**Devops Day2day Project**

**File Uploading Continuously *in***

**AWS s3 bucket Using Node.JS**

**Project roadmap:**

1. **Create an S3 Bucket**
2. **Make the Bucket Publicly Accessible**
3. **Create an IAM User**
4. **Create an Access Key for the IAM User**
5. **Create and Deploy the Node.js AWS S3 FileUpload Application on EC2**
6. **Upload Files and View Results**

**In Detail:**

**1. Create an S3 Bucket**

1. Go to the **S3 Console**.
2. Click **Create Bucket**.
3. Provide a **unique bucket name** (e.g., my-bucket-kent).
4. Select a **Region** (e.g., us-east-1).
5. Under **Object Ownership**, select **ACL enabled**.
6. Uncheck **Block all public access** and acknowledge that you are aware of the security risks by ticking the checkbox.
7. Click **Create bucket**.

**2. Make the Bucket Publicly Accessible**

1. Navigate to the **S3 Bucket** you just created.
2. Go to the **Permissions** tab.
3. Click **Bucket Policy** and then **Edit**.
4. Add the following **Bucket Policy** (remember to replace my-bucket-kent with your actual bucket name):

{

"Version": "2012-10-17",

"Statement": [

{

"Sid": "AllowGetObjectForAll",

"Effect": "Allow",

"Principal": "\*",

"Action": "s3:GetObject",

"Resource": "arn:aws:s3:::<**your-bucket-name**>/files\_from\_node/\*"

}

]

}

1. Save the changes. After this, you should see the "Publicly Accessible" badge under the bucket name.

**3. Create an IAM User**

1. Go to the **IAM Console**.
2. Click **Users** > **Add User**.
3. Provide the username s3-photouploader-user.
4. Set **Permissions**:
   * Attach the following policies:
     + **AmazonEC2FullAccess**
     + **AmazonS3FullAccess**
5. Click **Next: Review**, then **Create User**.
6. Note the **Access Key ID** and **Secret Access Key** under the **Security Credentials** tab (you will need them later).

**4. Create an Access Key for the IAM User**

1. Go to **IAM** > **Users** > Select **s3-photouploader-user**.
2. Click on **Security Credentials**.
3. Under **Access keys**, click **Create Access Key**.
4. Choose **CLI** as the type and click **Next**.
5. Save the **Access Key ID** and **Secret Access Key** for later use.

**5. Create and Deploy the Node.js AWS S3 FileUpload Application on EC2**

1. Launch an **EC2 instance** with the following configuration:
   * Name: photo-uploader
   * OS: **Ubuntu**
   * Key Pair: Choose or create a new key pair.
   * Launch the instance.
2. After the EC2 instance is up, connect to it:
   * SSH into the EC2 instance using the key pair.
   * Update the package manager:

**sudo apt update**

* + Install **nvm** (Node Version Manager):

**curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.3/install.sh | bash**

* + - After installation, run the following command to make nvm available:

**export NVM\_DIR="$HOME/.nvm"**

**[ -s "$NVM\_DIR/nvm.sh" ] && \. "$NVM\_DIR/nvm.sh"**

* + Install **Node.js** (version 16):

**nvm install 16**

1. Clone the file upload repository and navigate into the project directory:

**git clone** <https://github.com/azizulmaqsud/Node-s3-FileUploader.git>

**ls -a**

1. Copy the sample .env file to create a new .env file:

**cp .env.sample .env**

1. Edit the .env file to configure it with your bucket name, IAM user credentials, and AWS region:

**vim .env**

Replace the following placeholders in the .env file:

* + AWS\_BUCKET\_NAME: Your S3 bucket name
  + AWS\_ACCESS\_KEY\_ID: Your IAM user's access key.
  + AWS\_SECRET\_ACCESS\_KEY: Your IAM user's secret access key.
  + AWS\_REGION: The AWS region of your S3 bucket (e.g., us-east-1).

1. Install the required Node.js dependencies:

**npm install**

1. Start the application:

**npm start**

1. The application will run by default on port **7000**. You need to allow traffic on this port:
   * Go to the **EC2 dashboard**, select your **EC2 instance**, and navigate to the **Security** tab.
   * In the **Security Group**, add an **Inbound Rule** to allow TCP traffic on port **7000** from anywhere (or from your local IP).

**6. Upload Files and View Results**

1. Open a browser and navigate to:

**http://<your-ec2-public-ip>:7000**

* + Replace <ec2-public-ip> with your EC2 instance's public IP.

1. Use the **Choose Files** button to upload an image or file.
2. After the upload is successful, you should see a **success message**.
3. Go to your **S3 bucket** and reload the files. You should see the uploaded file there:
   * Click on the file to verify that it has been uploaded correctly.
   * Copy the **Object URL** and **paste** it into your browser to view the file.

**Conclusion**

You have successfully deployed an AWS S3 file uploader application using EC2, IAM, and S3. The application allows users to upload files to your S3 bucket, and the uploaded files can be accessed publicly via a URL.