

FTP-Scan

- [Neue Seite](#)

Neue Seite

FTP-Dienst

Technische Dokumentation für die zentrale Verarbeitung, Validierung und Übertragung von PDF-Dokumenten mit jobbasierter Konfiguration, täglichem Cleanup und Logrotation.

[Pakete](#)[Architektur](#)[Installation](#)[Jobs](#)[Cleanup](#)[Logrotate](#)[Betrieb](#)[Docker](#)

Pakete

Beide Archive enthalten denselben Stand. Das ZIP-Archiv eignet sich besonders für den Transfer von Windows-Systemen, das TAR.GZ-Archiv für Linux.

[ZIP-Paket herunterladen](#) [TAR.GZ-Paket herunterladen](#)

1. Architektur

Zentrale Logik

Ein generischer Starter ruft die gemeinsame PDF-Verarbeitung auf. Es gibt keine individuellen Wrapper pro Job.

Jobbasierte Konfiguration

Ein neuer Prozess benötigt nur eine neue Datei unter `jobs/<jobname>.conf`.

Globale Werte

FTP-Zugang, SMTP, Logging und Standardwerte liegen zentral in `config/global.conf`.

```
/opt/ftp-dienst/  
├─ bin/  
|   ├── run-job.sh  
|   ├── process-pdf-upload.sh  
|   └─ cleanup.sh  
├─ config/  
|   ├── global.conf  
|   └─ cleanup.conf  
├─ jobs/  
|   ├── hoese-bankanhaenge.conf  
|   ├── lobu-gehalt.conf  
|   ├── lobu-personal.conf  
|   ├── einkauf.conf  
|   └─ fibu-allgemein.conf  
├─ lib/  
├─ docs/  
└─ install.sh
```

2. Installation

ZIP

```
unzip ftp-dienst-neu.zip  
cd ftp-dienst-neu  
bash install.sh
```

TAR.GZ

```
tar -xzf ftp-dienst-neu.tar.gz  
cd ftp-dienst-neu  
bash install.sh
```

Wichtig: Das Installationsskript mit `bash install.sh` oder direkt als ausführbare Datei starten. Nicht mit `sh install.sh`, da die Skripte Bash-Funktionen wie `pipefail` verwenden.

Bestehende Dateien `global.conf` und `cleanup.conf` werden bei einer erneuten Installation nicht automatisch überschrieben.

3. Jobs und Konfiguration

Globale Konfiguration

Datei: `/opt/ftp-dienst/config/global.conf`

```
# FTP
FTP_USER="idoms"
FTP_PASS="HIER_EINTRAGEN"
FTP_PASSIVE_MODE="on"

# SMTP
SMTP_SERVER="terminator.spiegelblank.de"
SMTP_PORT="587"
SMTP_USER="ftp_dienst@spiegelblank.de"
SMTP_PASS="HIER_EINTRAGEN"
EMAIL_FROM="ftp_dienst@spiegelblank.de"
EMAIL_NAME="FTP-Dienst der Spiegelblank"
EMAIL_TO="martin.raddei@spiegelblank.de"

# Logging
LOGFILE="/home/ftp/log/ftp_upload.log"
```

Job-Konfiguration

Beispiel `jobs/lobu-gehalt.conf`:

```
JOB_NAME="lobu-gehalt"
FTP_HOST="10.10.99.24"
FTP_REMOTE_DIR="/gehalt#"
LOCAL_SOURCE_DIR="/home/ftp/lobu/gehalt/new"
ARCHIVE_DIR="/home/ftp/lobu/gehalt/cur"
ERROR_DIR="/home/ftp/lobu/gehalt/error"
CHECK_HOST_REACHABILITY="false"
```

Aufruf

```
/opt/ftp-dienst/bin/run-job.sh hoese-bankanhaenge  
/opt/ftp-dienst/bin/run-job.sh lobu-gehalt  
/opt/ftp-dienst/bin/run-job.sh lobu-personal
```

Dry-Run

```
/opt/ftp-dienst/bin/run-job.sh lobu-gehalt --dry-run
```

Neuen Job ergänzen

```
cp /opt/ftp-dienst/jobs/lobu-gehalt.conf \  
  /opt/ftp-dienst/jobs/neuer-job.conf  
nano /opt/ftp-dienst/jobs/neuer-job.conf  
/opt/ftp-dienst/bin/run-job.sh neuer-job --dry-run
```

4. Tägliches Cleanup

Bei der Installation wird `/etc/cron.daily/ftp-dienst-cleanup` angelegt. Das Cleanup läuft damit einmal täglich.

`cleanup.sh` liest automatisch alle Job-Dateien unter `/opt/ftp-dienst/jobs/*.conf` und übernimmt daraus:

Variable	Zweck	Standardfrist
<code>LOCAL_SOURCE_DIR</code>	Eingangsverzeichnis	14 Tage
<code>ARCHIVE_DIR</code>	Erfolgreich verarbeitete Dateien	90 Tage
<code>ERROR_DIR</code>	Fehlerhafte Dateien	30 Tage

Zentrale Einstellungen in `config/cleanup.conf`:

```
CLEANUP_SOURCE_DAYS="14"  
CLEANUP_ARCHIVE_DAYS="90"  
CLEANUP_ERROR_DAYS="30"  
DELETE_EMPTY_DIRS="false"
```

```
RUN_LOGROTATE="true"
```

Ein einzelner Job kann Fristen überschreiben:

```
CLEANUP_SOURCE_DAYS_JOB="7"  
CLEANUP_ARCHIVE_DAYS_JOB="180"  
CLEANUP_ERROR_DAYS_JOB="60"
```

Testlauf

```
/opt/ftp-dienst/bin/cleanup.sh --dry-run
```

5. Logrotate

Die Installation erzeugt `/etc/logrotate.d/ftp-dienst`. Benutzer und Gruppe werden aus der vorhandenen Logdatei beziehungsweise dem Logverzeichnis ermittelt und über die `su`-Direktive eingetragen.

```
/home/ftp/log/ftp_upload.log {  
    daily  
    rotate 14  
    compress  
    delaycompress  
    missingok  
    notifempty  
    dateext  
    su root root  
    create 0640 root root  
}
```

Die konkreten Werte für Benutzer und Gruppe können vom Beispiel abweichen.

Prüfung

```
logrotate -d /etc/logrotate.d/ftp-dienst
```

Rotation erzwingen

```
logrotate -v -f /etc/logrotate.d/ftp-dienst
```

6. Betrieb und Fehlersuche

Prüfung	Befehl
Shell-Syntax	<code>bash -n /opt/ftp-dienst/bin/*.sh</code>
Job testen	<code>/opt/ftp-dienst/bin/run-job.sh JOBNAME --dry-run</code>
Cleanup testen	<code>/opt/ftp-dienst/bin/cleanup.sh --dry-run</code>
Logrotate prüfen	<code>logrotate -d /etc/logrotate.d/ftp-dienst</code>
Log beobachten	<code>tail -f /home/ftp/log/ftp_upload.log</code>

Sicherheit: FTP- und SMTP-Passwörter nicht in Job-Dateien duplizieren. Die zentrale Konfiguration sollte nur für den ausführenden Dienstbenutzer und Administratoren lesbar sein.

7. Docker-Perspektive

Der Dienst kann grundsätzlich containerisiert werden. Die Verzeichnisse unter `/home/ftp`, die globale Konfiguration und die Job-Dateien würden dabei als Volumes eingebunden.

Für einen Docker-Betrieb empfiehlt es sich, Logs nach `stdout`/`stderr` auszugeben und die Rotation über Docker zu steuern, statt `logrotate` im Container zu betreiben.

```
logging:
  driver: json-file
  options:
    max-size: 10m
    max-file: "14"
```

Ein fertiges Docker-Setup ist in den beigefügten Paketen noch nicht enthalten.