

## AWS S3 + DynamoDB Photo Storage Architecture: Technical Setup Notes

### 1. IAM User Setup

- Create IAM user dedicated for backend (photos-backend-user).
- Assign policies: AmazonS3FullAccess and AmazonDynamoDBFullAccess or equivalent custom-scoped permissions.
- Generate Access Key ID and Secret Access Key.
- Store credentials securely in backend environment variables.

### 2. S3 Bucket Configuration

- Create bucket: cloudery-photos in ap-south-1.
- Disable "Block All Public Access" to support presigned POST public-upload workflow.
- Use folder structure: uploads/.

### 3. S3 CORS Configuration

Use the following CORS settings:

```
[  
{  
  "AllowedHeaders": ["*"],  
  "AllowedMethods": ["GET", "POST", "PUT"],  
  "AllowedOrigins": ["*"],  
  "ExposeHeaders": []  
}  
]
```

### 4. S3 Bucket Policy

Ensure the bucket policy includes both presigned upload and public read for uploaded objects:

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "AllowUploadViaPresignedURL",  
      "Effect": "Allow",
```

```

"Principal": "*",
"Action": "s3:PutObject",
"Resource": "arn:aws:s3:::cloudery-photos/*"
},
{
  "Sid": "AllowPublicRead",
  "Effect": "Allow",
  "Principal": "*",
  "Action": "s3:GetObject",
  "Resource": "arn:aws:s3:::cloudery-photos/uploads/*"
}
]
}

```

## 5. S3 Endpoint Requirements

- Always use the regional endpoint for presigned POST operations:  
<https://cloudery-photos.s3.ap-south-1.amazonaws.com>
- Do not use the global endpoint (s3.amazonaws.com), which will break CORS preflight.

## 6. Backend Presigned POST Generation

- Generate presigned POST with Fields and Conditions including Content-Type.
- Return url and fields to the frontend.
- Ensure boto3 client is configured with s3 virtual addressing style.

## 7. DynamoDB Table Schema

- Table name: photos
- Partition key (HASH): userId (String)
- Sort key (RANGE): uploadedAt (Number)
- Additional attributes: imageId, s3Key, mimeType.
- Store metadata during upload request to backend.

## 8. DynamoDB Access Patterns

- Query photos by userId.

- Use ScanIndexForward=False to return newest uploads first.
- Avoid full table scans; rely on key-based queries.

#### 9. Frontend Architecture Notes

- Call backend for presigned POST data.
- Upload using POST multipart/form-data directly to S3.
- After upload completes, call backend /photos to retrieve metadata.
- Render images using S3 public URL generation:

`https://cloudery-photos.s3.ap-south-1.amazonaws.com/{s3Key}`

#### 10. Security Considerations

- IAM user keys must never be exposed to the client.
- Only presigned URLs should be provided to the frontend.
- Public read is restricted to uploads/ path only.
- Consider migration to CloudFront and signed URLs for production environments.