Threads Media Downloader - Deployment Guide

1■■ Deployment Options

- Kubernetes: AWS EKS / GCP GKE (for scalable, containerized deployments)
- Serverless: AWS Lambda / Google Cloud Functions (for event-driven execution)
- Traditional VM / Docker-Based: Standalone deployment on any server

2■■ Set Up Custom Domain & HTTPS

- Configure AWS Route 53 or Google Cloud DNS
- Enforce HTTPS using Let's Encrypt (Kubernetes) or AWS ACM (Cloud)
- Secure API endpoints with TLS encryption

3■■ Monitoring & Performance Optimization

- Use Prometheus & Grafana for real-time monitoring
- Enable auto-scaling with Kubernetes HPA (Horizontal Pod Autoscaler)
- Log and analyze security events with ELK Stack (Elasticsearch, Logstash, Kibana)

4■■ Security & Compliance

- Enforce OAuth 2.0 authentication with Google IAP
- Protect against DDoS and SQL Injection with AWS WAF & Cloudflare
- Implement IAM roles and least-privilege access policies

5■■ AI Features

- NSFW detection with AI-based tagging (Marks but does not block downloads)
- Automatic caption generation for images & videos
- Optical Character Recognition (OCR) to extract text from images
- Sentiment analysis & keyword extraction for media files

6■■ CI/CD Pipeline for Automated Deployment

- GitHub Actions for automated builds & deployments
- Kubernetes Helm Charts for seamless updates
- AWS Lambda / GCP Functions for serverless auto-deployment

7■■ Final Steps Before Production

- Test end-to-end functionality & performance
- Set up incident response with PagerDuty & Slack alerts
- Optimize cost & scaling based on real-world usage

■ Deployment Complete!

Your Threads Media Downloader is now fully operational with AI-powered features, secure authentication, automated deployment, and cloud-scale performance. ■