Step 1: Create an S3 Bucket (Static Website Storage)

1. Go to **AWS Console → S3 → Create Bucket**.
2. Name it, e.g., `my-static-website-bucket`.
3. Region: choose your preferred region.
4. **Uncheck Block All Public Access** if this is a public website.
5. Click **Create**.
6. Enable **Static Website Hosting**:
    - Properties → Static website hosting → Enable.
    - Enter **index.html** and **error.html**.
7. Upload your website files (HTML/CSS/JS) to this bucket.

Step 2: Create a VPC and Subnet (Custom Network for EC2)

1. Go to **VPC → Create VPC**.
2. Name it, e.g., `MyVPC`.
3. IPv4 CIDR: `10.0.0.0/16`.
4. Click **Create VPC**.
5. Create a Subnet for EC2:
    - Subnets → Create subnet → Select `MyVPC`.
    - CIDR block: `10.0.1.0/24`.
    - Name it `EC2-Subnet`.

Step 3: Launch an EC2 Instance (Web Server + Log Storage)

1. **EC2 → Launch Instance**.
2. Select **Amazon Linux 2 AMI**.
3. Choose instance type: `t2.micro` (for testing).
4. Network settings:
    - VPC: `MyVPC`
    - Subnet: `EC2-Subnet`
    - Auto-assign Public IP: Yes (if you want public access)
5. Add **Storage** (EBS):
    - Root volume: default 8 GB (or larger if logs will be heavy)
    - Add another volume if needed for persistent log storage
6. Add **Tags** (optional).
7. Configure **Security Group**:
    - Allow SSH (port 22) from your IP.

- Allow HTTP (port 80) from anywhere.

Step 4: IAM Role for EC2 (Access S3)

1. Go to **IAM → Roles → Create Role → EC2**.
2. Attach **AmazonS3FullAccess** (or custom policy for your bucket only).
3. Name the role, e.g., `EC2-S3-Access`.
4. Attach this role to your EC2 instance (during launch or after).

Step 5: Configure EC2 User Data (Sync Files from S3)

1. During EC2 launch → Advanced → User Data.
2. Add a script to install Apache/Nginx and sync S3 bucket:

3. This script ensures the EC2 serves S3 files and can run daily using a **cron job**:

Step 6: Lambda for Monitoring S3 Uploads and Alerts

1. Go to **Lambda → Create Function → Author from scratch**.
2. Runtime: Python 3.x or Node.js.
3. Create an **SNS Topic** for alerts (AWS → SNS → Create topic).
4. Lambda function example (Python) to notify on new file:

5. Configure **S3 bucket → Properties → Event Notifications → Lambda**.
    - Trigger on `PUT` (file upload).

Step 7: Verify the Setup

1. Open EC2 public IP in browser → you should see the website.
2. Upload a new file to S3 → Lambda sends SNS alert.
3. Check EC2 `/var/www/html` → confirm files are synced.