# Implementing a Sample Application on AWS

**Scenario**:  
You want to build a web application where users can upload photos, which are stored in S3, and where these uploads are logged in a database on RDS.

**Step-by-step Guide:**

**1. Set Up the VPC:**

1. **Create a VPC**:
   * Go to the VPC Dashboard.
   * Click 'Your VPCs' > 'Create VPC'.
   * Provide a name and specify an IP CIDR block (e.g., **10.0.0.0/16**).
2. **Create Subnets**:
   * Create at least two subnets for high availability. Remember, these subnets should be in different Availability Zones (AZs).
3. **Internet Gateway**:
   * Create and attach an Internet Gateway to your VPC so your resources can communicate with the internet.

**2. Set Up RDS (Relational Database Service):**

1. **Create a DB Subnet Group**: This tells RDS in which subnets the database should be launched.
2. **Launch a Database Instance**:
   * Go to the RDS Dashboard.
   * Click 'Create database' and select a database engine (e.g., MySQL).
   * Configure the DB instance: size, instance type, master username/password.
   * Choose the VPC and DB Subnet Group created earlier.
   * Enable 'Public Accessibility' for ease of access (Note: Not recommended for production).

**3. Set Up S3 Bucket:**

1. **Create an S3 Bucket**:
   * Go to the S3 Dashboard.
   * Click 'Create bucket'.
   * Provide a unique bucket name.
   * Choose the nearest region.
   * Follow the wizard and leave default settings or modify as per your requirements.

**4. Set Up EC2 (Elastic Compute Cloud):**

1. **Launch an EC2 instance**:
   * Go to the EC2 Dashboard.
   * Click 'Launch Instance'.
   * Choose an AMI (e.g., Amazon Linux 2 AMI).
   * Choose an instance type (e.g., **t2.micro** for free tier).
   * Configure instance details: select the VPC and subnet.
   * Add storage if the default isn’t sufficient.
   * Configure a Security Group to allow traffic: Allow HTTP (port 80) and SSH (port 22) at a minimum.
   * Launch the instance with a key pair. Download and save this key pair securely.
2. **Install a Web Server and Application**:
   * SSH into the EC2 instance.
   * Install a web server (e.g., Apache or Nginx).
   * Write a simple HTML/PHP/Python application that:
     + Lets users upload photos.
     + Stores those photos in the S3 bucket.
     + Logs the upload details (e.g., file name, upload time) in the RDS database.
3. **Dependencies**: Ensure your EC2 instance has the necessary SDKs or libraries to interact with AWS services (like Boto3 for Python).

**Connect Everything:**

1. **S3 Bucket Policy**: Allow your EC2 instance to put objects into your S3 bucket. This requires adding permissions via IAM roles or directly through bucket policies.
2. **Database Connection**: In your web application, set up a connection string to your RDS instance using the endpoint provided in the RDS dashboard.

**Note**: This is a simplified setup and primarily for educational purposes. For real-world applications, consider:

* Using Elastic Load Balancing (ELB) in front of EC2 for high availability.
* Enabling HTTPS for secure communications.
* Tightening security around the RDS database.
* Using IAM roles and policies for fine-grained security control.
* Regularly backing up your RDS databases.
* Monitoring and logging via AWS CloudWatch.

Always refer to the AWS documentation for in-depth details and best practices.