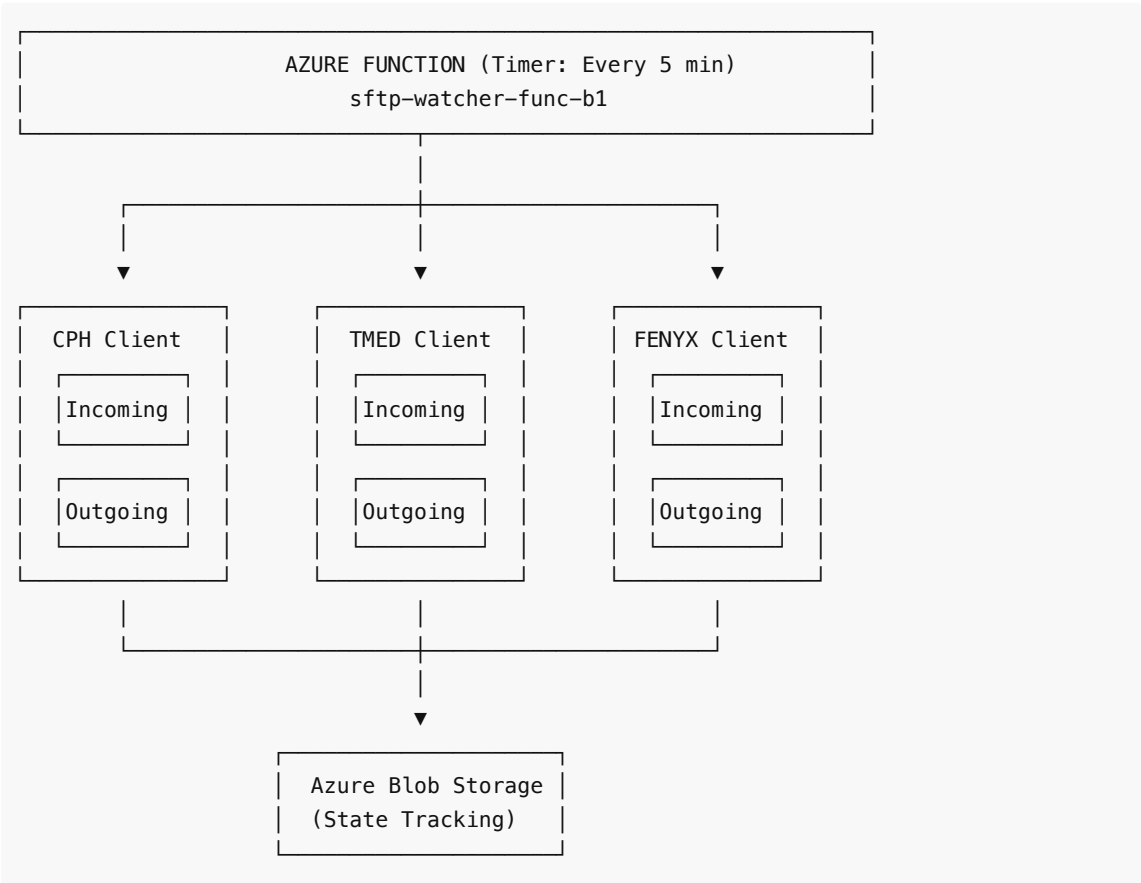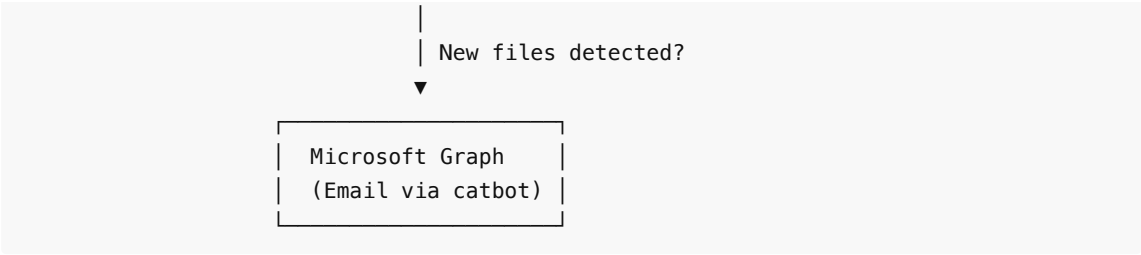# SFTP Watcher Azure Function

Executive Summary

The SFTP Watcher is an **Azure Functions-based file monitoring system** that continuously polls multiple SFTP directories and sends email notifications when new files are detected. It monitors file drops from three organizational clients (CPH, TMED, FENYX) across both incoming and outgoing directions, alerting relevant teams within minutes of file arrival.

Business Value

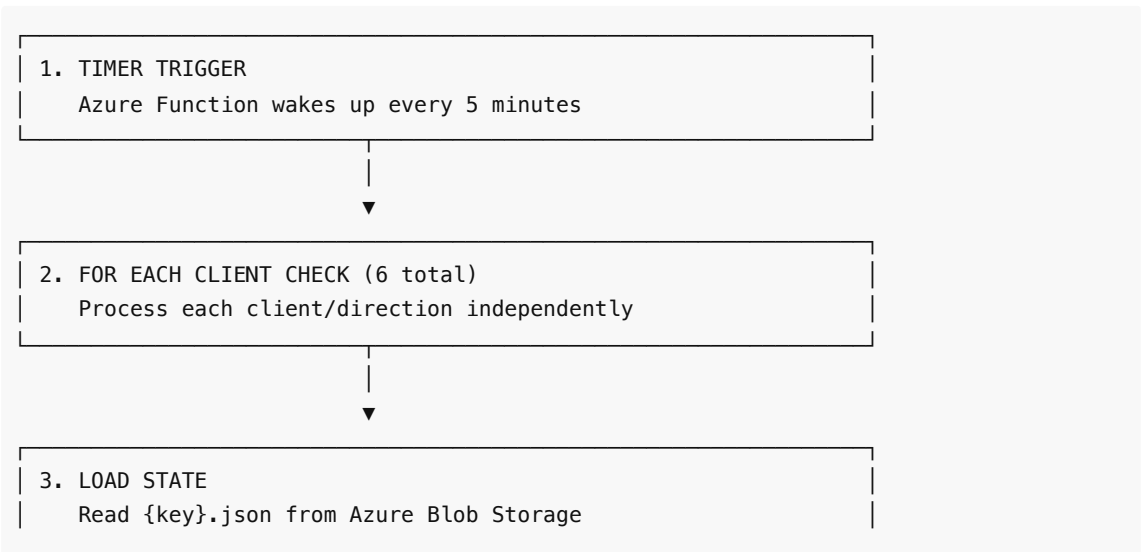| Benefit	Description
Real-Time Awareness	Teams notified within 5 minutes of new file arrivals
Multi-Client Support	Single solution monitors three clients with six directories
Automated Alerting	No manual checking required - emails sent automatically
Reliable State Tracking	Uses Azure Blob Storage to prevent duplicate notifications
Client Isolation	Separate credentials and notification recipients per client

Architecture Overview

```
┌─────────────────────────────────────────────────┐
│         AZURE FUNCTION (Timer: Every 5 min)     │
│               sftp-watcher-func-b1              │
└─────────────────────────────────────────────────┘
                        │
        ┌───────────────┼───────────────┐
        │               │               │
        ▼               ▼               ▼
  ┌───────────┐   ┌───────────┐   ┌───────────┐
  │ CPH Client│   │ TMED Client│  │FENYX Client│
  │ ┌───────┐ │   │ ┌───────┐ │   │ ┌───────┐ │
  │ │Incoming│ │   │ │Incoming│ │   │ │Incoming│ │
  │ └───────┘ │   │ └───────┘ │   │ └───────┘ │
  │ ┌───────┐ │   │ ┌───────┐ │   │ ┌───────┐ │
  │ │Outgoing│ │   │ │Outgoing│ │   │ │Outgoing│ │
  │ └───────┘ │   │ └───────┘ │   │ └───────┘ │
  └───────────┘   └───────────┘   └───────────┘
        │               │               │
        └───────────────┼───────────────┘
                        │
                        ▼
                ┌───────────────┐
                │ Azure Blob Storage │
                │ (State Tracking)   │
                └───────────────┘
```

```
            |
            | New files detected?
            ▼
      ┌───────────────────┐
      |   Microsoft Graph  |
      |  (Email via catbot) |
      └───────────────────┘
```

Monitoring Configuration

Clients and Directories

Client	Direction	State Key	SFTP User	Notification Recipient
CPH	Incoming	cph-in	catsolsasftp.cph	cphincoming@catalystsolutions.com
CPH	Outgoing	cph-out	catsolsasftp.cphout	cphtestemail@catalystsolutions.com
TMED	Incoming	tmed-in	catsolsasftp.tmed	tmedincoming@catalystsolutions.com
TMED	Outgoing	tmed-out	catsolsasftp.tmedout	tmedtestemail@catalystsolutions.com
FENYX	Incoming	fenyx-in	catsolsasftp.fenyx	dlfenyxsftpincoming@catalystsolutions.com
FENYX	Outgoing	fenyx-out	catsolsasftp.fenyxout	dlfenyxsftpout@catalystsolutions.com

Workflow

Processing Cycle (Every 5 Minutes)

```
┌─────────────────────────────────────────────────┐
| 1. TIMER TRIGGER                                 |
|    Azure Function wakes up every 5 minutes        |
└─────────────────────────────────────────────────┘
                      |
                      ▼
┌─────────────────────────────────────────────────┐
| 2. FOR EACH CLIENT CHECK (6 total)               |
|    Process each client/direction independently    |
└─────────────────────────────────────────────────┘
                      |
                      ▼
┌─────────────────────────────────────────────────┐
| 3. LOAD STATE                                    |
|    Read {key}.json from Azure Blob Storage        |
```

```
|       Contains list of previously seen file names       |
|_____|

                          |
                          |
                          ▼
 _____
|                                                         |
| 4. CONNECT TO SFTP                                      |
|    SSH2 connection to Azure Blob SFTP endpoint          |
|    Host: catsolsasftp.blob.core.windows.net             |
|_____|

                          |
                          |
                          ▼
 _____
|                                                         |
| 5. LIST FILES                                           |
|    Check root directory (.) and /incoming subdirectory  |
|    Return only regular files (not directories)          |
|_____|

                          |
                          |
                          ▼
 _____
|                                                         |
| 6. DETECT NEW FILES                                     |
|    Compare current listing against saved state          |
|    Files in listing but NOT in state = NEW              |
|_____|

                          |
                _____|_____
               |                       |
               |                       |
          New files?              No new files
               |                       |
               ▼                       ▼
 _____    _____
|                         |  |                         |
| 7a. SEND NOTIFICATION   |  | 7b. LOG & CONTINUE       |
|     Get OAuth token     |  |     Move to next client  |
|     Send email via      |  |                         |
|     Microsoft Graph     |  |                         |
|_____|  |_____|

               |
               |
               ▼
 _____
|                                                         |
| 8. UPDATE STATE                                         |
|    Add all current files to processed list              |
|    Save updated state to Azure Blob                     |
|_____|
```

External Service Integrations

Azure Blob Storage (SFTP)

- **Host**: `catsolsasftp.blob.core.windows.net`
- **Protocol**: SFTP over SSH2 (port 22)
- **Purpose**: Client file transfer endpoint
- **Accounts**: Six unique user accounts for client isolation

Azure Blob Storage (State)

- **Container**: `sftp-state`
- **Purpose**: Persist processed file lists between function runs
- **Files**: JSON blobs (`cph-in.json` , `tmed-out.json` , etc.)

Microsoft Entra ID (Azure AD)

- **Tenant**: `d39a588a-b5a6-4378-9f5f-f9a0e5484b06`
- **Authentication**: OAuth 2.0 Client Credentials
- **Purpose**: Obtain access token for Microsoft Graph API

Microsoft Graph API

- **Endpoint**:
  `https://graph.microsoft.com/v1.0/users/catbot@catalystsolutions.com/sendMail`
- **Purpose**: Send email notifications
- **Sender**: `catbot@catalystsolutions.com`

Email Notifications

Email Format

From: `catbot@catalystsolutions.com`

To: Client-specific distribution list

Subject: `{CLIENT} {DIRECTION} files`

Body (plain text):

```
Client: CPH
Direction: incoming

New files detected:
• data_export_2026-01-20.csv
• transactions_batch_42.xlsx
• customer_report.pdf
```

Technical Stack

Component	Technology
Runtime	Azure Functions v2.0 (Node.js 20.x)
SFTP Client	ssh2-sftp-client
HTTP Client	isomorphic-fetch
State Storage	@azure/storage-blob
Deployment	GitHub Actions
Timer	Azure Timer Trigger (CRON)

Timer Configuration

Setting	Value
Schedule	`0 */5 * * * *` (every 5 minutes)
Run on Startup	`true`
Function Timeout	10 minutes
Timer Monitor	Enabled

State Management

State File Structure

```
{
  "processedFiles": [
    "file1.csv",
    "report_2026-01-15.xlsx",
    "data_batch_99.txt"
  ]
}
```
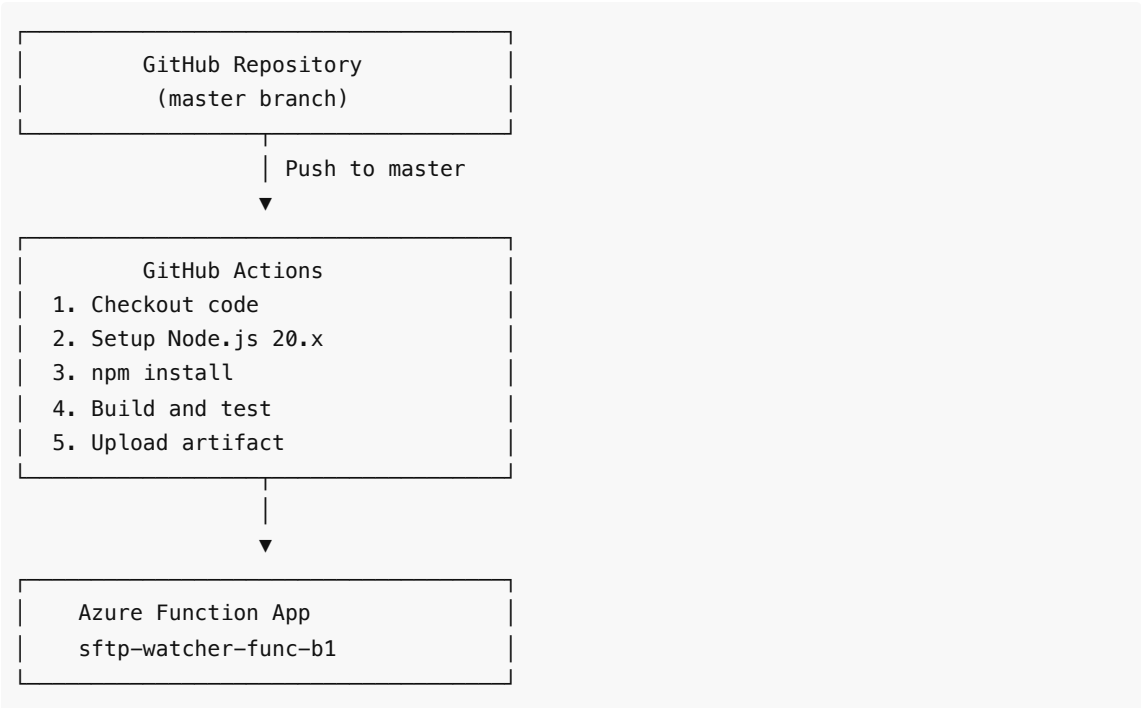
State Behavior

- **Append-only**: Files are added but never removed from state
- **Per-check isolation**: Each client/direction has its own state file
- **Container**: `sftp-state` (created automatically on first run)
- **Persistence**: State survives function app restarts

Project Structure

```
SftpWatcherFunction/
├── CheckSftp/
│   ├── function.json        # Timer trigger configuration
│   └── index.js             # Main function logic
├── config/
│   └── checks.js            # Client configuration
├── shared/
│   ├── blobState.js         # State persistence
│   ├── email.js             # Graph API email sender
│   └── sftp.js              # SFTP client wrapper
├── host.json                # Function runtime config
├── package.json             # Dependencies
└── .github/workflows/
    └── master_sftp-watcher-*.yml   # CI/CD pipeline
```

Deployment Pipeline

```
┌──────────────────────────────────┐
│         GitHub Repository        │
│          (master branch)         │
└──────────────────────────────────┘
                │
                │ Push to master
                ▼
┌──────────────────────────────────┐
│          GitHub Actions          │
│  1. Checkout code                │
│  2. Setup Node.js 20.x           │
│  3. npm install                  │
│  4. Build and test               │
│  5. Upload artifact              │
└──────────────────────────────────┘
                │
                │
                ▼
┌──────────────────────────────────┐
│     Azure Function App           │
│     sftp-watcher-func-b1         │
└──────────────────────────────────┘
```

Error Handling

Scenario	Behavior
SFTP connection timeout	Logged, continues to next client
Directory not found	Returns empty file list, continues
One client fails	Other clients still processed
Email send fails	Logged, state still updated
Connection cleanup error	Swallowed to prevent function failure

Dependencies

Package	Version	Purpose
ssh2-sftp-client	^11.0.0	SFTP protocol client
@azure/storage-blob	^12.19.0	Blob storage for state
@azure/identity	^4.13.0	Azure authentication
isomorphic-fetch	^3.0.0	HTTP client for OAuth
@microsoft/microsoft-graph-client	^3.0.7	Graph API client

Environment Variables

Azure Configuration

- `AzureWebJobsStorage` - Storage account connection string

Azure AD / OAuth

- `TENANT_ID` - Azure tenant ID
- `CLIENT_ID` - Service principal client ID
- `CLIENT_SECRET` - Service principal secret

SFTP Credentials (12 variables)

- `SFTP_HOST` - SFTP endpoint
- `SFTP_CPH_USER` / `SFTP_CPH_PASS` - CPH incoming
- `SFTP_CPHOUT_USER` / `SFTP_CPHOUT_PASS` - CPH outgoing
- `SFTP_TMED_USER` / `SFTP_TMED_PASS` - TMED incoming
- `SFTP_TMEDOUT_USER` / `SFTP_TMEDOUT_PASS` - TMED outgoing
- `SFTP_FENYX_USER` / `SFTP_FENYX_PASS` - FENYX incoming
- `SFTP_FENYXOUT_USER` / `SFTP_FENYXOUT_PASS` - FENYX outgoing

Notification Recipients

- `NOTIFY_CPHINCOMING` , `NOTIFY_CPH`
- `NOTIFY_TMEDINCOMING` , `NOTIFY_TMED`
- `NOTIFY_FENYXINCOMING` , `NOTIFY_FENYX`

Summary

The SFTP Watcher provides automated file monitoring for critical business file transfers. Running every 5 minutes, it ensures teams are notified promptly when new files arrive, enabling faster response times for data processing workflows. The stateless, serverless architecture provides reliability while the per-client isolation ensures security and proper notification routing.

Document generated: January 2026