

# [SOUTH AFRICA] SciELO Nginx Log Export to FTP --- Operations Manual

## 1. Purpose

This manual documents the daily process used to export the SciELO Nginx access log to an FTP server while keeping a local monthly archive.

The workflow:

1. Nginx writes requests to `scielo-access.log`.
2. `logrotate` rotates the active log daily.
3. The export script reads the newly rotated, uncompressed log without modifying it.
4. The script creates a gzip copy using the required filename convention.
5. The gzip file is validated.
6. The file is uploaded to the FTP server.
7. Only after the FTP server confirms a successful transfer is the generated copy stored in the local monthly archive.

The original Nginx/logrotate files remain under logrotate control.

---

## 2. File Naming Convention

The exported filename is:

```
YYYY-MM-DD_scielo.za.log.gz
```

Example:

```
2026-10-09_scielo.za.log.gz
```

The local archive is organized by year and month:

```
/var/log/nginx/scielo-org-za/YYYY-MM/
```

Example:

```
/var/log/nginx/scielo-org-za/2026-10/2026-10-09_scielo.za.log.gz
```

Using `YYYY-MM` instead of only the month number prevents files from different years from being mixed.

## 3. Existing Nginx Log Rotation

The server currently uses the following logrotate configuration:

```
/var/log/nginx/*.log {  
    create 0640 nginx root  
    daily  
    rotate 10  
    missingok  
    notifempty  
    compress  
    delaycompress  
    sharedscripts  
    postrotate  
        /bin/kill -USR1 `cat /run/nginx.pid 2>/dev/null` 2>/dev/null || true  
    endscrip  
}
```

## Important behavior

Because `delaycompress` is enabled, the most recently rotated file remains uncompressed for one rotation cycle.

For example:

```
scielo-access.log  
scielo-access.log-20261009  
scielo-access.log-20261008.gz  
scielo-access.log-20261007.gz
```

The export process must **not** modify or remove `scielo-access.log-20261009`. It creates an independent compressed copy.

---

## 4. FTP Credentials

Store FTP credentials separately from the export script.

Create:

```
vi /root/.scielo-ftp.conf
```

Contents:

```
FTP_USER='your_username'  
FTP_PASS='your_password'
```

Protect the file:

```
chmod 600 /root/.scielo-ftp.conf  
chown root:root /root/.scielo-ftp.conf
```

Verify:

```
ls -l /root/.scielo-ftp.conf
```

Expected permissions:

```
-rw-----. 1 root root ... /root/.scielo-ftp.conf
```

Do not commit this file to Git or copy its contents into operational documentation.

“ **Security note:** traditional FTP does not encrypt credentials or transferred data. If the service later supports SFTP or FTPS, migration to an encrypted transport is recommended.

---

## 5. Daily Export Script

Create:

```
vi /usr/local/sbin/export-scielo-za.sh
```

Use the following script:

```
#!/bin/bash

set -u
umask 027

LOGDIR="/var/log/nginx"
ARCHIVE="$LOGDIR/scielo-org-za"

HOST="ftp.ratchet.scielo.org"
CREDENTIALS="/root/.scielo-ftp.conf"

# Load FTP credentials
if [ ! -r "$CREDENTIALS" ]; then
    echo "ERROR: credentials file not found: $CREDENTIALS"
    exit 1
fi

source "$CREDENTIALS"

if [ -z "${FTP_USER:-}" ] || [ -z "${FTP_PASS:-}" ]; then
    echo "ERROR: FTP_USER or FTP_PASS is not defined."
    exit 1
fi

# The date suffix generated by logrotate
DATA=$(date +%Y%m%d)

SOURCE="$LOGDIR/scielo-access.log-$DATA"

if [ ! -s "$SOURCE" ]; then
    echo "ERROR: rotated log was not found or is empty:"
    echo "$SOURCE"
    exit 1
fi
```

```
YEAR="${DATA:0:4}"
MONTH="${DATA:4:2}"
DAY="${DATA:6:2}"

FORMATTED_DATE="${YEAR}-${MONTH}-${DAY}"

FILENAME="${FORMATTED_DATE}_scielo.za.log.gz"
DESTDIR="$ARCHIVE/${YEAR}-${MONTH}"
DEST="$DESTDIR/$FILENAME"
TMP="$DESTDIR/.${FILENAME}.tmp"

mkdir -p "$DESTDIR"

echo "====="
echo "SciELO ZA daily export"
echo "====="
echo "Source      : $SOURCE"
echo "Export file : $FILENAME"
echo "Destination : $DEST"
echo

# Prevent duplicate processing
if [ -f "$DEST" ]; then
    echo "File has already been processed:"
    echo "$DEST"
    exit 0
fi

# Create an independent compressed copy.
# The logrotate-managed source file is not changed.
echo "Compressing..."

gzip -c "$SOURCE" > "$TMP"

if [ $? -ne 0 ]; then
    echo "ERROR: compression failed."
    rm -f "$TMP"
    exit 1
fi
```

```
# Validate the gzip stream
if ! gzip -t "$TMP"; then
    echo "ERROR: invalid gzip file."
    rm -f "$TMP"
    exit 1
fi

echo "Compression successful:"
ls -lh "$TMP"

# FTP transfer
echo
echo "Uploading to FTP..."

FTP_LOG=$(mktemp)

ftp -n -v "$HOST" >"$FTP_LOG" 2>&1 <<EOF
quote USER $FTP_USER
quote PASS $FTP_PASS
binary
put $TMP $FILENAME
quit
EOF

cat "$FTP_LOG"

# Detect common FTP failures
if grep -Eqi \
    '530 |550 |425 |426 |not connected|failed|failure|error|refused|timed out|no such|not
found' \
    "$FTP_LOG"; then

    echo "ERROR: FTP reported a failure."
    rm -f "$FTP_LOG"
    rm -f "$TMP"
    exit 1
fi

# Require FTP server completion confirmation
```

```

if ! grep -Eqi '^226 ' "$FTP_LOG"; then
    echo "ERROR: FTP server did not confirm successful transfer."
    rm -f "$FTP_LOG"
    rm -f "$TMP"
    exit 1
fi

rm -f "$FTP_LOG"

echo
echo "FTP transfer completed."

# Archive only after FTP success
mv -f -- "$TMP" "$DEST"

if [ $? -ne 0 ]; then
    echo "ERROR: FTP transfer succeeded, but local archival failed:"
    echo "$DEST"
    exit 1
fi

echo
echo "=====
echo "PROCESS COMPLETED"
echo "=====
echo "Archived file:"
echo "$DEST"

exit 0

```

Set permissions:

```

chmod 750 /usr/local/sbin/export-scielo-za.sh
chown root:root /usr/local/sbin/export-scielo-za.sh

```

## 6. Why the Script Uses `gzip -c`

The command:

```
gzip -c "$SOURCE" > "$TMP"
```

reads the rotated log and writes a compressed copy.

It does **not** replace or delete:

```
/var/log/nginx/scielo-access.log-YYYYMMDD
```

This separation is intentional. The original file remains under `logrotate` lifecycle management.

---

## 7. FTP Transfer Method

The FTP implementation uses the same connection method already validated in the environment:

```
ftp -n -v "$HOST" <<EOF
quote USER $FTP_USER
quote PASS $FTP_PASS
binary
put "$LOCAL_FILE" "$REMOTE_FILE"
quit
EOF
```

`binary` is required for gzip files.

The process does not force passive mode because the existing FTP workflow has already been validated without that change.

## Successful transfer

The FTP server should return a `226` response, for example:

```
226 Transfer complete.
```

The script requires a `226` response before considering the upload successful.

## Common authentication failure

```
331 Please specify the password.
530 Login incorrect.
```

This indicates an authentication problem, not an upload or passive-mode problem.

Check the configured username/password and the credentials file.

---

## 8. Manual Test

Run:

```
/usr/local/sbin/export-scielo-za.sh
```

For a rotation dated October 9, 2026, the expected local file is:

```
/var/log/nginx/scielo-org-za/2026-10/2026-10-09_scielo.za.log.gz
```

Verify it:

```
ls -lh /var/log/nginx/scielo-org-za/2026-10/
```

Test gzip integrity:

```
gzip -t /var/log/nginx/scielo-org-za/2026-10/2026-10-09_scielo.za.log.gz  
echo $?
```

Expected:

```
0
```

Inspect the beginning of the archived log:

```
gzip -dc /var/log/nginx/scielo-org-za/2026-10/2026-10-09_scielo.za.log.gz | head
```

---

## 9. Daily Scheduling

The observed log rotation normally occurs around 03:00--04:00. Run the export after the rotation has completed.

Create:

```
vi /etc/cron.d/scielo-za-export
```

## Contents:

```
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/sbin

0 5 * * * root /usr/local/sbin/export-scielo-za.sh >> /var/log/export-scielo-za.log 2>&1
```

## Set permissions:

```
chmod 644 /etc/cron.d/scielo-za-export
chown root:root /etc/cron.d/scielo-za-export
```

## Verify:

```
cat /etc/cron.d/scielo-za-export
```

## Monitor execution:

```
tail -100 /var/log/export-scielo-za.log
```

---

# 10. Historical Log Migration

Previously compressed logrotate files may look like:

```
scielo-access.log-20260930.gz
scielo-access.log-20261001.gz
scielo-access.log-20261002.gz
```

These files are already compressed and should **not** be recompressed.

They can be uploaded with a new remote filename and then moved/renamed locally.

Example mapping:

```
scielo-access.log-20260930.gz
-> 2026-09-30_scielo.za.log.gz

scielo-access.log-20261001.gz
-> 2026-10-01_scielo.za.log.gz
```

Local destinations:

```
/var/log/nginx/scielo-org-za/2026-09/2026-09-30_scielo.za.log.gz  
/var/log/nginx/scielo-org-za/2026-10/2026-10-01_scielo.za.log.gz
```

## Historical migration script

```
#!/bin/bash

source /root/.sciELO-ftp.conf

HOST="ftp.ratchet.scielo.org"
ARCHIVE="/var/log/nginx/scielo-org-za"

for f in /var/log/nginx/scielo-access.log-??????.gz; do

    [ -f "$f" ] || continue

    base=$(basename "$f")

    data="${base#sciELO-access.log-}"
    data="${data%.gz}"

    year="${data:0:4}"
    month="${data:4:2}"
    day="${data:6:2}"

    FILE_REMOTE="${year}-${month}-${day}_sciELO.za.log.gz"
    DESTDIR="${ARCHIVE}/${year}-${month}"
    DEST="${DESTDIR}/${FILE_REMOTE}"

    echo
    echo "===== "
    echo "Source      : $f"
    echo "FTP filename: $FILE_REMOTE"
    echo "Destination : $DEST"
    echo "===== "

    if ! gzip -t "$f"; then
```

```
        echo "ERROR: invalid gzip file: $f"
        continue
    fi

    mkdir -p "$DESTDIR"

    FTP_LOG=$(mktemp)

    ftp -n -v "$HOST" >"$FTP_LOG" 2>&1 <<EOF
quote USER $FTP_USER
quote PASS $FTP_PASS
binary
put $f $FILE_REMOTE
quit
EOF

    cat "$FTP_LOG"

    if grep -Eqi \
        '530 |550 |425 |426 |not connected|failed|failure|error|refused|timed out|no such|not
found' \
        "$FTP_LOG"; then

        echo "ERROR: FTP transfer failed: $FILE_REMOTE"
        rm -f "$FTP_LOG"
        continue
    fi

    if ! grep -Eqi '^226 ' "$FTP_LOG"; then
        echo "ERROR: FTP server did not confirm transfer: $FILE_REMOTE"
        rm -f "$FTP_LOG"
        continue
    fi

    rm -f "$FTP_LOG"

    mv -f -- "$f" "$DEST"

    if [ $? -ne 0 ]; then
        echo "ERROR: FTP succeeded, but local move failed:"
```

```
        echo "$f -> $DEST"
        continue
    fi

    echo "OK: $FILE_REMOTE"
done
```

The order is intentionally:

```
Validate gzip
    |
    v
Upload to FTP
    |
    v
Receive FTP 226
    |
    v
Move/Rename locally
```

If FTP fails, the original historical file remains in place for a later retry.

---

# 11. Troubleshooting

## Rotated source file does not exist

Example:

```
ERROR: rotated log was not found or is empty:
/var/log/nginx/scielo-access.log-20261009
```

Check:

```
ls -lh /var/log/nginx/scielo-access.log*
```

Also check logrotate status and execution time.

---

# FTP login fails

Typical response:

```
331 Please specify the password.  
530 Login incorrect.
```

Verify that credentials are loaded:

```
source /root/.scielo-ftp.conf  
  
printf 'FTP_USER=[%s]\n' "$FTP_USER"  
printf 'FTP_PASS length=%d\n' "${#FTP_PASS}"
```

Do not print the actual password.

Check for Windows CRLF characters:

```
cat -A /root/.scielo-ftp.conf
```

If lines contain `^M`, normalize them:

```
sed -i 's/\r$//' /root/.scielo-ftp.conf
```

---

## FTP transfer starts but fails

Relevant FTP response codes include:

```
425 Cannot open data connection  
426 Connection closed; transfer aborted  
550 Requested action not taken
```

Check firewall rules, FTP data-channel behavior, available disk space and server-side permissions.

---

## Gzip validation fails

Test manually:

```
gzip -t FILE.gz
echo $?
```

A successful validation returns:

```
0
```

Do not upload or archive a file that fails gzip validation.

---

## File was already processed

The daily script checks:

```
/var/log/nginx/scielo-org-za/YYYY-MM/YYYY-MM-DD_scielo.za.log.gz
```

If the file already exists, the script exits successfully without overwriting or retransmitting it.

---

## 12. Operational Verification

Useful daily checks:

```
tail -100 /var/log/export-scielo-za.log
```

```
find /var/log/nginx/scielo-org-za -type f -name '*_scielo.za.log.gz' -printf '%TY-%Tm-%Td
%TH:%TM %s %p\n' | sort
```

Check gzip integrity for a specific file:

```
gzip -t /var/log/nginx/scielo-org-za/2026-10/2026-10-09_scielo.za.log.gz
```

Check the most recent archive:

```
find /var/log/nginx/scielo-org-za -type f -name '*_scielo.za.log.gz' -printf '%T@ %p\n' \
| sort -nr \
| head
```

---

# 13. Recovery Considerations

The current daily script uses:

```
DATA=$(date +%Y%m%d)
```

This means it expects the logrotate output for the current date to exist when the script runs.

If FTP is unavailable or the scheduled job fails, the next day's run does not automatically retry the previous day's source file.

For a more resilient production implementation, a future version should:

- discover all eligible `scielo-access.log-YYYYMMDD` and `.gz` files;
- determine whether the corresponding `YYYY-MM-DD_scielo.za.log.gz` has already been archived;
- process only missing exports;
- safely retry failed FTP transfers;
- keep a clear success/failure audit trail.

This removes the dependency on a single daily execution window.

---

# 14. Security and Preservation Notes

The export archive and the original Nginx logs serve different operational purposes.

The export process should not alter the original logrotate-managed source. Keeping the source untouched provides a reference copy if an export must be regenerated or investigated.

Recommended controls include:

- keep FTP credentials readable only by root;
  - do not store credentials in source control;
  - validate gzip files before transfer;
  - require an FTP success response before considering the transfer complete;
  - retain execution logs;
  - monitor failed scheduled jobs;
  - periodically verify that local and remote expected files are present;
  - migrate from unencrypted FTP to SFTP or FTPS when supported.
-

# 15. Key Paths

---

Purpose Path

---

Active Nginx access log `/var/log/nginx/scielo-access.log`

Rotated Nginx logs `/var/log/nginx/scielo-access.log-YYYYMMDD[.gz]`

Export archive `/var/log/nginx/scielo-org-za/YYYY-MM/`

Daily export script `/usr/local/sbin/export-scielo-za.sh`

FTP credentials `/root/.sciELO-ftp.conf`

Cron configuration `/etc/cron.d/scielo-za-export`

Export execution log `/var/log/export-scielo-za.log`

FTP host `ftp.ratchet.scielo.org`

---

# 16. Expected Final State

After successful daily processing:

```
/var/log/nginx/  
├─ sciELO-access.log  
├─ sciELO-access.log-20261009  
├─ sciELO-access.log-20261008.gz  
└─  
├─ sciELO-org-za/  
├─ 2026-10/  
│   ├── 2026-10-01_sciELO.za.log.gz  
│   ├── 2026-10-02_sciELO.za.log.gz  
│   └─ ...  
└─ 2026-10-09_sciELO.za.log.gz
```

The FTP server should contain the corresponding exported filenames:

2026-10-01\_scielo.za.log.gz

2026-10-02\_scielo.za.log.gz

...

2026-10-09\_scielo.za.log.gz

---

**Document purpose:** operational runbook for the SciELO Nginx access-log FTP export workflow.

---

Revision #1

Created 9 October 2026 18:10:44 by Rondineli G. Saad

Updated 9 October 2026 18:12:08 by Rondineli G. Saad