

# File Storage Alternatives for ScripTrack

---

## Current Status

---

- **FIXED:** Enhanced local storage now using `persistent-uploads` directory
- **RESTORED:** All 8 backed up files restored to persistent storage
- **SECURE:** Aggressive cleanup script modified to prevent accidental deletion

## Alternative Cloud Storage Solutions

---

### Option 1: Google Cloud Storage (Free Tier Available)

```
npm install @google-cloud/storage
```

- 5GB free storage
- Simple setup with service account key
- Similar API to AWS S3

### Option 2: Cloudinary (Media-focused)

```
npm install cloudinary
```

- Free tier: 25GB storage, 25GB bandwidth
- Excellent for images with automatic optimization
- Built-in transformations and CDN

### Option 3: Supabase Storage

```
npm install @supabase/storage-js
```

- Free tier: 1GB storage
- PostgreSQL-based (matches your current DB)
- Built-in authentication integration

### Option 4: Firebase Storage

```
npm install firebase
```

- Free tier: 5GB storage, 1GB/day transfer
- Google-backed reliability
- Real-time capabilities

## Option 5: Azure Blob Storage

```
npm install @azure/storage-blob
```

- Free tier available
- Enterprise-grade reliability
- Global CDN available

## Option 6: DigitalOcean Spaces

```
npm install aws-sdk # Uses S3-compatible API
```

- \$5/month for 250GB storage
- S3-compatible API (easy migration)
- Built-in CDN

## Implementation Examples

### Google Cloud Storage Implementation

```
import { Storage } from '@google-cloud/storage';

const storage = new Storage({
  projectId: 'your-project-id',
  keyFilename: 'path/to/service-account-key.json',
});

const bucket = storage.bucket('your-bucket-name');

// Upload file
const file = bucket.file(fileName);
await file.save(buffer, {
  metadata: { contentType }
});
```

### Clouinary Implementation

```
import { v2 as clouinary } from 'clouinary';

clouinary.config({
  cloud_name: 'your-cloud-name',
  api_key: 'your-api-key',
  api_secret: 'your-api-secret',
});

// Upload file
const result = await clouinary.uploader.upload_stream(
  { resource_type: 'auto' },
  (error, result) => {
    if (result) {
      console.log('Upload successful:', result.secure_url);
    }
  }
).end(buffer);
```


## Recommendations

---

### For Your Use Case:

1. **Cloudinary** - Best for images and mixed media
2. **Google Cloud Storage** - Most similar to S3, reliable
3. **DigitalOcean Spaces** - Cost-effective, S3-compatible

### Quick Setup Priority:

1. **Enhanced Local Storage** (Already implemented) 
2. **Cloudinary** (Easy setup, great for studios)
3. **Google Cloud Storage** (Enterprise-grade)

Would you like me to implement any of these alternatives?