Flag	Description	Example
--retry-count	If an error occurs, File Transfer will attempt to transfer your files as many times as the retry count.	filetransfer download <i>[remote configuration relative path]</i> -- retry-count 4
--chunk-size	Chunk size in MB (default 25)	filetransfer download <i>[remote configuration relative path]</i> -- chunk-size 50
--auto-tuning	Allow tool to decide the best configuration values to optimize transfer	filetransfer download <i>[remote configuration relative path]</i> -- auto-tuning true

Flag	Description	Example
<code>--max-active-checksums</code>	Max number of active checksums (default 1)	<code>filetransfer download [remote configuration] [relative path] --max-active-checksums 5</code>
<code>--max-active-transfers</code>	Max number of transfers	<code>filetransfer download [remote configuration] [relative path] --max-active-transfers 10</code>
<code>--prefix</code>	S3 prefix path for uploads and downloads	<code>filetransfer download [remote configuration] [relative path] --prefix my/s3/path</code>

Flag	Description	Example
--profile	AWS named profile	filetransfer download <i>[remote configuration] [relative path]</i> -- profile my_named_ profile
--threads	Number of threads per download (default 10)	filetransfer download <i>[remote configuration] [relative path]</i> -- threads 10



Flag	Description	Example
FILETRANSFER_CONFIG_DIR	<p>Overrides the default <code>.filetransfer</code> folder.</p> <p>This variable can define any directory to store the configuration file and the database file. If <code>FILETRANSFER_CONFIG_DIR</code> isn't set, the default value, <code>~/.filetransfer</code>, is used. The files must still be named <code>configuration.yaml</code> or <code>configuration.yaml</code>, and <code>checksum-cache.db</code>. If they don't exist, they are created.</p>	<pre>FILETRANSFER_CONFIG_DIR=Desktop/config1filetransferdownload <i>[remote configuration]</i> <i>[relative path]</i></pre>

# File Transfer checksums

File Transfer performs checksums in the background for your uploads to verify the integrity of the files on disk against the files in the S3 bucket. Checksums are calculated for each file you upload, and the checksum values are stored in the File Transfer database.

The following explains File Transfer's native checksum process:

1. Checksums are calculated for files that you upload.
2. If the upload file doesn't exist in the S3 bucket, then the checksum is added to the File Transfer database, and the file is uploaded to the Amazon S3 bucket.
3. If the upload file already exists in the S3 bucket, then the upload file's checksum is checked against the checksum in the File Transfer database.
  - a. If the checksums match, then the file is not uploaded because it is identical to the file in the S3 bucket.
  - b. If the checksums don't match, the upload file has been modified and it is uploaded to the S3 bucket. The new checksum is added to the File Transfer database.

If you want to skip the native checksum process within File Transfer, add a [Media Hash List \(MHL\)](#) to the same folder, or any parent folder, of the file that you want to upload. If you provide your own MHLs, File Transfer verifies file hashes against the MHL. A single MHL in the root of your local File Transfer folder can recursively reference files within sub-folders. We recommend that you have a single MHL file, that has checksums for most, if not all, of the files in the folder, rather than an MHL file for every file.

The following are some important concepts to understand about File Transfer checksums.

## Native checksums

Checksums are calculated for files that you upload. The checksums are checked against the checksums in the File Transfer database. If there is a mismatch in checksums, File Transfer uploads the file again. A mismatch in checksums occurs if you have changed the file since the original upload. The first time the file is uploaded, there will be no existing file in Amazon S3 that File Transfer can use to compare against. The CPU count impacts checksum performance.

## MHL checksums

If you want to skip the native checksum process of File Transfer, supply a Media Hash List (MHL) file in the upload directory. The MHL file is used to verify the integrity of the files as they move to different places.

File Transfer treats the MHL as the authoritative source and appends the checksum value to the uploaded object's metadata. The MHL file must contain one of the following fields: `<md5>HEXVALUE</md5>`, `<xxhash64>HEXVALUE</xxhash64>`, or `<xxhash64be>HEXVALUE</xxhash64be>`. To learn more about MHL specification, see [About Media Hash List](#).

## Configurable checksums

By default, File Transfer uses one less than the total logical core count to concurrently compute checksums. This value is the maximum threshold.

For example, if your host machine has 12 logical cores, then the maximum threshold is 11. The minimum threshold will always be 1, regardless of the number of cores in the machine. By default, 1 checksum runs at a time. There is a safeguard in place to ensure that the number of max active checksums doesn't surpass your maximum threshold.

You can adjust the number of checksums running at the same time by modifying the `max_active_checksums` configuration property. An example of when you might want to adjust the number of checksums is if you wanted to reduce the amount of resources that File Transfer uses. This frees CPU resources for other processes.

# Using the remote daemon

The remote daemon starts a daemon that a GUI running on a different machine can connect to. (A different machine might be one with more bandwidth or specific access to certain file systems.) You can shift your resource load to another computer so that you're not impacting your machine's performance. This is also useful if you have multiple people working on the same File Transfer application.

## Topics

- [Configure the remote daemon](#)
- [Run a remote daemon](#)

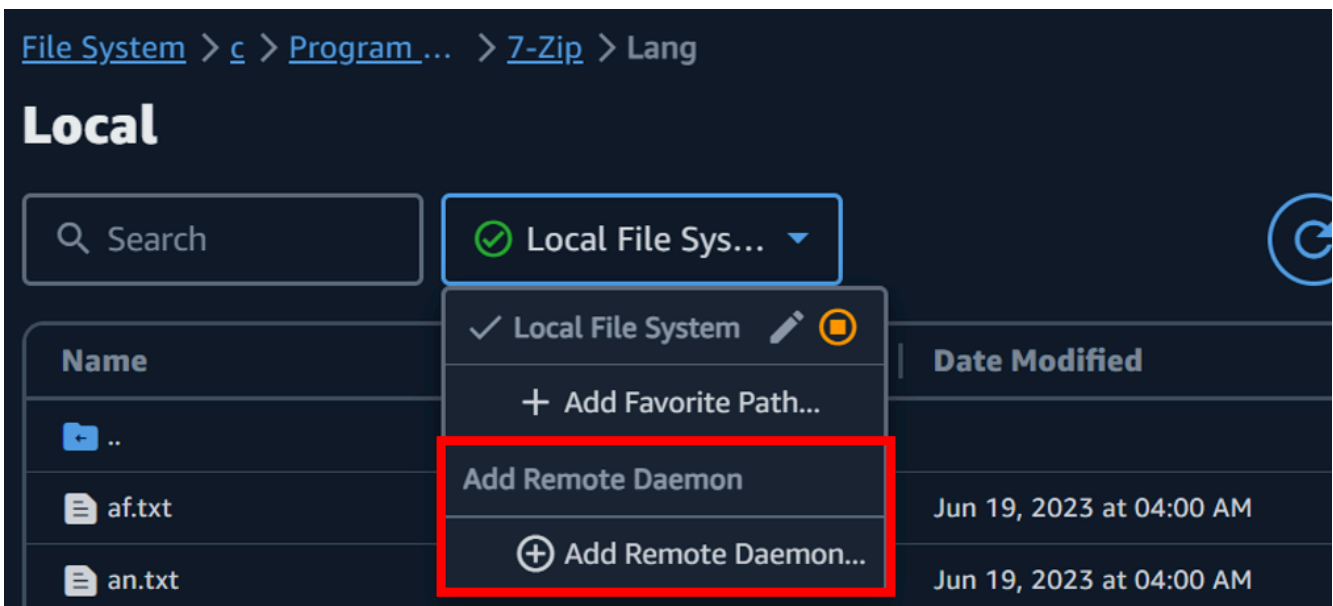
## Configure the remote daemon

Before you can use the remote daemon, you must configure it. You can configure it through the GUI or by using the command line interface (CLI) to modify the configuration file.

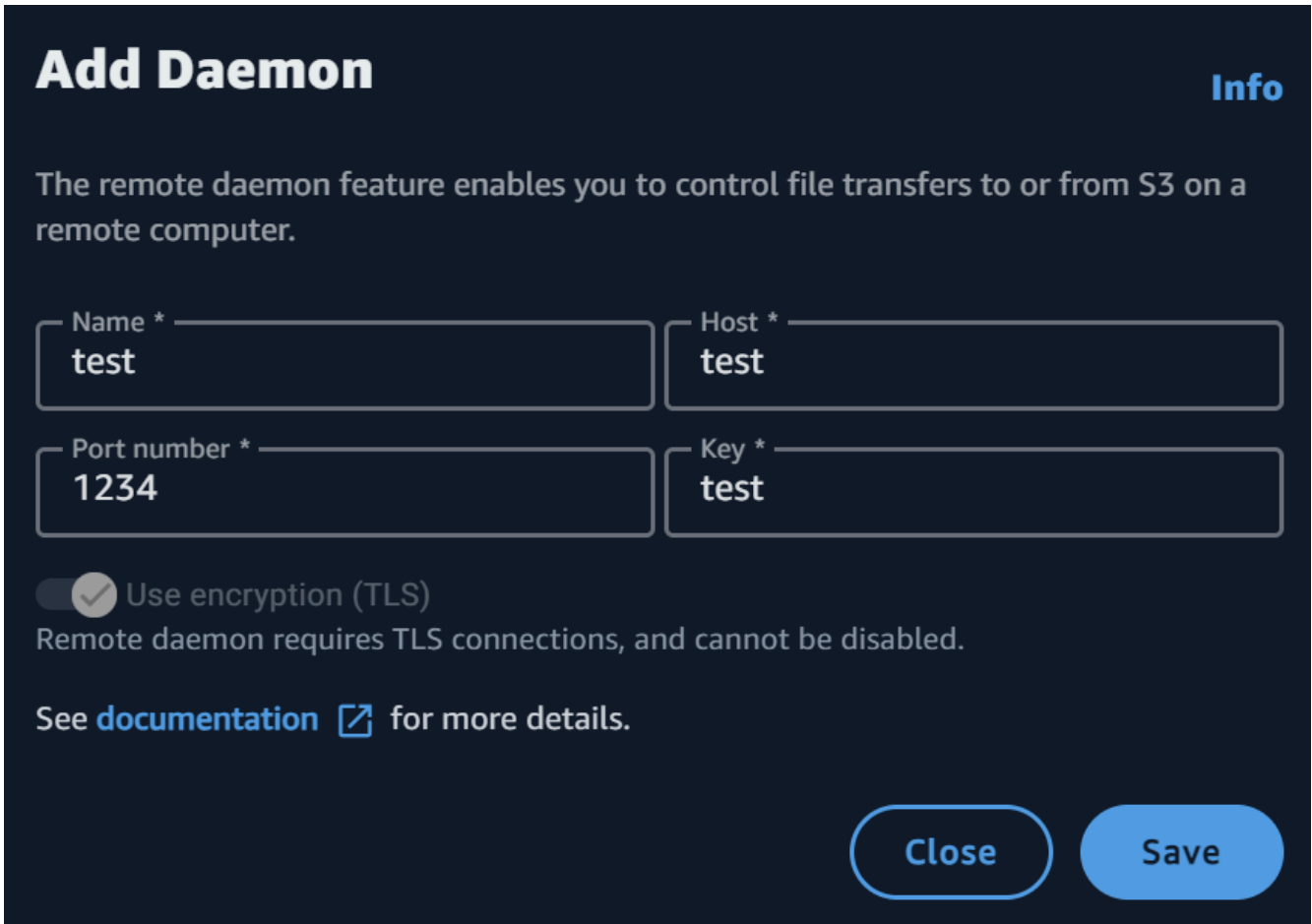
### GUI

#### To set up the remote daemon

1. In the Local file browser, select the **File System** dropdown, and then choose **Add Remote Daemon**.



2. In the popup screen, add **Name**, **Host**, and **Port Number**, and choose **Save**.
  - Set up a valid Transport Layer Security (TLS) certificate. TLS is required for remote daemon setups. To turn on TLS, set up a TLS certificate and go through the trust process. Contact your IT admin for guidance about how to set up a TLS certificate.



**Add Daemon** Info

The remote daemon feature enables you to control file transfers to or from S3 on a remote computer.

Name \*  Host \*

Port number \*  Key \*

Use encryption (TLS)  
Remote daemon requires TLS connections, and cannot be disabled.

See [documentation](#) [↗](#) for more details.

Close Save

3. The remote daemon will now display in the **File System** dropdowns, ready for uploads and downloads.

## CLI

### To set up the remote daemon using CLI

1. Open File Transfer.
  - a. Go to the **Start Menu** and search for **File Transfer**.
  - b. Select **Nimble Studio File Transfer** from the list.

2. Open the configuration file with any text editing software on your computer.
  - a. Windows: Navigate to the `User` folder on your computer. Open the `.filetransfer` folder, and then open the `configuration.yaml` file with a text editor.
  - b. macOS: Enter `Cmd+Shift+G`. Then enter `~/filetransfer`. Open the `configuration.yaml` file with a text editor.
  - c. Linux: Open the `configuration.yaml` file using any text editor. The file is located in `~/filetransfer/configuration.yaml`.
3. Set up a valid Transport Layer Security (TLS) certificate. TLS is required for remote daemons.
  - a. To turn on TLS, set up a TLS certificate and go through the trust process. Contact your IT admin for guidance about how to set up a TLS certificate.
  - b. After you receive TLS certificates, add them to your configuration file by modifying the following settings:
    - i. `api_server.tls_enabled`: This indicates if the remote daemon should try to run the daemon using HTTPS. To run a remote daemon, this must be set to `true`.
    - ii. `api_server.tls_certificate_file`: The full path to your certificate file's location.
    - iii. `api_server.tls_key_file`: The full path to your key file's location.
4. (Optional) Adjust your firewall settings. Firewall settings vary based on your network, system, and other factors, so there isn't a specific guide to create and adjust your rules. However, there are some important things to be aware of when you create a rule:
  - a. You can change what addresses are being used and what port(s) are being listened on. Your firewall rule must be adjusted based on the following:
    - The addresses and ports that are used.
    - If it's a Transmission Control Protocol (TCP) connection.
  - b. By default, the remote daemon listens to all addresses of the machine running the remote daemon on port 50006 over TCP.
5. Modify the contents of the configuration file so that the following variables are defined:

- a. (Optional) `api_server.allowed_origins`: Enter any cross-origin resource sharing (CORS) allowed origin headers here. These validate the source of the Google Remote Procedure Call (GRPC) request.
- b. `api_server.allow_ui_configuration`: This determines if the GUI can make changes to the remote daemon's configuration file. If set to `false`, GUI users can't change the configuration.
- c. `api_server.remote.enabled`: Determines if File Transfer starts a remote daemon by default. If set to `true`, running `filetransfer` daemon will start a remote daemon.
- d. (Optional) `api_server.remote.ports`: A comma-separated list of ports for the remote daemon to monitor. If undefined, File Transfer will use the default port of 50006.
- e. (Optional) `api_server.remote.address`: The address for the remote daemon to use. Can be an IP or host name. If undefined, File Transfer will listen on all available interfaces on the machine.
- f. `api_server.remote.key`: The key used by GUI users to connect to the remote daemon. We recommend that you set up a strong key that follows the National Institute of Standards and Technology (NIST) password guidelines in the [NIST Special Publication 800-63B](#). If you already have an alternative pre-shared key (PSK) distribution mechanism, we recommend that you use 128-bit keys. We recommend that you use keys that are generated by a cryptographically strong, random bit generator.
- g. (Optional) `api_server.blocked_paths`: Block paths from being viewed or uploaded to Amazon S3 by the GUI. This is especially useful when the remote daemon user might not want to allow the GUI user to have full access to their file system.
  - i. You can enter an absolute path or just a folder name. Relative paths with multiple folders aren't allowed. The following two path examples lead to the folder name: Desktop.
    - A. Absolute path example: `/Users/User1/Desktop`
    - B. Relative path example (not allowed): `User1/Desktop`
  - ii. File Transfer will block any paths that contain the folder name or absolute path.  
Default blocked paths:

- A. All operating systems: ``.aws`` and `.filetransfer`
  - B. Windows: `%SYSTEMROOT%`
  - C. Linux & macOS:  ``/etc` , /dev`
- iii. Absolute paths are treated as case-insensitive, and explicit folder names are case-sensitive. For example, `/Users/User1/Desktop` will also block `/users/user1/desktop`. However, `Desktop` won't block the `desktop` folder.
  - iv. If an absolute path is a symbolic link, File Transfer will also block the path pointed to by the symbolic link. However, if the user blocks a specific folder that is a symbolic link, File Transfer won't detect the symbolic link. For example, `/Users/User1/Desktop` becomes `/Users/Downloads` and both paths are blocked. `Desktop` becomes `/Users/Downloads`, and only paths that contain `Desktop` are blocked.
6. Save the configuration file.

The following example is a portion of the configuration file that sets up the remote daemon.

```
api_server:
  allowed_origins: ""
  allow_ui_configuration: true
  blocked_paths:
    - .aws
    - .filetransfer
    - /dev
    - /etc
  enabled: true
  remote:
    enabled: true
    key: example_key
    ports: 50007, 50008
    address: 10.0.0.68
  tls:
    enabled: true
    certificate_file: /your/path/to/cert/server.crt
    tls_key_file: /your/path/to/cert/server.\key
```



# Run a remote daemon

After you set up a remote daemon, you can use it to run transfers.

## To run a remote daemon from the CLI

1. To start a remote daemon, run the following command from the command line interface (CLI) of the host machine: `filetransfer daemon --remote --address=address --ports=ports`
  - a. Replace *address* with the file location that you want to transfer.
  - b. Replace *ports* with the port that you want to transfer to.
  - c. The `--address` and `--ports` flags only work when used in conjunction with the `--remote` flag. If these flags are used, they override the ports and the address values in the configuration file. If an address or ports flag isn't provided, and if there isn't a value in the configuration file, then the remote daemon defaults to using all local IPv4 address and port 50006
2. Successfully starting the remote daemon results in the following status message: File Transfer daemon is listening on *host-address* on port(s) *port-numbers*

## To run a remote daemon from the GUI

### Note

Your host machine running the remote daemon must first start a remote daemon using the CLI (see the CLI instructions above).

1. In the Local file browser, select the **File System** dropdown, and then select the desired **remote daemon**.
2. Begin uploading or downloading.
3. You can add favorites on your remote daemon just like your Local Filesystem.

# File Transfer best practices

To maximize benefits from Nimble Studio File Transfer, we recommend that you perform the best practices on this page.

## Contents

- [Amazon Simple Storage Service \(Amazon S3\)](#)
- [AWS Key Management Service \(AWS KMS\)](#)
- [Hardware](#)
- [Configuration](#)
- [Performance optimization](#)

## Amazon Simple Storage Service (Amazon S3)

- Follow the Amazon Simple Storage Service (Amazon S3) bucket naming practices that are described in the [Creating object key names](#) tutorial.
- To optimize transfer speeds from across the world into Amazon S3 buckets, follow the instructions in the [Configuring fast, secure file transfers using Amazon S3 Transfer Acceleration](#) tutorial.
- To minimize storage costs, configure a lifecycle rule by following the instructions in the [Configuring a bucket lifecycle configuration to abort incomplete multipart uploads](#) tutorial. For more information about decreasing costs, see the blog post, [Discovering and Deleting Incomplete Multipart Uploads to Lower Amazon S3 Costs](#).

## AWS Key Management Service (AWS KMS)

- When you create an S3 bucket in [Create an S3 bucket](#), we recommend that you choose an **AWS Key Management Service key (SSE-KMS)**. For more information about KMS keys, see [Customer keys and AWS keys](#).

## Hardware

We recommend that your computer meets the following requirements for you to use File Transfer.

- 8 logical CPU cores
- 8 GB RAM

File Transfer can run on a machine with fewer specifications than these, but that can decrease performance.

## Configuration

You have the option to use autotuning or manually tune parameters to best meet your file transfer use cases. If you choose to manually tune your settings, poor performance may be encountered if done improperly. We recommend most users keep **autotuning** enabled.

### Autotuning configuration

We recommended that most users leave the **Transfer autotuning** setting enabled. If the **Transfer autotuning** setting is enabled, File Transfer automatically sets values for **Number of threads** and **Chunk size**. File Transfer determines the most effective settings based on the type of data that you transfer.

The **Transfer autotuning** setting adjusts your performance settings on a per-file basis. If you manually set the values for **Number of threads** and **Chunk size**, those settings are applied to the entire batch of transfers. This is why **Transfer autotuning** usually has increased performance when you transfer different sized files. The performance of the **Transfer autotuning** setting is comparable to manual settings adjustments when you transfer files of similar sizes. If you have advanced knowledge of your hardware and transfer data, your manual settings can outperform the values that **Transfer autotuning** chooses.

### Threads

With File Transfer, each transfer is split into multiple, individual threads that are used to transfer each file. Threads are most effective when you upload large files (> 1 GB). Threads can also help with small files, but the differences in transfer speeds won't be noticeable.

By default, the number of threads is 10. We recommend that you raise this value by increments of 5 until you fully use your bandwidth resources. You can monitor the bandwidth resources from the graphical user interface (GUI) by looking at the download and upload speeds for transfers.

## Chunk size

The chunk size is the size (in MB) that is delivered by each thread. Chunk size is helpful if a file size is repeated within the set of files being uploaded.

We recommend that you set the chunk size to 5 to 10 times greater than the average file size that you are transferring. For example, if the average file size in a dataset is 50 MB, set the chunk size between 55 and 60. If the file size is larger than 1 GB, this won't show as much benefit.

## Max active transfers

Max active transfers determine how many individual files that File Transfer processes at the same time. Adjusting max active transfers is most effective when you're transferring multiple small files that are less than 1 GB. We recommend that you increase the value of max active transfers as the file size decreases.

The following table shows the recommended starting points and increments for max active transfers. Start at the value in the max active transfers column and raise it by the increment amount until you reach the desired performance.

File size	Max active transfers	Increments
< 1 MB	100	20
> 1 MB–< 100 MB	50	10
> 100 MB–< 1 GB	25	5
> 1 GB	10	2

## Checksums

[File Transfer checksums](#) are the number of individual checksums that File Transfer processes at a time. The checksum algorithm is the algorithm that File Transfer uses for file integrity when transferring files.

You can choose between four checksum algorithms: MD5, XXHash, XXHash64, and XXH3. This is a preference based on what level of security and speed that you want. The earliest and most standard checksum method that is secure is MD5.

The recommended max active checksums value is the total number of CPU cores minus 1.

## Performance optimization

This topic explains the causes for slow upload speeds, and provides some changes that you can make to increase speed.

### Topics

- [Network bandwidth](#)
- [Disk throughput](#)
- [Latency](#)
- [Throttling](#)
- [Maximum limit of open files](#)
- [Bucket visibility](#)
- [Optimize uploads \(when not autotuning\)](#)
- [Configuration and database file location](#)
- [Turning off the API server](#)

## Network bandwidth

File Transfer increases network use and saturation. It can't deliver faster than the bandwidth that it's allocated to use. If your machine has been allocated a network bandwidth of 500 Mbps, the fastest File Transfer can try to deliver at is 500 Mbps. If you want faster transfers, allocate additional bandwidth to the host system.

## Disk throughput

Disk throughput must scale accordingly with increasing [Network bandwidth](#). You need enough I/O throughput to support a high max active transfer or thread count configuration. Your transfers will slow down if the storage that is attached to the host machine (such as NAS, SAN, local SSD, and

external HDD) doesn't have enough I/O throughput. To avoid this, upgrade your infrastructure by upgrading your hardware, CPU, and internet.

## Latency

We recommend that you deploy File Transfer infrastructure in the AWS Region that is geographically closest to the download and upload facility. The latency between the transfer profile's internet service provider to the destination will vary, unless the transfer profile is using AWS Direct Connect. For more information about AWS Direct Connect, see the [AWS Direct Connect User Guide](#).

## Throttling

File Transfer can't throttle its bandwidth use. To work around this issue, use Quality of Service (QoS) to limit the firewall layer, or traffic shape at the virtual local area network (VLAN) layer.

## Maximum limit of open files

Some host machines (mostly Linux and macOS) come with preconfigured soft and hard limits for the maximum number of open files. At minimum, File Transfer creates file descriptors to access disk and network resources. We recommend that your host machine has a 20,000 limit for maximum open files.

## Bucket visibility

File Transfer uses the default Amazon S3 endpoints. You can choose to use the default Amazon S3 accelerated endpoints. For more information about accelerated endpoints, see [Amazon Simple Storage Service \(Amazon S3\)](#).

You can use the AWS Command Line Interface (AWS CLI) to list content in your bucket. Do this by using your File Transfer access and secret keys, or through the [Amazon S3](#) console.

## Optimize uploads (when not autotuning)

For all-around best performance, keep the chunk size between 25–100MB. Threads and max active transfers will vary depending on the characteristics of your upload package. Single session transfer speeds are limited by the protocols in use (TCP/HTTP). The optimal configuration includes lower chunk sizes, and it includes higher thread and max active transfers settings. It is a best practice to

set the chunk size to be slightly bigger than the median file size. However, the best practice is to not exceed 50 MB on most hardware.

## Configuration and database file location

The configuration file and the database file are located in any directory, as defined by the `FILETRANSFER_CONFIG_DIR` environment variable. If the variable isn't set, these files are located in `~/filetransfer` by default. The configuration file is named `configuration.yaml` and the database file is named `checksum-cache.db`.

## Turning off the API server

By default, File Transfer listens on port 50005 for incoming connections from the File Transfer graphical user interface (GUI) application. To turn this off, define `api_server.enabled` in the configuration file and set it to `false`.

# Monitoring Nimble Studio File Transfer

Monitoring is an important part of maintaining the reliability, availability, and performance of Nimble Studio File Transfer and your AWS solutions. We recommend collecting monitoring data from all of the parts of your AWS solution. This helps you debug a multipoint failure if one occurs.

File Transfer uploads files to Amazon Simple Storage Service (Amazon S3) by using Amazon S3 APIs. Therefore, all CloudTrail, CloudWatch, and AWS CloudFormation information about File Transfer is logged as S3 usage. To learn more about how to monitor your S3 usage, see the [Monitoring Amazon S3](#) chapter in the *Amazon Simple Storage Service User Guide*.

## Contents

- [Logging](#)
- [Bucket report](#)

## Logging

File Transfer can log messages to external files. To do this, define `logging.directory` in the configuration file. When this is defined, File Transfer creates logs that are formatted by date in the specified directory.

`logging.log_severity` should always be defined in the configuration file when `logging.directory` is defined. `logging.directory` can have the following values: `info`, `warn`, `error`, or `fatal`. All messages seen in the console output logged to a file as long as the message has the same, or higher, severity as `logging.log_severity`. The log files are located in the specified log directory.

File Transfer generates two types of log files: `filetransfer.log` and `crash.log`. `filetransfer.log` is a general, regular operations log file. It contains all of the messages and events that are logged given the users configuration for severity. The default severity level is `info`. This is same level of information that the user sees on the GUI. The `crash.log` is only generated if the command line interface (CLI) application crashes.



# Bucket report

The Bucket report provides you with a detailed report of the content that exists under a specified transfer configuration. This can be helpful if you want to see what exists in your bucket. You can generate an Bucket report by using the File Transfer GUI or CLI.

## GUI

### To create an inventory report by using the GUI

1. Open File Transfer.
  - a. Go to the **Start Menu** and search for **File Transfer**.
  - b. Select **Nimble Studio File Transfer** from the list.
2. On the landing page, the **Local** file browser displays on the left, and the **S3 Bucket** file browser displays on the right.
3. In the **S3 Bucket** file browser, choose **Bucket Report**.
4. A **Generate Bucket Report** popup displays. Select your desired **Remote configuration** and **Output format**.
5. Then, choose **Generate report**.
6. The report will now appear in the **Reports** tab at the bottom of the screen.
7. Select an output format.

## CLI

### To create a Bucket report by using the CLI

1. Open a terminal.
2. Run the following command to generate an inventory report of all assets in a transfer profile's corresponding bucket and prefix: `filetransfer inventory [remote configuration] [options]`
  - a. The following is an example command: `filetransfer inventory [remote configuration] --output-format yaml`
  - b. `--output-format`: Output format for report. The default format is JSON. Supported formats: YAML, CSV, JSON, or XML

# Troubleshooting Nimble Studio File Transfer

If you experience problems using Nimble Studio File Transfer, use the following information to help you troubleshoot your issues.

We recommend that you turn on logging by following the instructions in [Logging](#).

## Contents

- [Generate a support file](#)
- [Troubleshooting the GUI](#)
- [Troubleshooting the CLI](#)

## Generate a support file

You can generate a support file to help you troubleshoot. A support file is a zip file that you can provide to a support engineer.

### GUI

#### To generate a support file using the graphical user interface (GUI)

1. Open File Transfer.
  - a. Go to the **Start Menu** and search for **File Transfer**.
  - b. Select **Nimble Studio File Transfer** from the list.
2. Select the dropdown menu in the upper right of the screen and choose **Support**.
3. A file browser menu opens. Choose where you want to download the file.

### CLI

#### To generate a support file using the CLI

- Open a terminal and run the following command: `filetransfer support-file`
  - a. This will generate a zip file in `C:\Users\username\.filetransfer\support-files\support-file-20230310-110834.zip` (Windows)

or in `/Users/username/.tiletransfer/support-files/supportfile-20230227-185212.zip` (Linux & macOS).

- b. The CLI will output the path to the generated file.

## Troubleshooting the GUI

Many errors with the GUI can be resolved with the command line interface (CLI) troubleshooting section. If you are receiving errors within the GUI, try the following steps:

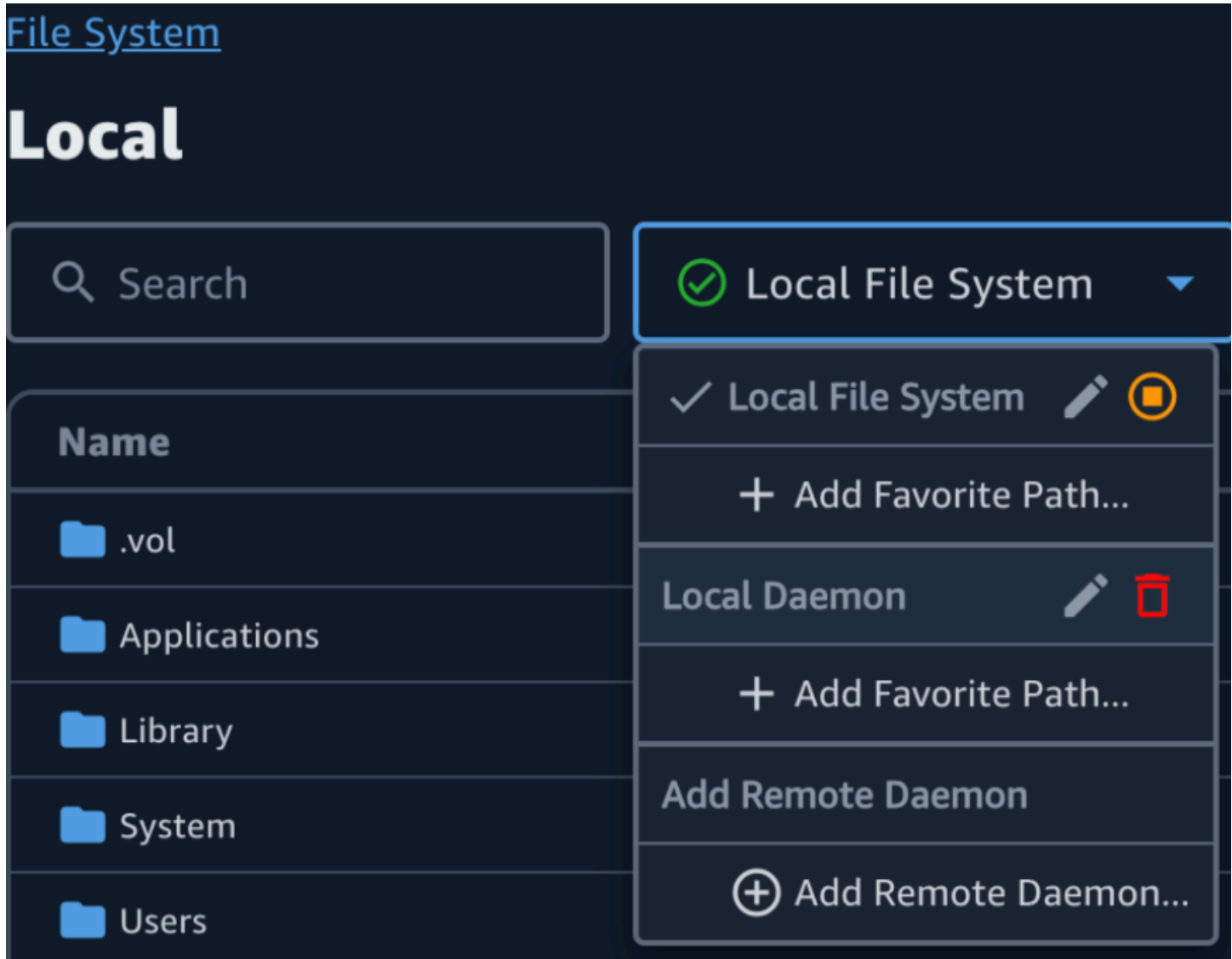
1. Restart File Transfer.
2. Open terminal on macOS or `cmd.exe` on Windows.
3. Run the following command to start an active session: `filetransfer daemon`
4. Begin an upload as you usually do. After you receive an error in the application, check the CLI window. An error should display there.

You can troubleshoot this error in the [Troubleshooting the CLI](#) section.

## File Transfer is unable to connect after upgrading from v1.x to v2.0

**Problem:** You upgraded from File Transfer v1.x to v2.0 and the File Transfer GUI won't enter the **Connected** status.

**Solution:** Delete **Local Daemon** from the **Local File System Dropdown** menu. We've updated the naming of components and some customers may be affected depending on their prior configuration.



## File Transfer is unable to connect

**Problem:** The File Transfer GUI won't enter the **Connected** status.

**Solution:** Update the YAML file.

1. Open the configuration file in your preferred text editor.
  - The configuration file is located in `C:\Users\username\.filetransfer\configuration.yaml` (Windows) or `~/.filetransfer/configuration.yaml` (Linux & macOS).
2. Verify that `api_server.enabled` exists in the file and that it is set to `true`.

- a. If it is set to `false`, the GUI will be unable to communicate with the File Transfer CLI and all GUI functionality will be disabled.
- b. If `api_server.enabled` isn't defined in the `configuration.yaml`, it defaults to `true`.

## Troubleshooting the CLI

### Expired or invalid credentials

**Problem:** If the credentials that you provided File Transfer have issues, you will receive one of the following errors.

```
FATAL [*202X-XX-XX XX:XX:XX]* Failed establishing a session to
AWS:InvalidAccessKeyId: The AWS Access Key Id you provided does not exist *in*
our records. status code: 403, request id: FFYEFCKZX6F1YN8H, host id: aFtP0ImvXdJQ
+Ukf8SYRobDx4xmZsikoJUyJszJf3Wv74w0Q5cP9TCDz/YLKwSi53hc0hBScd58*=*
or
FATAL [*202X-XX-XX XX:XX:XX]* Failed establishing a session to AWS:ExpiredToken: The
provided token has expired. status code: 400, request id: 130NC8C984YZJMjH, host id:
j7aA3Zs/0/H3QMYeoDv5Y62o7Mu/9tvi5m7jUVqTnveLZX4qr1/bKJl1j3dLVnhVda/WaUbEg08*=*
```


**Solution:** Refresh the credentials for the AWS profile by following the instructions in the [Configuration and credential file settings](#) page in the AWS Command Line Interface User Guide.

### Invalid transfer profile

**Error:** : FATAL [202X-XX-XX XX:XX:XX] Invalid transfer profile. Valid transfer profiles:

**Problem:** You're using a remote configuration name that hasn't been set up yet.

**Solution:** Update the remote configurations.

1. Choose the dropdown menu  ). Then choose **Settings**.
2. If there aren't any remote configurations listed under the **Valid remote configurations** section, add a remote configuration by following the instructions in [Step 2: Configure File Transfer](#).

3. If there is a remote configuration, make sure that you haven't misspelled the name of that remote configuration.
4. If you haven't misspelled anything, check the "Valid remote configurations: " part of the error to see if a specific remote configurations is listed.
5. If you still don't see your remote configuration, make sure that your YAML is properly formatted and that you are editing the correct YAML file. The YAML file is tied to the user signed in.

### **Important**

On Windows, don't run CMD.exe or PowerShell as an administrator. If you do, your computer will try to read from a configuration file that isn't in the local user file.

## TCP I/O

**Error:** FATAL [202X-XX-XX XX:XX:XX] Unrecoverable error: retryable: retryable: RequestError:

**Problem 1:** Your computer disconnected from the internet and it lost connection to the S3 bucket.

**Solution 1:** In this case, check for a network outage or any firewall restrictions.

**Problem 2:** The drive where the media is stored is being stored isn't able to handle the load that File Transfer is placing on it. This causes a loss in connection to the media. This can be common among network drives.

**Solution 2:** Lower the max active transfers and the number of threads to 1 and try the upload again.

## GUI

### To lower the max active transfers and the number of threads to 1 by using the GUI

1. Open File Transfer.
  - a. Go to the **Start Menu** and search for **File Transfer**.
  - b. Select **Nimble Studio File Transfer** from the list.
2. Choose the dropdown menu in the upper right of the screen, and then choose **Settings**.
3. In the **S3 settings** section, change **Max active transfers** and **Number of threads** to **1**.

4. Choose **Save** and retry your upload.

## CLI

### To lower the max active transfers and the number of threads to 1 by using the CLI

1. Open the configuration file with any text editing software on your computer.
  - a. Windows: Navigate to the User/<your username> folder on your computer. Open the `.filetransfer` folder and open the `filetransfer.yaml` file with a text editor.
  - b. macOS: Enter **Cmd+Shift+G**. Then enter `~/filetransfer`. Open the `filetransfer.yaml` file with a text editor.
  - c. Linux: Open the `filetransfer.yaml` file using any text editor. The file is located in `~/filetransfer/configuration.yaml`.
2. Update the values of `max_active_transfers` and `threads` to 1.
3. Save the configuration file.

Slowly raise the values of max active transfers and number of threads until you reach a configuration that won't overpower your drive.

## Absolute path

**Error:** WARN [202X-XX-XX XX:XX:XX] Absolute paths are not supported, ignoring /media/drive

**Problem:** Receiving this warning means that you are using an absolute path that isn't supported. An absolute path contains the drive letter. For Windows this is `C:\`. For Linux and macOS, this is the leading `/`.

**Solution:** If you're at the root level, remove the leading `C:\` (Windows) or `/` (Linux & macOS). If not, replace the path relative to your current working directory.

## Unable to open connection

**Error:** Unable to open connection.

**Problem 1:** Another File Transfer application is running.

**Solution 1:** Close any other running File Transfer applications. Alternatively, you can change `api_server.enabled` to `false` in the configuration file.

**Problem 2:** File Transfer is trying to listen on a port that you can't listen to. This can happen if your user doesn't have permission to listen to the port, or if you are using port 1023 or lower. These are considered privileged ports. These ports require that you run as an administrator to listen to them.

**Solution 2:** Make sure that whoever is running File Transfer has the permission to listen to these ports. You can also change the port to 1024 or higher.

**Problem 3:** Another program is using the same port.

**Solution 3:** Stop the other program that is using the same port.



# Security in Nimble Studio File Transfer

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from data centers and network architectures that are built to meet the requirements of the most security-sensitive organizations.

Nimble Studio File Transfer uploads files to Amazon Simple Storage Service (Amazon S3) by using Amazon S3 APIs. To use File Transfer securely, follow the [Security Best Practices for Amazon S3](#) section in the *Amazon Simple Storage Service User Guide*. For more information about securing your S3 resources, see the [Amazon S3 security](#) chapter.

For information about what you should expect from AWS and what is your responsibility, see [Shared Responsibility Model](#).

# Support for Nimble Studio File Transfer

There are several ways to get the help you need when you encounter an issue deploying or using Nimble Studio File Transfer. See the following sections to learn about the different support options that are available to you.

## Topics

- [Amazon Nimble Studio Support](#)
- [AWS Premium Support plans](#)
- [AWS Support Center](#)

## Amazon Nimble Studio Support

Get expert guidance and assistance in achieving your objectives. Amazon Nimble Studio Support provides the help you need to achieve success. Nimble Studio support is available from 9am — 5pm CST. For more information, visit [aws-nsft.zendesk.com](https://aws-nsft.zendesk.com).

## AWS Premium Support plans

AWS Premium Support is available 24/7 and offers reduced wait times for support response. You will have contact options that include email, chat, or phone. Our Support plans are designed to give you the right tools and access to expertise so that you can be successful with leveraging AWS to help you optimize performance, manage risk, and keep costs under control. For more information about AWS Support plans, see [Compare AWS Support Plans](#) .

For more information about how AWS can support you, visit the [Contact us](#) page.

## AWS Support Center

The [AWS Support Center](#) gives you access to a variety of resources. There are links to the knowledge center, knowledge center videos, AWS documentation, plus training and certification.

# Release notes for Nimble Studio File Transfer

This page contains all of the Nimble Studio File Transfer release notes, showing the latest release date first.

Release	Version	Changes
March 7, 2024	v2.5.0	<a href="#">Nimble Studio File Transfer 2.5.0 release notes - March 7, 2024</a>
December 27, 2023	v2.1.0	<a href="#">Nimble Studio File Transfer 2.1.0 release notes - December 27, 2023</a>
December 1, 2023	v2.0	<a href="#">Nimble Studio File Transfer 2.0 release notes - December 1, 2023</a>
July 6, 2023	v1.1.0	<a href="#">Nimble Studio File Transfer 1.1.0 release notes - July 6, 2023</a>
May 19, 2023	v1.0.1	<p>Nimble Studio File Transfer (v1.0.1) is released with the following updates:</p> <ul style="list-style-type: none"><li>• Added an extra security check to the local daemon.</li><li>• Added a loading indicator when browsing S3 buckets.</li><li>• Moved Studio ID to Daemon Configuration Settings.</li><li>• Added specific error messages to provide more</li></ul>

Release	Version	Changes
		<p>visibility when browsing S3 buckets.</p> <ul style="list-style-type: none"><li>• Progress bar now includes a percentage and shows bytes downloaded/bytes remaining.</li><li>• Added a link to the support portal.</li><li>• Simplified transfer profile configurations.<ul style="list-style-type: none"><li>• Added the following sections: General and Advanced. All settings in General are required for NSFT functionality.</li><li>• Removed “Credentials” and “Paths” sections. Configuration settings were merged into the General and Advanced sections where applicable. All functionality has been preserved.</li><li>• Added info tabs to both sections.</li><li>• Removed tooltips for all fields.</li><li>• Renamed “Local” to “Local Directory”.</li><li>• Renamed “Remote” to “S3 Bucket Prefix”.</li></ul></li></ul>

Release	Version	Changes
April 14, 2023	v1.0.0	Nimble Studio File Transfer (v1.0) is released. With this release, Nimble Studio customers can transfer production files into and out of Amazon Simple Storage Service (Amazon S3) using the graphical user interface (GUI) and command line interface (CLI).

## Nimble Studio File Transfer 2.5.0 release notes - March 7, 2024

This page contains the release notes for Nimble Studio File Transfer 2.5.0.

### Major updates

- Added new feature for bandwidth throttling to control the target speed at which File Transfer transfers files.
- Added support for S3 Access Points / S3 VPC Endpoints.
- Added feature to enable resubmitting previously initiated jobs.
- Added an exit prompt that allows the daemon to continue running and maintain file transfers or active hot folders while GUI is closed.
- Added right-click menu features to create child folders, delete/rename files, and set a navigation starting directory for both Local and S3 file browsers.
- Added right-click menu features to configure hot folders and open files in the Local file browser.
- Added the ability to manage file actions by the administrator. Please refer to documentation for more information.
- Checksumming files when uploading can now be disabled.
- Checksumming will be disabled on newly created Remote Configurations. This can be managed for each Remote Configuration.
- File Transfer now reports checksumming progress.

- Added more descriptive job statuses to better reflect the current state of transfers.
- File Transfer no longer supports symlinks as they are unsupported by S3
- Windows only: Added a new helper application to launch the Windows daemon.
- Windows only: File Transfer now supports paths longer than 260 characters when LongPathsEnabled is set in the Windows registry.

## Bug fixes

- Fixed an issue where items selected in one file browser would get deselected when using the other file browser.
- Fixed a display issue with the navigation breadcrumbs.

## Known issues

Linux only: Exporting a Bucket report using the .xlsx file format may result in an unexpected filename.

# Nimble Studio File Transfer 2.1.0 release notes - December 27, 2023

This page contains the release notes for Nimble Studio File Transfer 2.1.0.

## Major updates

- **Auto-refresh** is a new feature for users' local and Amazon S3 bucket file browsers, that eliminates the need to select the **Refresh** button.
- The GUI is now able to reconnect to a running daemon started manually by running **filetransfer daemon** in a terminal window. This allows the user to monitor and manage running transfers, even if the transfers were initiated before connection to the daemon.
- The **default checksum hashing algorithm** for new remote configurations has been changed from MD5 to xxHash to improve transfer job speeds with large files. This doesn't affect any existing remote configurations.
- Added functionality to create folders in both Local and Amazon S3 file browsers in the GUI.

## Bug fixes and minor updates

- Includes important security fixes.
- Improved GUI performance when conducting transfers containing large numbers of files.

## Known issues

If you are upgrading from a previous version earlier than 2.1.0, your daemon bookmarks will be reset.

# Nimble Studio File Transfer 2.0 release notes - December 1, 2023

This page contains the release notes for Nimble Studio File Transfer 2.0.

## Major updates

- The new **"drag and drop" Graphical User Interface (GUI)** helps users browse, transfer files between their local file system and Amazon S3, and monitor transfer job progress.
- **Jobs** is a new feature that groups individual transfers so users can pause, resume, or cancel one or more jobs, while allowing other jobs to continue transferring.
- **Logs** is a new feature that creates daemon logs for users to view in the GUI.
- The **Hot Folder** feature has been reimaged for an improved user experience with more flexibility.
- **Known Issue:** If you are upgrading from a previous version, your daemon bookmarks will be reset.

## Bug fixes and minor updates

- Miscellaneous bug fixes and usability improvements.
- Renamed "Transfer Profile" to "Remote Configuration" to improve clarity.
- Changed the configuration file layout by transitioning some configuration parameters from global settings to "per remote configuration". For example, the checksum algorithm, filtering, and sorting options can now be set individually instead of globally.

# Nimble Studio File Transfer 1.1.0 release notes - July 6, 2023

This page contains the release notes for Nimble Studio File Transfer 1.1.0.

## Major updates

- **First time setup wizard** guides users through setting up Nimble Studio File Transfer for the first time.
- **Transfer auto tuning** is a new feature that automatically adjusts transfer settings based on file size. Users can rely on File Transfer to set these values and don't need to manually optimize transfer settings. Transfer auto tuning improves speeds for transfers containing mixed file sizes.

## Bug fixes and minor updates

- Miscellaneous bug fixes and usability improvements.
- Renamed "Daemon Configuration" to "Settings" to improve clarity.



# Document History for Nimble Studio File Transfer User Guide

The following table describes the documentation for this release of Nimble Studio File Transfer.

- API version: latest
- Latest documentation update: March 11, 2024

Change	Description	
End of support notice	End of support notice: On October 22, 2024, AWS will discontinue support for Nimble Studio File Transfer. After October 22, 2024, you will no longer be able to access the Nimble Studio File Transfer console or Nimble Studio File Transfer resources.	October 2, 2024
Updated guide and release notes for File Transfer v2.5.0	File Transfer <a href="#">2.5.0 release notes</a> have been added and the user guide has been updated for the release of 2.5.0.	March 11, 2024
Release notes for File Transfer v2.1.0	File Transfer <a href="#">2.1.0 release notes</a> have been added.	December 27, 2023
Updated guide for File Transfer v2.0	<b>File Transfer User Guide</b> has been updated for the release of File Transfer 2.0.	December 1, 2023
New service and guide	This is the initial release of File Transfer and the <b>File Transfer User Guide</b> .	April 14, 2023

# AWS Glossary

For the latest AWS terminology, see the [AWS glossary](#) in the *AWS Glossary Reference*.