Project Closure Document

Resize an Image in AWS S3 Using a Lambda Function

User Manual

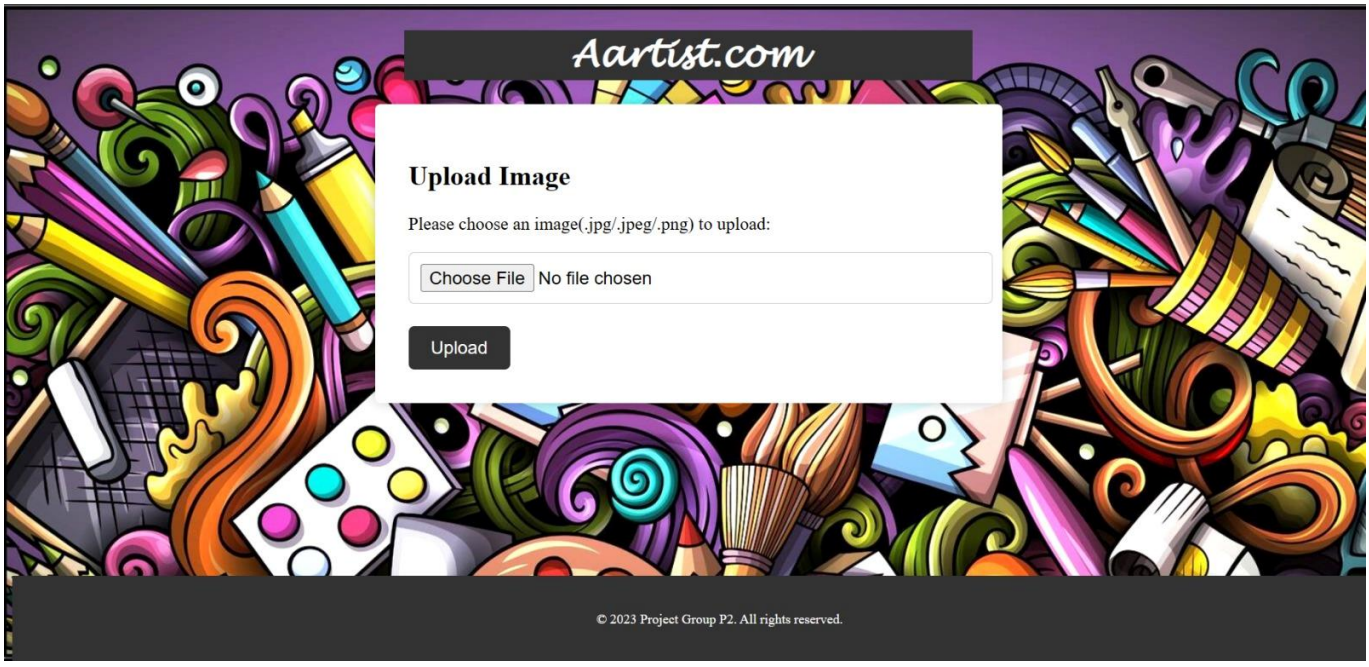
This section will give you the minimum system requirements required to access the webpage.

Minimum System requirements

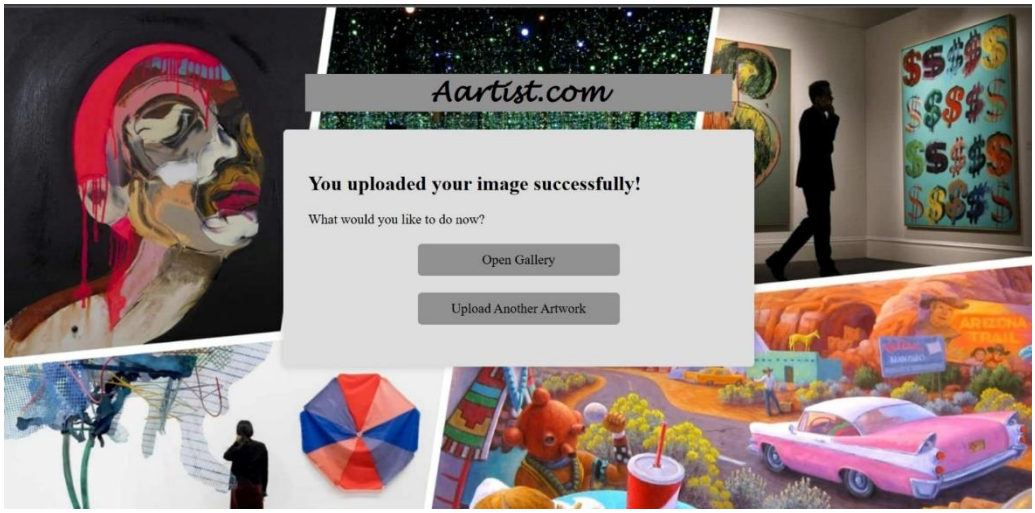
- Windows 8 or above
- Microsoft Edge browser or Google Chrome or Mozilla Firefox
- Memory (RAM): at least 2Gb available

Procedure:

1. Open the webpage using the URL
    a. This will take you to the following webpage

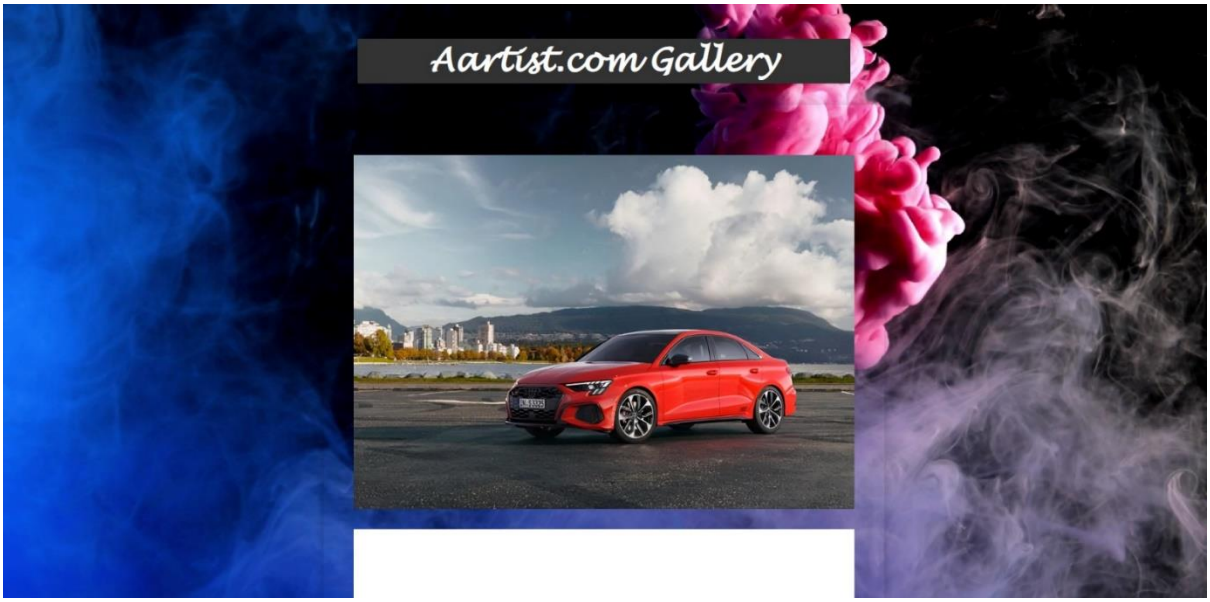


2. Click on choose file and click the upload button to upload your image
    a. Make sure that the image is of .png, .jpeg or .jpg format only
    b. This will take you to the following page



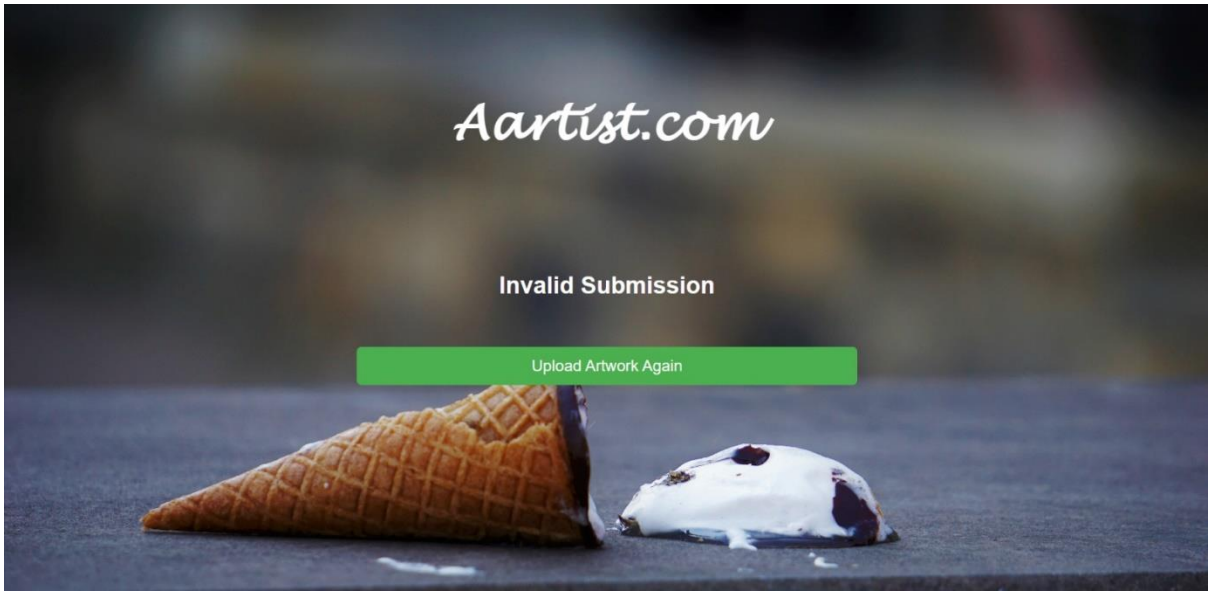
3. To upload another image, click on “Upload Another Artwork” and take you back to the previous uploading page.

4. To check if your images are uploaded and to see others. This will take you to the gallery webpage.



Trouble Shooting:

1. If you upload an image of different format, it will take you to a try again page shown below



In this case click on “Upload Artwork Again” and upload a image of the format of .png, .jpg or .jpeg

2. Even if you submit no image you will land on the same page

Redressal Contact:

For any grievances, please feel free to contact at

Customer Care Service Helpline: 1215481348

E-mail ID: aartist_ccsh@gmmail.com

Client End

Problem Statement and work flow:

You need to resize an image stored in an S3 bucket whenever a new object is created and save the resized image in another S3 bucket using AWS Lambda.

Solution:

Here is a high-level overview of the solution:

- Create two S3 buckets, one for storing the original images and another for storing the resized images.
- Create a Lambda function that is triggered whenever a new object is created in the original image S3 bucket.
- The Lambda function downloads the original image, resizes it, and uploads the resized image to the resized image S3 bucket.

Here are the detailed steps to implement this solution:

1. Create two S3 buckets:
   - One bucket for storing the original images, e.g., "original-images-bucket"
   - Another bucket for storing the resized images, e.g., "resized-images-bucket"
   - Upload an original image to the "original-images-bucket" to trigger the Lambda function during testing.

2. Create a new Lambda function:

   - Go to the AWS Lambda console, click "Create function", and select "Author from scratch".
   - Give the function a name, e.g., "image-resizer-lambda".
   - Select the Python 3.8 runtime.
   - Expand "Choose or create an execution role" and select "Use an existing role".
   - Choose the existing IAM role that has permission to read from and write to the two S3 buckets.
   - Click "Create function".

3. Add an S3 trigger to the Lambda function:

   - In the Designer section of the Lambda function console, click "Add trigger".
   - Select "S3" as the trigger type.
   - Choose the "original-images-bucket" as the bucket.
   - Select "All object create events" as the event type.
   - Leave the other settings as default and click "Add".

4. Write the Lambda function code:

    - In the Function code section of the Lambda function console, replace the default code with the code from the previous answer.
    - Save the code.

5. Test the Lambda function:

    - Upload an original image to the "original-images-bucket".
    - Verify that the Lambda function is triggered and the resized image is saved to the "resized-images-bucket".
    - Check the "CloudWatch" logs for the Lambda function to troubleshoot any issues.

Resources

- Code documentation:
  Below is the link for reference-
    - https://hiteshdiwate.atlassian.net/jira/software/projects/AAR/boards/3/backlog
    - https://github.com/CSD-project/webpage

- Platforms and Tools: Based on thorough research, we have identified and selected the most appropriate software and tools that align with the client's requirements while staying within their specified budget. Our team conducted a comprehensive evaluation of various options and analysed their features, capabilities, and pricing to determine the best fit for the project. By utilizing our expertise and experience in the field, we have identified the optimal software and tools that will enable us to deliver a high-quality solution that meets the client's needs while also being cost-effective.

    - Storage: - AWS S3
    - Resizing of Image: - AWS LAMBDA
    - User Interface: - Python(Flask), HTML, CSS
    -

- Work force: - Our workforce includes
    - Scrum Master
    - Product Owner
    - Frontend Team
    - Backend Team
    - Testing and Integration Team

- Key members (contacts)
    - Levin Dsouza–xxxxxxxx
    - Hitest Diwate – xxxxxxxx

Support Team

Trouble-shooting:

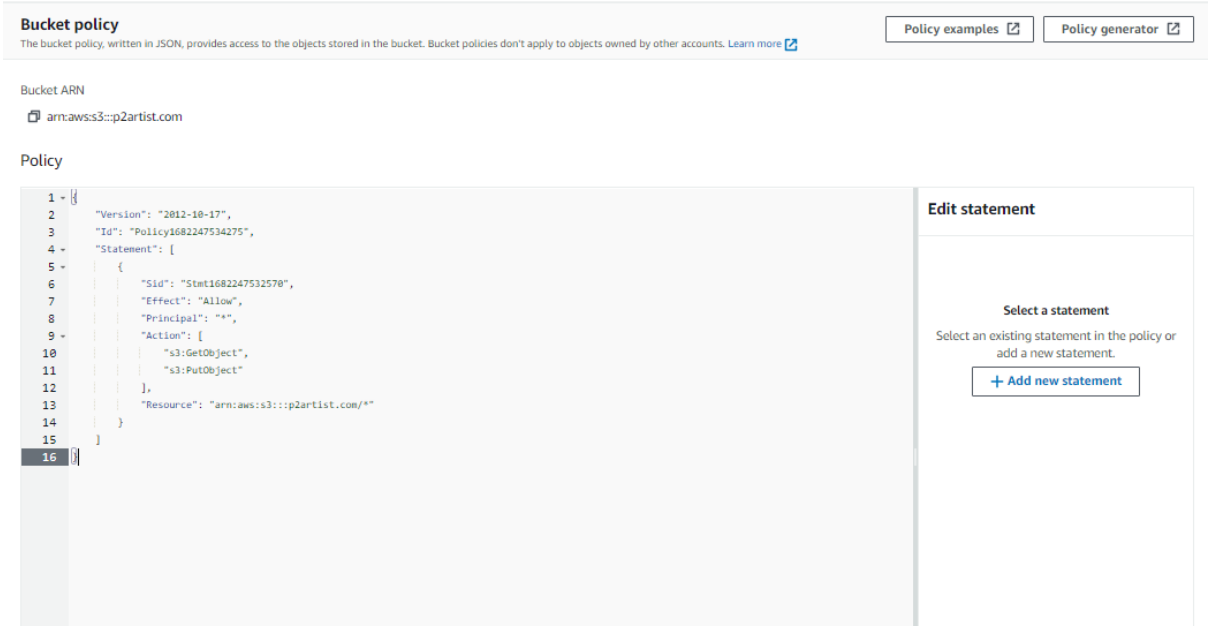
1) Merchant uploaded image but it is not showing in gallery.

   Solution-Refresh the web-page

2) Merchant uploaded the image but it is not showing in s3 bucket which stores original image or it is not getting resized.

   Solution –

   a) check bucket policy, it should have Get-object, Put-object bucket policy and should be applicable to all images in bucket. Refer below picture.



   b) Lambda function and S3 bucket should be created in same region.

3) If resized Bucket is get Destroyed by any reason
   Solution- refer the backup bucket for retrieving data which has been lost.

Security and Recommendations

Security and recommendations are critical components when designing a solution. Here are some security considerations and recommendations for the problem statement of resizing images in S3 using AWS Lambda:

- Authentication and Authorization:
  Ensure that all users and systems interacting with S3 buckets and Lambda function are authenticated and authorized to access the resources. Implement proper security policies and access controls such as IAM roles, security groups, and bucket policies to restrict access and prevent unauthorized users from accessing the resources.

- Encryption:
  Enable server-side encryption on the S3 buckets to protect data at rest. Additionally, configure the Lambda function to encrypt data in transit using SSL/TLS.

- Network Security:
  Implement network security measures such as VPCs, private subnets, and security groups to control inbound and outbound traffic to the Lambda function and S3 buckets. Limit access to specific IP ranges and ports and configure proper routing to ensure secure communication.

- Monitoring:
  Implement monitoring and logging using AWS CloudWatch to track the Lambda function and S3 bucket activity. Monitor access logs, error logs, and performance metrics to detect any anomalies and take appropriate action.

- Version Control:
  Version control is essential when deploying a Lambda function to ensure that code changes are tracked and revertible if necessary. Use a source code version control tool such as Git and deploy the function using a CI/CD pipeline that automates testing, building, and deployment.

- Error Handling:
  Implement proper error handling in the Lambda function to prevent unauthorized access and prevent unexpected behaviour that can result in data loss or security breaches.

- Regular Updates:
  Regularly update the software and tools used in the solution to ensure that security vulnerabilities and bugs are patched. Follow AWS security best practices and guidelines to ensure that the solution is up-to-date and secure.

Outcomes:

Implementing the security considerations and recommendations will help to ensure the following outcomes:

- Improve security
- Compliance
- Reliability
- Scalability

Overall, implementing these security considerations and recommendations can help you build a secure, reliable, and scalable solution that meets your requirements and protects your data.

Recommendations for Future:

To maintain the security of the system, the following recommendations are made:

1. Regular security audits: Conduct regular security audits to identify new risks and vulnerabilities.
2. On-going security training: Provide ongoing security training for employees to ensure that they are aware of security risks and how to avoid them.
3. Keep software up-to-date: Ensure that all software used in the system is up-to-date to prevent vulnerabilities caused by outdated software.

Decommissioning of a product

Standard Operating Procedure (SOP) for decommissioning a software product:

1. Notify stakeholders:

All stakeholders, including users and management, will be informed of the decision to decommission the software product. This will be done well in advance of the actual decommissioning to allow users to plan accordingly.

A mail from our end requesting the approval for decommissioning would be sent to each of these to you.

Point of contact would be appointed and available to you for all further communication regarding the decommissioning with signed document. These documents would then be validated by our legal team and by a hired lawyer agreed upon by both parties.

Upon validation further steps will take effect:

2. Determining the reason for decommissioning:

Before beginning the decommissioning process, it's important to establish the reason why the software product is being decommissioned. This could be due to the product becoming outdated, no longer being used by the organization, or being replaced by a newer product.

So the representative must have a document mentioning the reason for decommissioning ready.

3. Planning of the decommissioning process:

A detailed plan for the decommissioning process would be created, outlining the steps that need to be taken, the timeline, and the resources required. This plan would also include contingency measures in case of any unforeseen issues.

This plan must be approved by your representative.

4. Backup and archive data:

Before decommissioning the software product, all data associated with it should be backed up and archived. This data may be needed in the future for auditing purposes, legal reasons, or to provide continuity for any ongoing processes.

This backed up data would be given to the representative. The representative's signature would be required acknowledging the receiving of backed data.

5.Notify stakeholders of completion:

Once the decommissioning process is complete, all stakeholders should be notified of the successful decommissioning and any next steps that may be required.

A mail from our end would be sent to the stakeholder's informing them of the completion.

6.Conduct post-decommissioning review:

After the decommissioning is complete, a post-decommissioning review should be conducted to evaluate the effectiveness of the process and identify any areas for improvement.

Security aspects to ensure while decommissioning a software product:

1. Data backup and retention:

Before decommissioning the software, it is essential to back up and retain all data associated with the application. This ensures that critical information is not lost and can be accessed if needed for legal or regulatory purposes.

This would be done by us and the back-up would be given to the representative.

2. Data destruction:

All data and information associated with the software product would be securely destroyed or deleted after the software has been decommissioned. This includes any backups, copies, or versions of the data that may exist.

3. Access control:

Access to the decommissioned software product would be restricted to authorized personnel only. This ensures that sensitive data or information is not accidentally or intentionally accessed, modified, or deleted.

As to who these authorized personnel would be, should be decided by the stake holders at the time of decommissioning.

4. Configuration management:

All configurations of the decommissioned software product would be removed, and any security-related configurations or settings would be reviewed to ensure that they are no longer active and do not pose any risk to your organization.

5. System hardening:

The systems and servers on which the software product was installed would be hardened to prevent any unauthorized access or attacks. This may include applying security patches, disabling unnecessary services or ports, and changing default passwords.

6. Audit trails:

An audit trail of all actions taken during the decommissioning process would be maintained to provide a record of who accessed the system and what actions were taken.

7. Communication:

Stakeholders, including users and management, would be informed of the decommissioning process and the security measures being taken to ensure that sensitive data is protected. This process would be done through mail.

By considering these security-based aspects, organizations would ensure that decommissioning the software product is carried out in a secure and controlled manner.

Responsibilities of our support team during decommissioning of a software product:

1. **Communicate with stakeholders:**
   The support team would communicate with all stakeholders, including users, management, and other relevant teams, to inform them of the decommissioning process and answer any questions or concerns they may have.
2. **Plan the decommissioning process:**
   The support team would work with other teams to plan the decommissioning process, including identifying the systems and applications that need to be decommissioned, determining the timeline, and identifying any potential risks or challenges.
3. **Provide technical support:**
   The support team should provide technical support to other teams during the decommissioning process, including assisting with the backup and retention of data, removal of the software from systems, and ensuring that all configurations and settings are properly removed.
4. **Document the process:**
   The support team should maintain accurate documentation of the decommissioning process, including any steps taken, decisions made, and any issues or challenges encountered. This documentation can be used to evaluate the process after it is complete and for future reference.
5. **Test the decommissioning:**
   The support team should work with other teams to test the decommissioning process to ensure that it is effective and that all data and systems are properly secured.
6. **Provide post-decommissioning support:** The support team should continue to provide support to other teams after the decommissioning process is

complete, including answering questions or addressing any issues that may arise.

By taking on these responsibilities, the support team can ensure that the decommissioning process is carried out smoothly, with minimal disruption to the organization, and that all sensitive data and systems are properly secured.