

☒ Why the PowerShell “rename + reopen” approach fails in your case

What's happening internally:

1. system.log is **renamed** to system-17-02-2025.log (700 MB)
2. NGINX **reopens** logs
3. NGINX **continues writing the same stream**
4. Result:
5. system-17-02-2025.log (700 MB)
6. system.log (750 MB, includes same content again)

Root cause

- On **Windows**, file handles behave differently than Linux
- Some processes keep **buffered handles open**
- NGINX ends up **re-writing overlapping data**
- This is a known limitation when trying to mimic `logrotate` on Windows

☞ So your conclusion is correct:

Renaming logs on Windows = duplication risk

□ REAL Solutions That Actually Work (No Duplicate Logs)

☒ BEST SOLUTION (No Rotation Needed): Date-based log files (Native NGINX)

Instead of rotating logs, **never write to the same file twice**.

NGINX supports variables in `access_log` paths, even on Windows.

□ Access log with daily file names (NO duplication)

```
log_format main
'$remote_addr - $remote_user [$time_local] '
'"$request" $status $body_bytes_sent '
'"$http_referer" "$http_user_agent"';  
  
access_log logs/access_$year-$month-$day.log main;
```

■ Result:

```
access_2025-02-17.log
access_2025-02-18.log
```

access_2025-02-19.log

- ✓ No rotation
 - ✓ No copy
 - ✓ No reopen
 - ✓ Zero duplication
 - ✓ Fresh file every day automatically
-

□ Important limitation (very important)

Log Type Variable Support

access_log YES
error_log NO

So error_log still needs a different approach.

□ Proper Solution for error.log (No duplication)

✓ Option 1: Truncate error.log (NOT rename)

Instead of renaming (which duplicates), **truncate the file**.

PowerShell (safe approach)

```
$log = "C:\nginx\logs\error.log"

# Backup once
Copy-Item $log "C:\nginx\logs\error-$((Get-Date -Format yyyy-MM-dd)).log"

# Truncate original (zero size)
Clear-Content $log
```

- ✓ Keeps file handle intact
- ✓ No duplicate content
- ✓ Disk usage controlled

This mimics `copytruncate` from Linux

✓ Option 2: Limit error log verbosity (strongly recommended)

```
error_log logs/error.log warn;
```

or even:

```
error_log logs/error.log error;
```

This alone often reduces size by **80–90%**.

⊖ What NOT to do on Windows

- Rename + reopen
 - Restart NGINX daily
 - Copy full logs without truncation
 - Rely on Linux-style logrotate logic
-

□ Best-Practice Architecture for Windows + NGINX

Component	Approach
Access logs	Date-based filenames (no rotation)
Error logs	Truncate + backup
Health checks	<code>access_log off;</code>
Retention	Scheduled deletion
Large scale	Centralized logging

□ Optional: Auto-delete old logs (safe)

```
Get-ChildItem "C:\nginx\logs\access_*.log" |  
Where-Object { $_.LastWriteTime -lt (Get-Date).AddDays(-14) } |  
Remove-Item -Force
```

□ Final Recommendation (Most Stable Setup)

- ✓ `access_log` → date-based files (no rotation)
 - ✓ `error_log` → truncate, don't rename
 - ✓ retention → delete old files
 - ✓ verbosity → reduce log level
-

If you want, I can:

- Validate your **current nginx.conf**
- Provide a **Windows-safe log strategy document**
- Give a **PDCA / audit-ready explanation**

- Create a **single, clean PowerShell script** that ONLY truncates + cleans

Just tell me 

```
log_format main '$remote_addr - $remote_user [$time_local] "'"$request" $status
$body_bytes_sent '"$http_referer" "$http_user_agent"'; access_log logs/access_$year-
$month-$day.log main; where to add
```

Good question — this **must be placed in the correct context**, otherwise NGINX will either fail to start or silently ignore it.

Below is the **exact, correct placement** for Windows + NGINX.

Correct location in `nginx.conf`

1 Open your main config file

```
C:\nginx\conf\nginx.conf
```

2 Add `log_format` inside the `http {}` block only

Do NOT put `log_format` inside `server` or `location`

Correct structure:

```
http {
    log_format main
        '$remote_addr - $remote_user [$time_local] '
        "'"$request" $status $body_bytes_sent '
        '"$http_referer" "$http_user_agent"';

    access_log logs/access_$year-$month-$day.log main;

    include      mime.types;
    default_type application/octet-stream;

    sendfile      on;
    keepalive_timeout 65;

    include conf.d/*.conf;
}
```

Optional: Per-server override (recommended)

If you want this **only for one reverse-proxy server**, move **only the `access_log`** into the `server` block.

nginx.conf

```
http {  
    log_format main  
    '$remote_addr - $remote_user [$time_local] '  
    '"$request" $status $body_bytes_sent '  
    '"$http_referer" "$http_user_agent"';  
  
    include conf.d/*.conf;  
}
```

conf.d/reverse-proxy.conf

```
server {  
    listen 80;  
    server_name example.internal;  
  
    access_log logs/access_$year-$month-$day.log main;  
  
    location / {  
        proxy_pass http://backend;  
    }  
}
```

Q Verify configuration before restart

From **CMD or PowerShell**:

```
cd C:\nginx  
nginx.exe -t
```

Expected output:

```
nginx: the configuration file C:\nginx\conf\nginx.conf syntax is ok  
nginx: configuration file C:\nginx\conf\nginx.conf test is successful
```

↻ Reload (no downtime)

```
nginx.exe -s reload
```

▀ Resulting log files

```
C:\nginx\logs\  
└── access_2026-01-18.log  
└── access_2026-01-19.log  
└── access_2026-01-20.log
```

- ✓ No duplication
- ✓ No rotation

- ✓ No PowerShell renaming
 - ✓ Windows-safe
-

□ Important Notes (Windows-specific)

- \$year-\$month-\$day works **only for access_log**
 - error_log **cannot** use variables
 - NGINX creates new file automatically at midnight
 - Old logs are never re-opened
-

□ Best Practice Add-Ons (Highly Recommended)

Disable access logs for health checks

```
location /health {  
    access_log off;  
    return 200 "OK";  
}
```

Reduce log size

```
access_log logs/access_$year-$month-$day.log main buffer=64k flush=1m;
```

If you want, I can:

- Review your **full nginx.conf**
- Add **error log safe truncation**
- Provide **retention + cleanup strategy**
- Create a **Windows audit-ready explanation**

Just say the word 