Aleksandar Karamfilov

Founder @ Pragmatic

Senior QA Manager @ Kiteworks

The game of files

Moving files at scale

Managed File Transfer

"Managed File Transfer (MFT) is a technology platform that allows organizations to reliably exchange electronic data between systems and people in a secure way to meet compliance needs. These data movements can be both internal and external to an enterprise and include various types, including sensitive, compliance-protected or high-volume data."

Managed File Transfer (server)
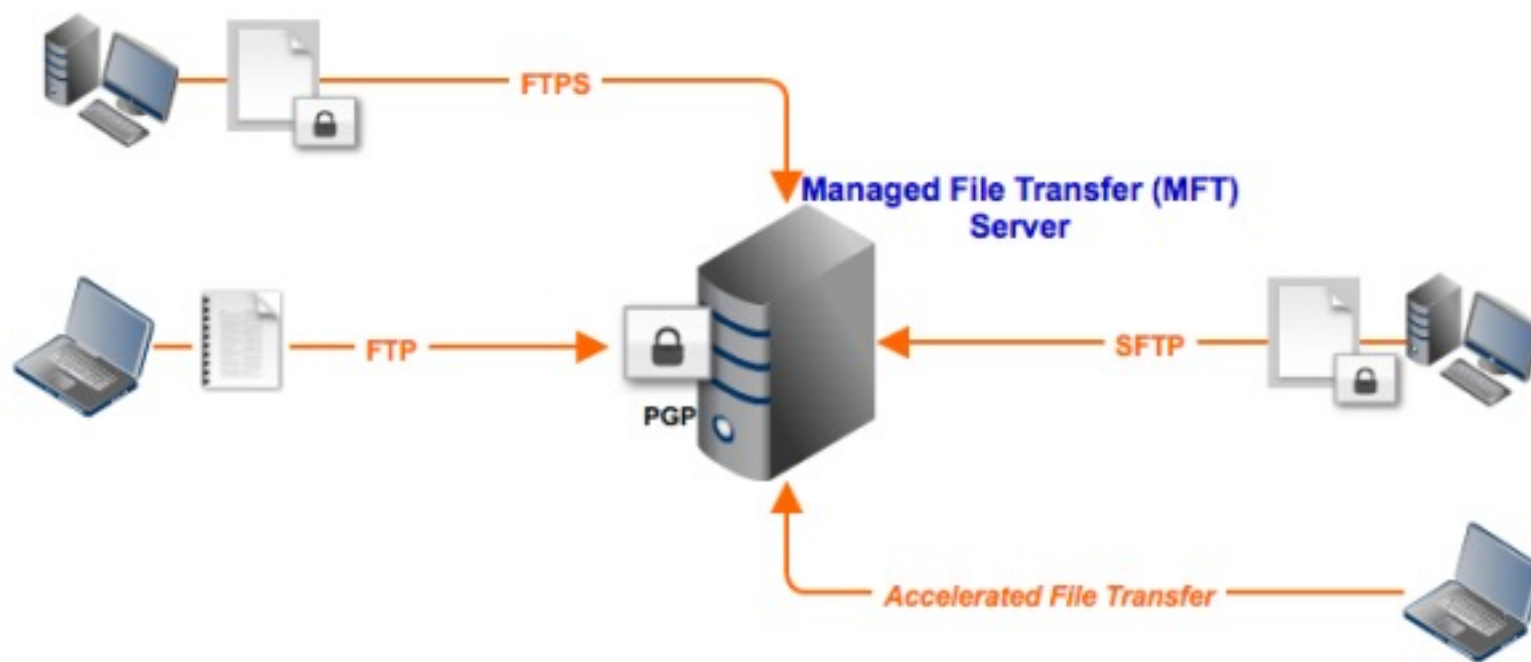
Variety of sources/destinations

- SFTP
- SMB
- FTPS
- Amazon S3
- Box
- Dropbox
- Kiteworks
- You name it

Available in different forms

- Desktop client
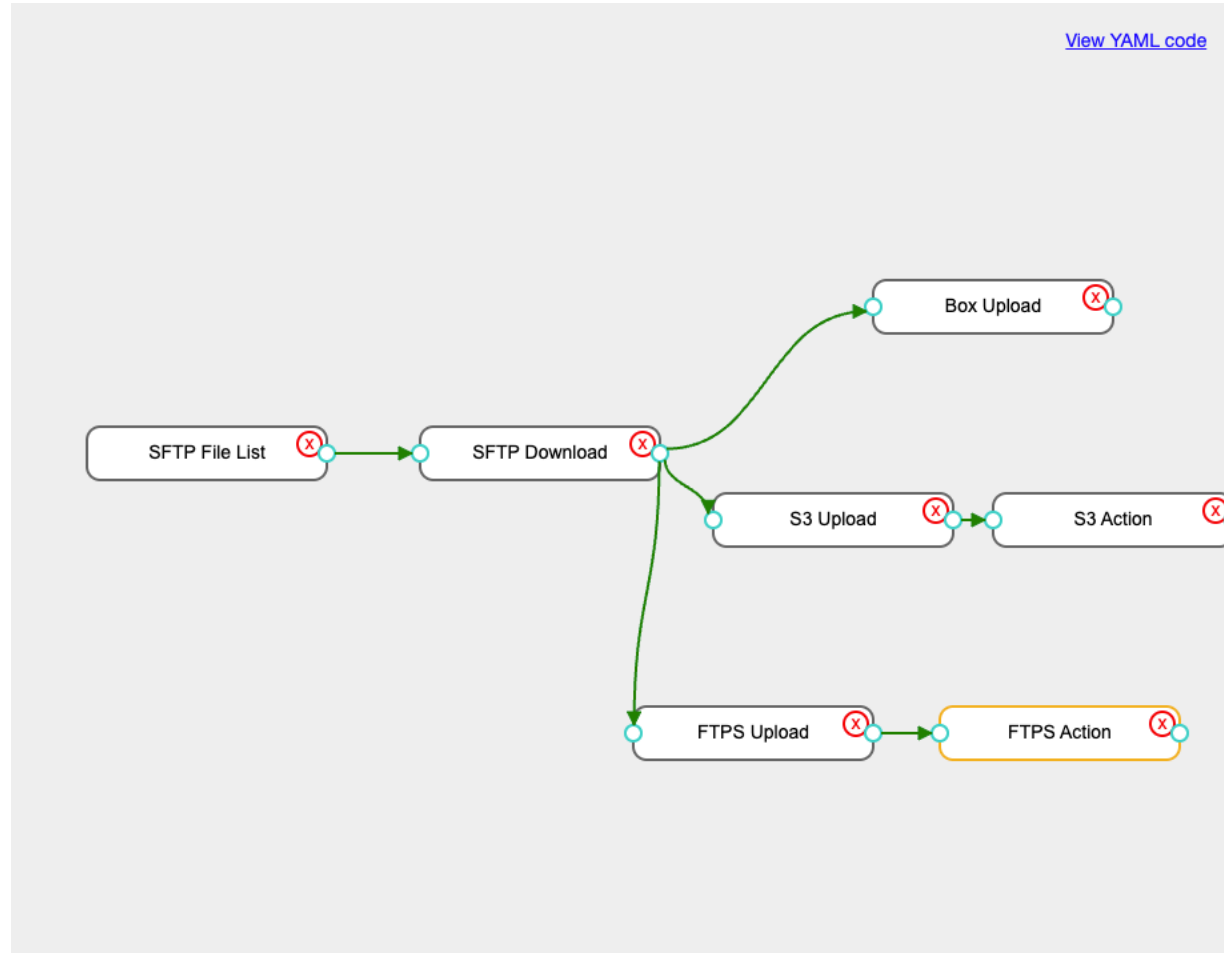- Web server (self-hosted)
- Cloud based (as service)

Directed Acyclic Graph (flow)

Следете актуалните обяви за **Software QA**

File and directory utilities

Because we need something to move

Generating files with specific size

```
truncate -s 200M 200MB.txt
```

```
dd if=/dev/zero of=output.dat  bs=24M  count=1
```

Copy file multiple times

```
for i in {1..10000};
do
 cp source.txt copy${i}.txt;
done
```

Rename multiple files

```
for f in *.html; do
    mv -- "$f" "${f%.html}.php"
done
```

```
rename 's/.html/.php/' \*.html
```

```
rename -f 's/.html/.php/' \*.html
```

Change directory permissions

```
chmod 777 foldername will give
read, write, and execute
permissions for everyone.

chmod 700 foldername will give
read, write, and execute
permissions for the user only.

chmod 327 foldername will give
write and execute (3) permission
for the user, w (2) for the
group, and read, write, and
execute for the users.
```

Create deep folder structure

```
mkdir -p parent/child/grandchild
```

Check disk space

```
shows disk space in human-readable format:
df -h
```

Copy files from the host to remote machine

```
scp file.txt remote_username@10.2540.100.10:/remote/directory
```

Connect to sftp server

```
sftp remote_username@server_ip_or_hostname
```

```
get filename.zip
```

Using the MFT automation

We can create DAG which performs specific operations on files (copy, delete, rename, move)

Manual testing

How to design good manual tests quickly and effectively

Test design

- Input/Output domain testing
- State testing
- Decision testing
- Combinatorial testing
- Special use case testing
- Defect based testing

Input/Output domain testing

- Equivalence partitioning
- Boundary value analysis
- Best representative

State testing

MFT systems are basically specific implementation of a state diagram.

Executing flows/directed graphs covers n-switch out of the box

Decision/Business logic testing

- Identify conditions
- Identify actions
- Create decision tables
- Create tests from the tables

Decision testing (example)

- Conditions:
    - Folder exists
    - User has write permissions
    - Enough disk space

- Actions:
    - Move files
    - No write permissions error
    - Disk not sufficient error
    - Folder not found error

Decision testing (example)

	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Test 8
Folder exists?	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE
User has write permissions?	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE
Disk space is sufficient?	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE
Move files	X							
No write permissions error			X	X				
Disk not sufficient error		X						
Folder not found error					X	X	X	X

	Test 1	Test 2	Test 3	Test 4	Test 5
Folder exists?	TRUE	TRUE	TRUE	TRUE	FALSE
User has write permissions?	TRUE	TRUE	FALSE	FALSE	TRUE
Disk space is sufficient?	TRUE	FALSE	TRUE	FALSE	TRUE
Move files	X				
No write permissions error			X	X	
Disk not sufficient error		X			
Folder not found error					X

Combinatorial testing (pairwise)

- Define input model (model.txt)
- Use combinatorial tool like PICT
- Create combinations
- Create detailed tests from the combinations

Combinatorial testing (model)

```
Source: SFTP, FTPS, Box, Kiteworks, SMB, S3
Destination: SFTP, FTPS, Box, Kiteworks, SMB, S3
Schedule: no, minutes, hours, days, custom
Multi destination: yes, no
Action on source: yes, no
Custom action: rename, copy, move, delete
On name conflict: ignore, overwrite, create new version
External scan: yes, no
Encrypt: yes, no
Regex: no, on source, on destination
Timestamp: no, suffix, prefix
```

Combinatorial testing (pairs)

Source	Destination	Schedule	Multi destination	Action on source	Custom action	On name conflict	External scan	Encrypt	Regex	Timestamp
Box	SFTP	no	yes	no	move	overwrite	no	no	on source	no
S3	S3	minutes	no	yes	copy	create new version	yes	yes	no	prefix
FTPS	SMB	hours	yes	yes	delete	ignore	no	yes	on destination	suffix
Box	Box	days	no	no	rename	ignore	yes	no	no	suffix
FTPS	SFTP	custom	no	yes	rename	overwrite	yes	yes	on source	prefix
FTPS	S3	custom	yes	no	delete	create new version	yes	no	on destination	no
SFTP	SFTP	no	no	no	copy	create new version	no	yes	on destination	suffix
FTPS	Kiteworks	days	yes	yes	move	ignore	no	no	no	prefix
Box	FTPS	minutes	no	no	delete	overwrite	no	yes	on destination	prefix
Kiteworks	SMB	hours	no	no	copy	overwrite	yes	no	no	no
SFTP	S3	minutes	yes	yes	move	ignore	yes	yes	on source	no
SMB	S3	days	yes	yes	rename	overwrite	no	yes	on source	suffix
SMB	Box	hours	no	yes	move	create new version	no	no	on destination	no
S3	SFTP	custom	yes	no	delete	ignore	no	no	on source	suffix
S3	Kiteworks	no	no	no	rename	overwrite	yes	yes	on destination	no
FTPS	FTPS	days	yes	yes	copy	create new version	yes	no	on source	suffix
Box	S3	no	no	yes	delete	create new version	yes	no	no	suffix
SFTP	FTPS	hours	yes	yes	rename	ignore	no	no	no	prefix
S3	FTPS	custom	no	no	move	ignore	yes	no	no	no
FTPS	Box	no	yes	yes	delete	overwrite	no	yes	on source	prefix
S3	SMB	minutes	no	no	rename	create new version	no	no	on source	prefix
Kiteworks	SFTP	minutes	yes	yes	move					
SMB	SMB	no	yes	no	copy					
Box	SMB	custom	no	no	copy					
SMB	Kiteworks	hours	no	no	delete					
S3	Box	hours	no	no	copy					

```
bash-3.2$ pict model.txt /s
Combinations:   709
Generated tests:51
Generation time:0:00:00
```

Special case testing

- Filename collisions
- Missing permissions to source/destination folder
- Exceeding max filename length via rename/timestamp
- Exceeding max filename length copying from different source
- Network interruptions
- Duplicate prevention
- Partial uploads

Special cases

- Path/directory traversal (../../)
- Forbidden listing on destination
- Tar uploads (compression)
- File versioning
- File overwrites (modified date)
- Exceeding disk space
- Exceeding single file max size

Special cases

- File extensions(suffix/prefix) in combination with features like timestamp/encryption

- Deep folder structures (exceeding path length)

- Regex to match source/destination folders

- External file services taking very long

- Folder/File name case sensitivity

- File checksums

Special cases

- File integrity being impacted (modification)
- Listing root directory (/)
- Recursive upload (include nested)
- Temporary storage/logs cleanup
- Filenames and extension with special chars
- Moving empty files/folders
- Using hostnames instead of IPs

Special cases

- Time zones and logs
- Stuck executions
- Event listeners
- Email notifications
- Availability/Concurrency/Disaster recovery
- Retries on failure
- Partial success
- Upgrades

Test Automation

Because the older you get the less time you have left

High level approach

- Define high priority tests to be automated
- Create client for every source/destination
- Create methods for basic operations with folders, files
- Create directed graphs/flows as yaml/json/xml (avoid UI automation for the flow designers)
- Automate the MFT tests (on API level if possible)
- Integrate in CI/CD process

Java clients (SMB, FTPS, SFTP)

```java
SSHClient sshClient = SSHService.getClient(HOST, PORT, USERNAME, PASSWORD);
SFTPClient sftpClient = sshClient.newSFTPClient();
SFTPService sftpService = new SFTPService(sftpClient);
sftpService.createDirectory("/sftp/automated");
sftpService.uploadFile("src/test/resources/file.txt", "/sftp/automated/alex.txt");
sftpClient.close();
sshClient.close();
```

```java
FTPSClient ftpsClient = FTPSClientService.getClient(HOST, PORT, USERNAME, PASSWORD);
FTPService ftpService = new FTPService(ftpsClient);
ftpService.createDirectory("Test");
ftpService.uploadFile("src/test/resources/file.txt", "Test/file.txt");
ftpsClient.disconnect();
```

HTTP clients

Most of the file services expose RESTful API.

Communication can be automated with libraries like Rest Assured.

Automated tests for file transfer

```java
//Create ssh client providing correct credentials
SSHClient sshClient = SSHService.getClient(HOST, PORT, USERNAME, PASSWORD);
//Create sftp client
SFTPClient sftpClient = sshClient.newSFTPClient();
SFTPService sftpService = new SFTPService(sftpClient);
//Create directory at the remote sftp server
sftpService.createDirectory("/sftp/automated");
//Upload files from the host (git lfs)
sftpService.uploadFile("src/test/resources/file.txt", "/sftp/source/automated/file.txt");
//Create directed graph using yaml file
Response createResp = mftAPI.createDAG("/srt/test/resources/sftp_to_sftp.yaml");
//Check that the operation was successfull
Assertions.assertEquals(201, createResp.statusCode())
//Extract DAG id
....
//Run the DAG and wait to finish running (few secs)
Response runResp = mftAPI.runDAG(dagId);
Assertions.assertEquals(200, runResp.statusCode());
...
mftpAPI.waitForRunState(runId, State.FINISHED);
...
//Retrieve files from the directory
DirectoryListing listing = sftpService.list("/sftp/destination/automated");
//Assert all files are moved
...
//Clean
sftpService.deleteDirectory("/sftp/source/automated");
sftpServce.deleteDirectory("/sftp/destination/automated");
...
```

Java libraries

- Rest Assured
- SMBJ
- Apache Commons Net
- SSHJ
- JUnit 5

Performance testing

How quickly you can move files

Thank you!

Contacts:

https://www.linkedin.com/in/akaramfilov

https://github.com/karamfilovs/game-of-files

СЛЕДВАЩО СЪБИТИЕ

Лектор	Дата	Език

Следете актуалните обяви за **Software QA**

DEV.BG