Project Title: CloudFolks Watermarking Platform

Objective: Develop a platform that allows users to upload images and have the"CloudFolks" watermark automatically added to their content.

Core Features:

User Authentication:

● Service: AWS Cognito.

● Description: Enable user registration and sign-in. Ensure secure access

to the platform.

Image Upload and Storage:

● Service: API Gateway, EC2, and S3.

● Description: Users can upload images via an API. Original images are

stored in an 'incoming' S3 bucket.

Watermarking:

● Service: EC2.

● Description: As images are uploaded, the EC2 instance automatically

overlays the "CloudFolks" text as a watermark on the image. Users can

specify settings like opacity, size, and position of the watermark.

Notifications:

● Service: SNS.

● Description: Notify users when their watermarked image is ready for

download.

Image Download and Access:

● Service: API Gateway, EC2, and S3.

● Description: Users can access and download their watermarked

images from a 'processed' S3 bucket.

Frontend Platform:

● Service: S3 Static Website Hosting.

● Description: A user-friendly frontend built using HTML, CSS, and

JavaScript, hosted as a static website on S3. This serves as the

primary interface for user interaction.

Workflow:

Users sign up or log in via the Cognito-powered authentication system.

They navigate the S3-hosted frontend to upload their images.

Once uploaded, users can specify their watermark preferences for

"CloudFolks", such as its opacity, size, and position.

The EC2 instances automatically handle watermarking tasks.

After watermarking, images are moved to the 'processed' S3 bucket.

Users are notified via SNS when their images are ready.

They can then download or view their watermarked images through the

platform.

Considerations & Enhancements:

● Scalability: Implement EC2 Auto Scaling to ensure efficient watermarking

during high upload times.

● Cost-Optimization: Consider EC2 Spot Instances and S3 lifecycle policies to

manage costs.

● Security: Implement data encryption in-transit and at-rest. Define strict IAM

roles and policies.

● User Experience: Allow users to see a real-time watermark preview as they

adjust settings. Implement a drag-and-drop UI for watermark position

selection.

● Monitoring: Integrate AWS CloudWatch for system health, performance

metrics, and logging.

Flow control :

1. User Interaction with S3-hosted Website: The user visits the static website hosted on S3. On this site, there's an option to upload an image.

2. Uploading Image through API Gateway: When the user selects an image to upload, your frontend JavaScript code makes an HTTP request (e.g., a POST request) directly to an API Gateway endpoint, not to another S3 bucket. This request contains the image file the user wants to watermark.

3. API Gateway to EC2 or Lambda: The API Gateway endpoint is configured to forward this request to a backend service, which can be an AWS Lambda function, an EC2 instance, etc. This backend service handles the watermarking process.

4. Storing the Watermarked Image: Once the image is processed by the backend, it's then stored in an S3 bucket (which can be the same bucket where the website is hosted or a different one).

5. Returning the Result to the User: Optionally, after processing, the URL of the

watermarked image in the S3 bucket can be returned to the user through the same API Gateway endpoint, and the frontend can then display it.

So, in essence:

1. The user interacts with the S3-hosted website.

2. The image they want to watermark is sent directly to an API Gateway endpoint (not

stored in S3 at this point).

3. API Gateway forwards the request to EC2 (or Lambda) for processing.

4. The processed image is saved in S3.

5. The link or status is optionally returned to the user.