

Upload 503 Error - Diagnostic Report

Date: December 13, 2025

Issue: Gallery and Document uploads returning 503 Service Unavailable

Commit: 54cbc40

🔍 Root Cause Analysis

The Problem

Your uploads are failing with **503 Service Unavailable** errors because the Cloudinary environment variables are not being detected by the application.

What the Logs Show

From `/home/ubuntu/Uploads/log4.txt`:

```
[Documents API] Request timed out after 15003ms
[Documents API] Request timed out after 15001ms
```

The uploads aren't actually timing out—they're **failing immediately** with 503 errors because Cloudinary configuration is missing.

Why This Happens

The code checks for these environment variables:

```
CLOUDINARY_CLOUD_NAME
CLOUDINARY_API_KEY
CLOUDINARY_API_SECRET
```

If **any** of these are missing, the upload endpoints return:

- **Status:** 503 Service Unavailable
- **Message:** "File upload is not configured. Please contact your administrator."

✓ What I've Done

1. Added Diagnostic Endpoint ✓

Created `/api/diagnostic/cloudinary` to check environment variables:

- Shows which env vars are set
- Shows their lengths (but not the values for security)
- Lists all Cloudinary-related env vars

How to use it:

```
curl https://carelinkai.onrender.com/api/diagnostic/cloudinary
```

2. Enhanced Error Logging

Updated both upload routes to log:

- Which environment variables are set/missing
- Cloudinary configuration status
- Detailed error information in 503 responses

Files modified:

- `src/app/api/family/gallery/upload/route.ts`
- `src/app/api/family/documents/route.ts`

3. Better Error Messages

503 errors now include:

```
{
  "error": "File upload service not configured",
  "code": "UPLOAD_SERVICE_NOT_CONFIGURED",
  "details": {
    "cloudName": false,
    "apiKey": false,
    "apiSecret": false
  }
}
```



How to Fix This Issue

Step 1: Verify Environment Variables on Render

1. Go to your Render dashboard: <https://dashboard.render.com>
2. Select your **carelinkai** service
3. Go to **Environment** tab
4. Check that you have **exactly** these variable names:
 - `CLOUDINARY_CLOUD_NAME`
 - `CLOUDINARY_API_KEY`
 - `CLOUDINARY_API_SECRET`



IMPORTANT: Variable names are **case-sensitive** and must match exactly.

Step 2: Check for Common Issues

Issue A: Wrong Variable Names



Wrong:

- `CLOUDINARY_CLOUD` (missing `_NAME`)
- `CLOUDINARY_KEY` (missing `API_`)
- `CloudName` (wrong case)



Correct:

- `CLOUDINARY_CLOUD_NAME`
- `CLOUDINARY_API_KEY`
- `CLOUDINARY_API_SECRET`

Issue B: Extra Spaces

Environment variable values should not have leading/trailing spaces:

- X Wrong: dygtSnu8z (has spaces)
- ✓ Correct: dygtSnu8z (no spaces)

Issue C: Missing Values

From your screenshot, you should have:

- **Cloud name:** dygtSnu8z
- **API key:** 328392542172231
- **API secret:** KhpohAEF0sjVKuxRENaBhCoIYFQ

Step 3: Deploy Changes

After verifying environment variables:

1. Manual Deploy (fastest):

```
bash
# In Render dashboard
Click "Manual Deploy" → "Deploy latest commit"
```

2. OR Auto-Deploy (if enabled):

- Render will automatically redeploy when you push the diagnostic code

Step 4: Test with Diagnostic Endpoint

Once deployed, visit (while logged in):

```
https://carelinkai.onrender.com/api/diagnostic/cloudinary
```

You should see:

```
{
  "isConfigured": true,
  "environmentVariables": {
    "CLOUDINARY_CLOUD_NAME": {
      "exists": true,
      "value": "***SET***",
      "length": 9
    },
    "CLOUDINARY_API_KEY": {
      "exists": true,
      "value": "***SET***",
      "length": 15
    },
    "CLOUDINARY_API_SECRET": {
      "exists": true,
      "value": "***SET***",
      "length": 24
    }
  }
}
```

Step 5: Check Render Logs

After deploying, watch the logs when trying to upload:

```
# You should see:
[Gallery Upload] Checking Cloudinary configuration: {
  CLOUDINARY_CLOUD_NAME: '****SET***',
  CLOUDINARY_API_KEY: '****SET***',
  CLOUDINARY_API_SECRET: '****SET***',
  isConfigured: true
}
```

If you see `NOT SET` for any variable, that's the problem!



Expected vs Actual

Expected Configuration

```
CLOUDINARY_CLOUD_NAME=dygtSnu8z
CLOUDINARY_API_KEY=328392542172231
CLOUDINARY_API_SECRET=KhpoAEF0sjVKuxRENaBhCoIYFQ
```

What to Check

1. Variable names match exactly (case-sensitive)
 2. No extra spaces in values
 3. All three variables are set
 4. Render has redeployed after adding variables
-



Next Steps

Immediate Actions:

1. **Push diagnostic code to GitHub**

`bash`

```
git push origin main
```

(Note: You'll need to authenticate - see `GITHUB_PUSH_INSTRUCTIONS.md`)

2. **Verify environment variables in Render dashboard**

- Check exact names
- Check for typos
- Check for extra spaces

3. **Redeploy on Render**

- Manual deploy or wait for auto-deploy

4. **Test diagnostic endpoint**

```
https://carelinkai.onrender.com/api/diagnostic/cloudinary
```

5. **Try uploading again**

- Gallery upload
- Document upload

If Still Failing:

1. Check Render logs for the new diagnostic output
 2. Screenshot the diagnostic endpoint response
 3. Screenshot the Render environment variables page
 4. Share these with me for further diagnosis
-



Files Changed

New Files:

- `src/app/api/diagnostic/cloudinary/route.ts` - Diagnostic endpoint

Modified Files:

- `src/app/api/family/gallery/upload/route.ts` - Enhanced logging
- `src/app/api/family/documents/route.ts` - Enhanced logging

Commit:

54cbc40 - fix: Add diagnostic logging **for** Cloudinary upload 503 errors



Security Note

The diagnostic endpoint:

- Requires authentication (login required)
- Never exposes actual credentials (shows `***SET***`)
- Only shows if variables exist and their lengths
- Should be **removed or restricted** after diagnosis



Support Checklist

When reporting issues, provide:

- [] Screenshot of Render environment variables page
- [] Response from `/api/diagnostic/cloudinary` endpoint
- [] Render logs showing upload attempt
- [] Network tab showing 503 error details
- [] Browser console errors



Expected Outcome

After fixing environment variables:

- Gallery uploads work
- Document uploads work
- No more 503 errors

- Files uploaded to Cloudinary successfully
 - Diagnostic endpoint shows `isConfigured: true`
-

Status: Diagnostic code committed (54cbc40)

Next: Push to GitHub → Verify Render env vars → Redeploy → Test