Deployment requirements and steps

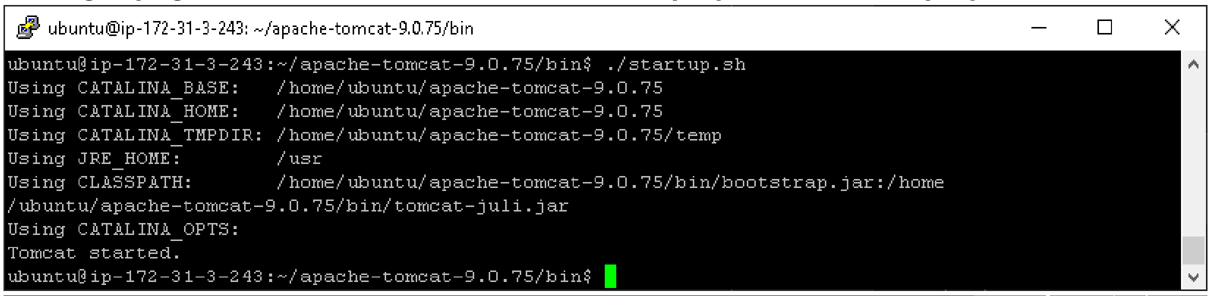
- JDK 1.8.0_361

If one wants to deploy the codes themselves, they also need:

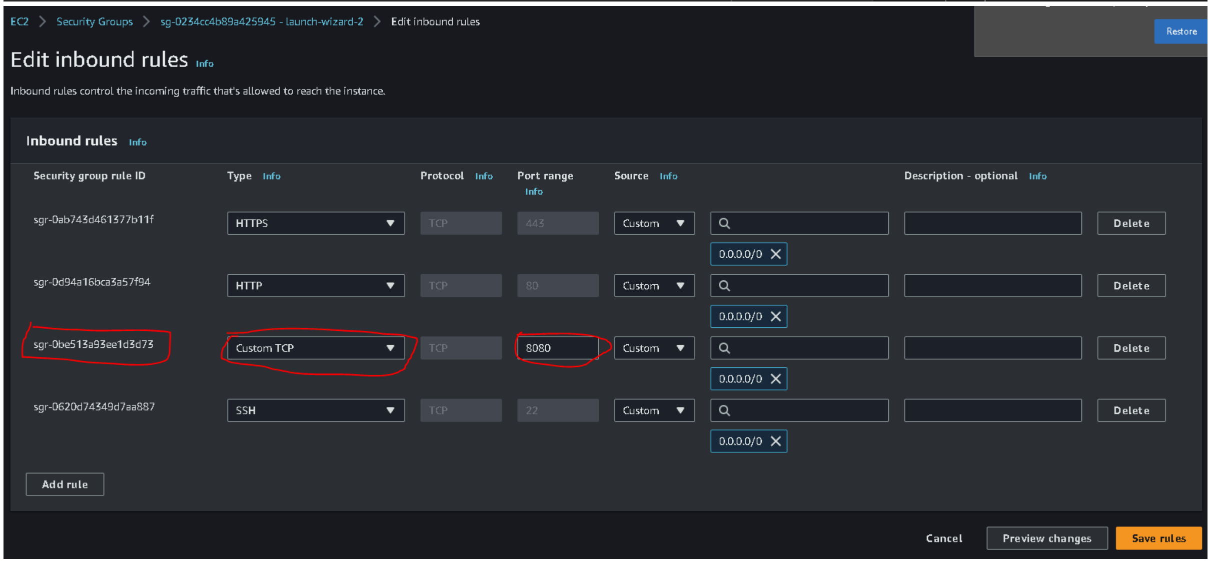
- AWS Toolkit in Eclipse
- Apache Tomcat 9.0.75
- Two SQS FIFO queues
- 2 EC2 instances configured with IAM role that grant access to S3 and SQS, with the web server adding an extra port
- A S3 bucket will ACL policy set to publicly accessible

If one wants to locally run my code and deploy my website, they must first build 2 FIFO SQS queues, a S3 bucket and an EC2 instance (an extra one for the web server). Then, the initialization of AWS objects like AmazonS3 and AmazonSQS should be renamed to the correct instance names. For EC2 instances, one needs to supply credentials or grant IAM role to access S3 and SQS. For S3 bucket, Access control list (ACL) is set to allow public access. For SQS, check Content-based deduplication.

For the EC2 web server, I used an Ubuntu instance and downloaded Tomcat and Java environment there. Upon setting up permissions with "sudo chmod +x" to startup.sh, shutdown.sh and catalina.sh, one can launch the server by "./startup.sh". Before the manager can be accessed, one needs to create an additional port available to everyone in the EC2 security policy. After everything is done, the Apache Tomcat manager page can be seen and one can then deploy the .war web project there.


ubuntu@ip-172-31-3-243: ~/apache-tomcat-9.0.75/bin

```
ubuntu@ip-172-31-3-243:~/apache-tomcat-9.0.75/bin$ ./startup.sh
Using CATALINA_BASE:   /home/ubuntu/apache-tomcat-9.0.75
Using CATALINA_HOME:   /home/ubuntu/apache-tomcat-9.0.75
Using CATALINA_TMPDIR: /home/ubuntu/apache-tomcat-9.0.75/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /home/ubuntu/apache-tomcat-9.0.75/bin/bootstrap.jar:/home
/ubuntu/apache-tomcat-9.0.75/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
ubuntu@ip-172-31-3-243:~/apache-tomcat-9.0.75/bin$
```

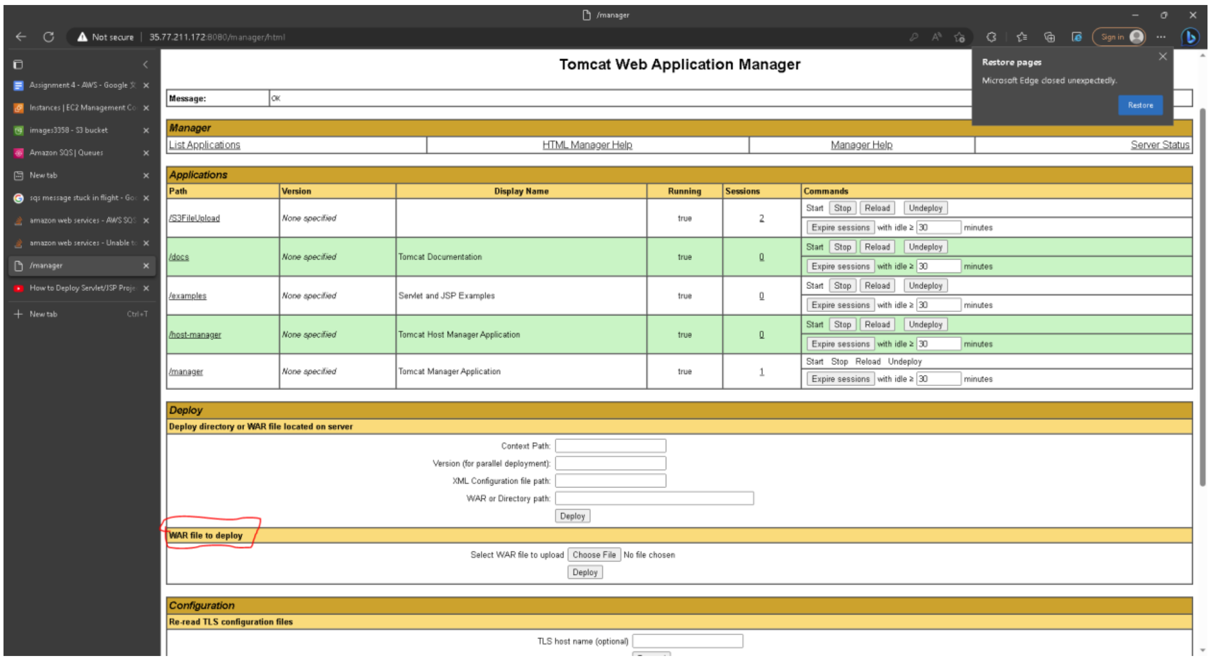

EC2 > Security Groups > sg-0234cc4b89a425945 - launch-wizard-2 > Edit inbound rules

Edit inbound rules

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules

Security group rule ID	Type	Protocol	Port range	Source		Description - optional
sgr-0ab743d461377b11f	HTTPS	TCP	443	Custom	0.0.0.0/0	
sgr-0d94a16bca3a57f94	HTTP	TCP	80	Custom	0.0.0.0/0	
sgr-0be513a93ee1d3d73	Custom TCP	TCP	8080	Custom	0.0.0.0/0	
sgr-0620d74349d7aa887	SSH	TCP	22	Custom	0.0.0.0/0	


/manager
Not secure | 35.77.211.172:8080/manager/html
Assignment 4 - AWS - Google
Instances | EC2 Management Co
images3358 - S3 bucket
Amazon SQS | Queues
New tab
sqs message stuck in flight - Go
amazon web services - AWS SQS
amazon web services - Unable t
/manager
How to Deploy Servlet/JSP Proje
New tab Ctrl+T
Sign in
Restore pages
Microsoft Edge closed unexpectedly.
Restore

Tomcat Web Application Manager

Message:	OK

Manager

List Applications	HTML Manager Help	Manager Help	Server Status

Applications

Path	Version	Display Name	Running	Sessions	Commands
/S3FileUpload	*None specified*		true	2	Start Stop Reload Undeploy / Expire sessions with idle ≥ 30 minutes
/docs	*None specified*	Tomcat Documentation	true	0	Start Stop Reload Undeploy / Expire sessions with idle ≥ 30 minutes
/examples	*None specified*	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy / Expire sessions with idle ≥ 30 minutes
/host-manager	*None specified*	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy / Expire sessions with idle ≥ 30 minutes
/manager	*None specified*	Tomcat Manager Application	true	1	Start Stop Reload Undeploy / Expire sessions with idle ≥ 30 minutes

Deploy

Deploy directory or WAR file located on server

Context Path:
Version (for parallel deployment):
XML Configuration file path:
WAR or Directory path:
Deploy

WAR file to deploy

Select WAR file to upload Choose File No file chosen
Deploy

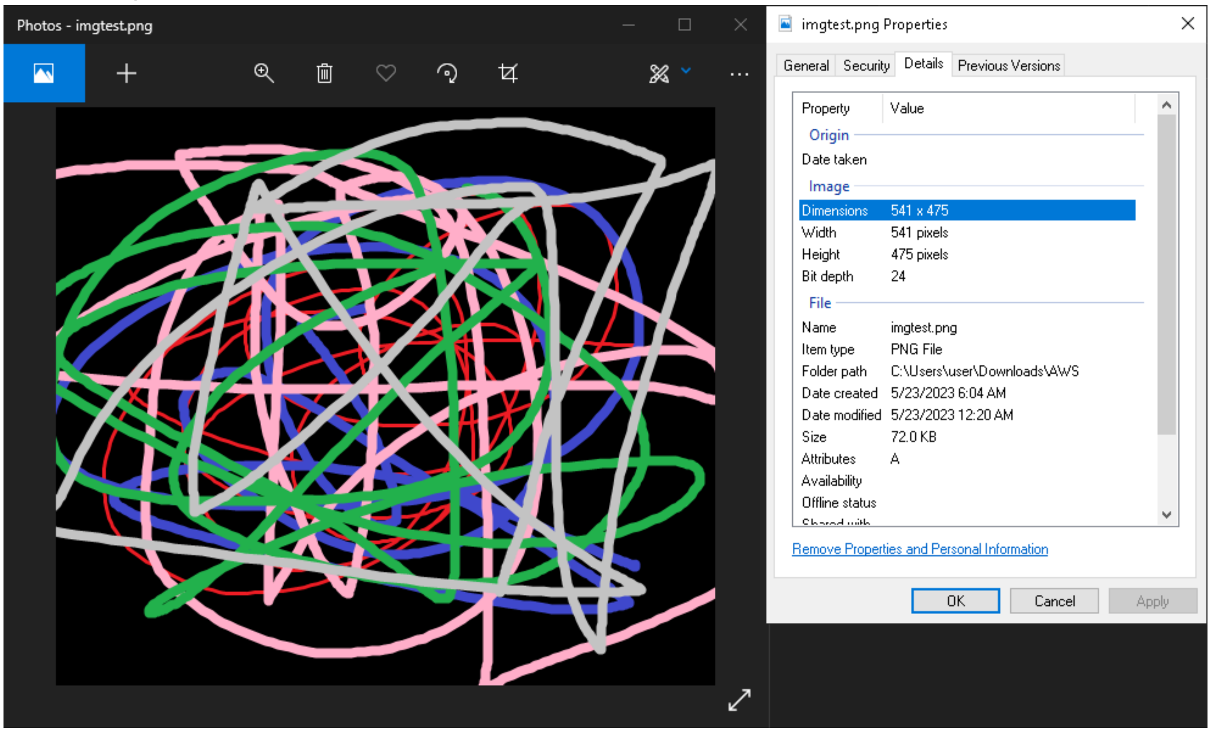
Configuration

Re-read TLS configuration files

TLS host name (optional)

Runtime screenshots

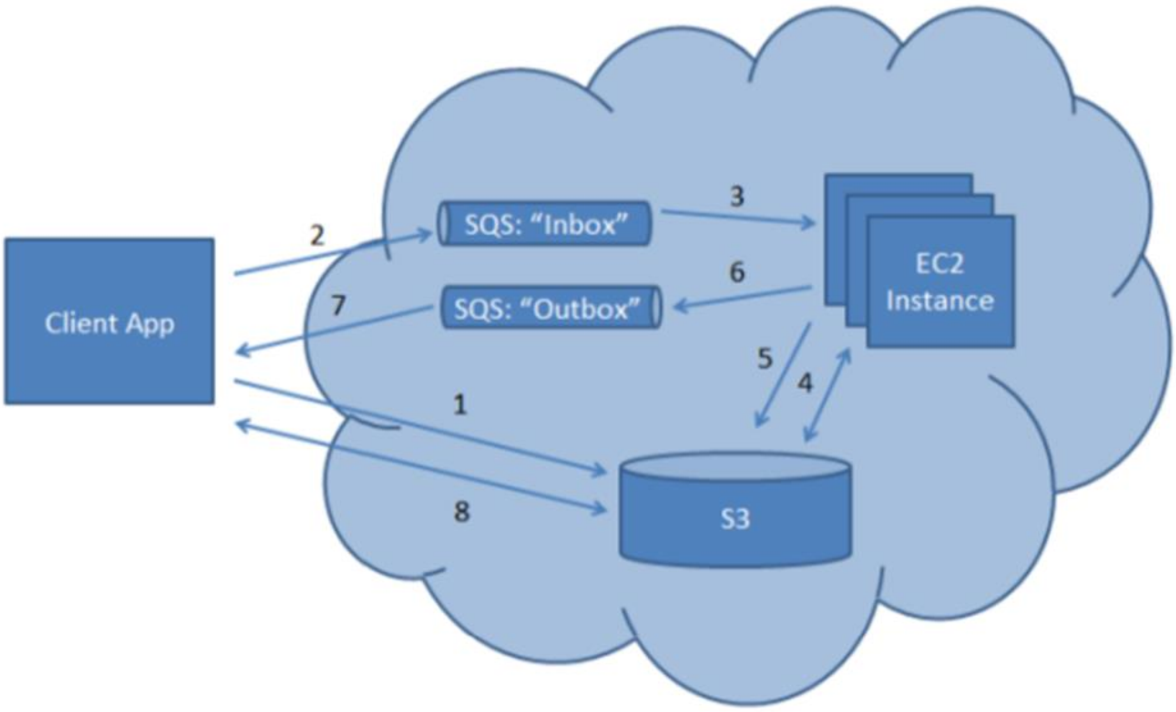
Initial image & its size



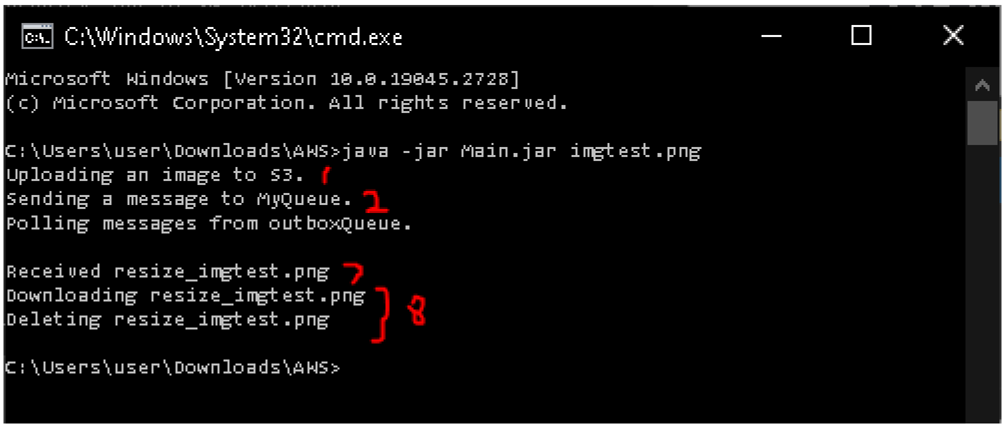
Property	Value
Origin	
Date taken	
Image	
Dimensions	541 x 475
Width	541 pixels
Height	475 pixels
Bit depth	24
File	
Name	imgtest.png
Item type	PNG File
Folder path	C:\Users\user\Downloads\AWS
Date created	5/23/2023 6:04 AM
Date modified	5/23/2023 12:20 AM
Size	72.0 KB
Attributes	A
Availability	
Offline status	

Local console

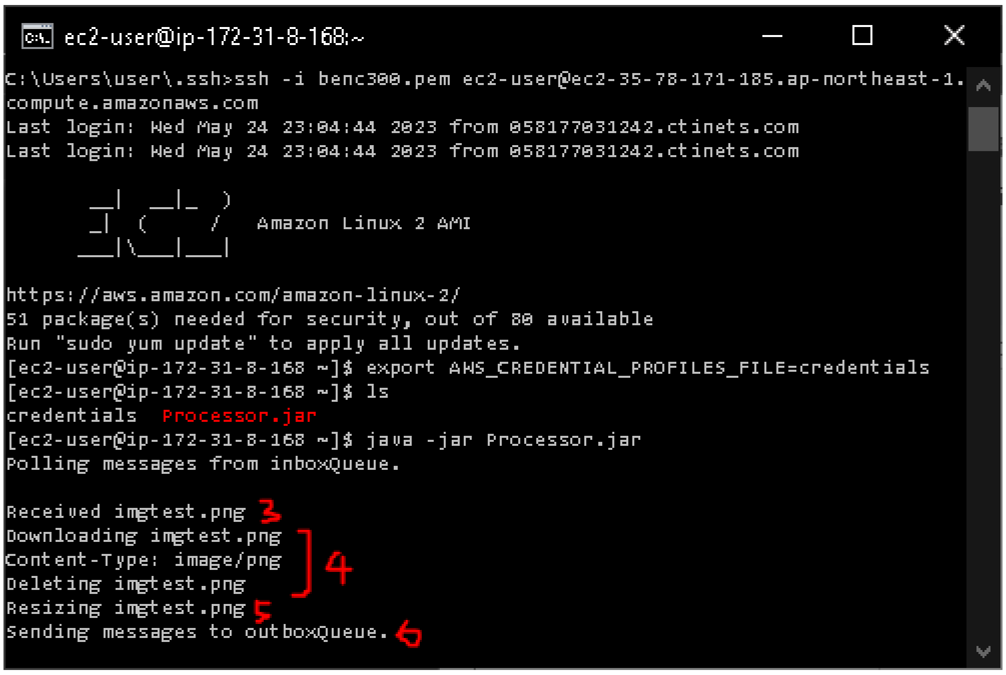
Please refer to the architecture diagram as a reference to each step.



*** IMPORTANT: Please use the filename directly as input (i.e. without path), and do not include any spaces and special characters in the file name. Also, please do not use include "resize_" in the file name to avoid confusion.**



```
Microsoft Windows [Version 10.0.19045.2728]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user\Downloads\AWS>java -jar Main.jar imgtest.png
Uploading an image to S3. 1
Sending a message to MyQueue. 2
Polling messages from outboxQueue.

Received resize_imgtest.png 7
Downloading resize_imgtest.png } 8
Deleting resize_imgtest.png    }

C:\Users\user\Downloads\AWS>
```

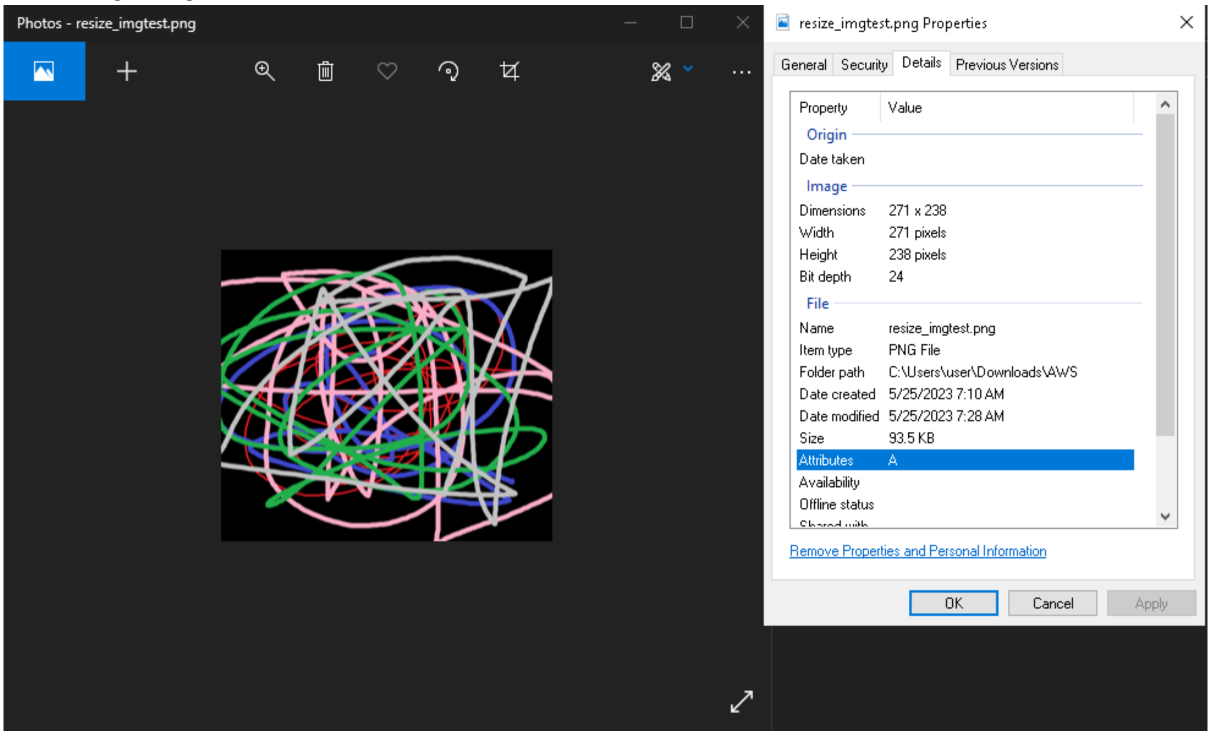


```
ec2-user@ip-172-31-8-168:~
C:\Users\user\.ssh>ssh -i benc300.pem ec2-user@ec2-35-78-171-185.ap-northeast-1.
compute.amazonaws.com
Last login: Wed May 24 23:04:44 2023 from 058177031242.ctinets.com
Last login: Wed May 24 23:04:44 2023 from 058177031242.ctinets.com

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
51 package(s) needed for security, out of 80 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-8-168 ~]$ export AWS_CREDENTIAL_PROFILES_FILE=credentials
[ec2-user@ip-172-31-8-168 ~]$ ls
credentials  Processor.jar
[ec2-user@ip-172-31-8-168 ~]$ java -jar Processor.jar
Polling messages from inboxQueue.

Received imgtest.png 3
Downloading imgtest.png   ]
Content-Type: image/png   ] 4
Deleting imgtest.png      ]
Resizing imgtest.png 5
Sending messages to outboxQueue. 6
```

Note:
3 & 7: Received implies consumption of the message. (The message body is the file name.)
5: Resizing implies sending the resized image to S3.

*** The above EC2 console view sorely demonstrates what is going on inside EC2 instances after running Processor.jar. In the actual case, I used the “screen” command to run the EC2 instances indefinitely even after I exited the SSH session.**

Resulting image

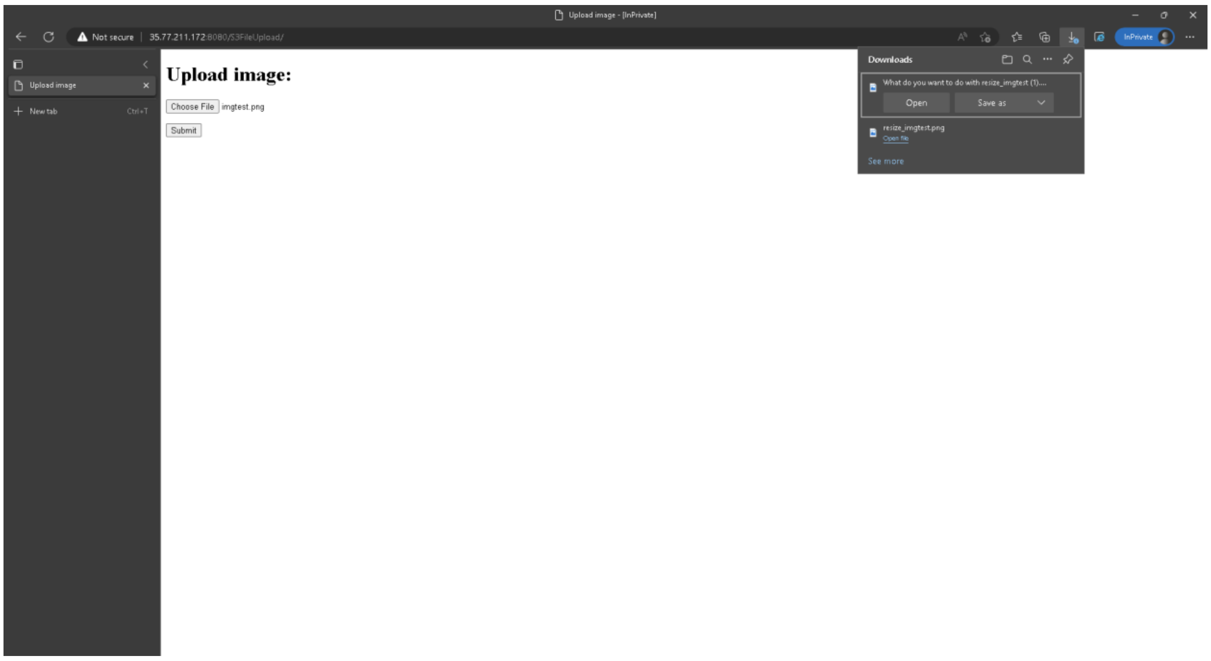


resize_imgtest.png Properties

General | Security | Details | Previous Versions

Property	Value
Origin	
Date taken	
Image	
Dimensions	271 x 238
Width	271 pixels
Height	238 pixels
Bit depth	24
File	
Name	resize_imgtest.png
Item type	PNG File
Folder path	C:\Users\user\Downloads\AWS
Date created	5/25/2023 7:10 AM
Date modified	5/25/2023 7:28 AM
Size	93.5 KB
Attributes	A
Availability	
Offline status	

Website: http://35.77.211.172:8080/S3FileUpload/



Upload image:

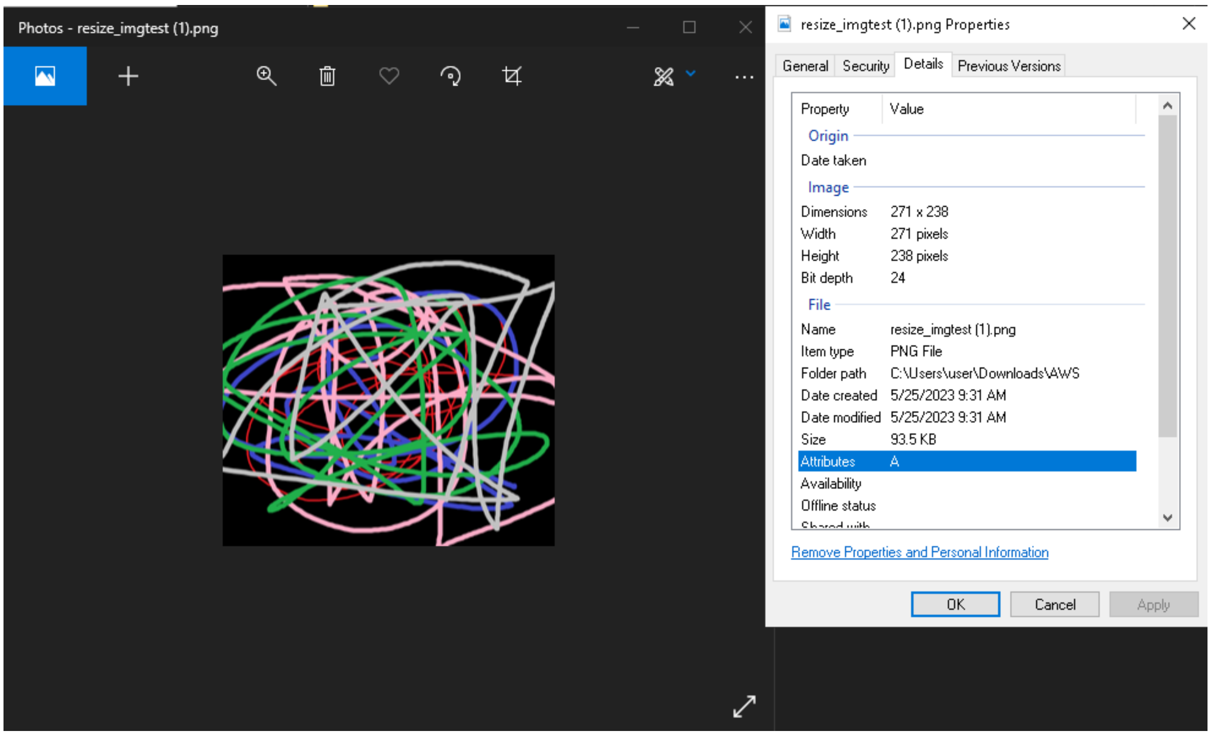
Choose File imgtest.png

Submit

What do you want to do with resize_imgtest (1)....

resize_imgtest.png

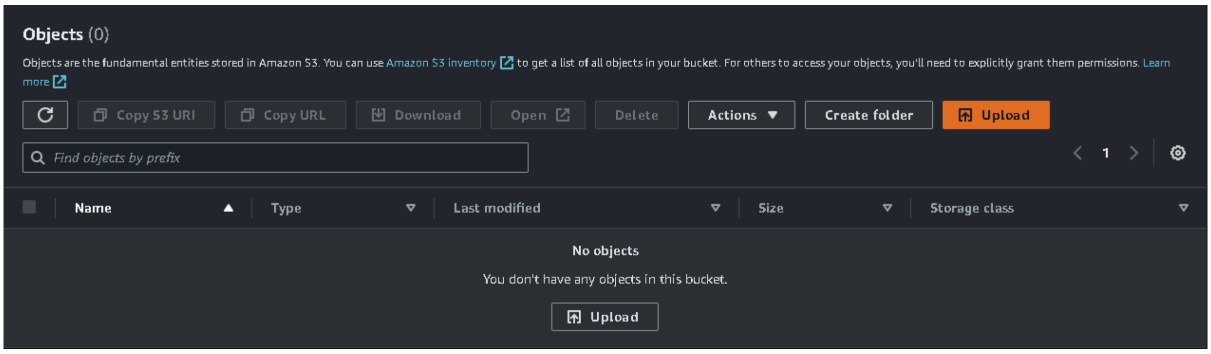
After uploading the image to my website and waiting for roughly a minute, the download for the resized image will show up.


Photos - resize_imgtest (1).png
resize_imgtest (1).png Properties
General
Security
Details
Previous Versions
Remove Properties and Personal Information
OK
Cancel
Apply

Property	Value
Origin	
Date taken	
Image	
Dimensions	271 x 238
Width	271 pixels
Height	238 pixels
Bit depth	24
File	
Name	resize_imgtest (1).png
Item type	PNG File
Folder path	C:\Users\user\Downloads\AWS
Date created	5/25/2023 9:31 AM
Date modified	5/25/2023 9:31 AM
Size	93.5 KB
Attributes	A
Availability	
Offline status	

The downloaded image is the same as the previous resized image.

On successful flow, both SQS queues and the S3 bucket should be empty.



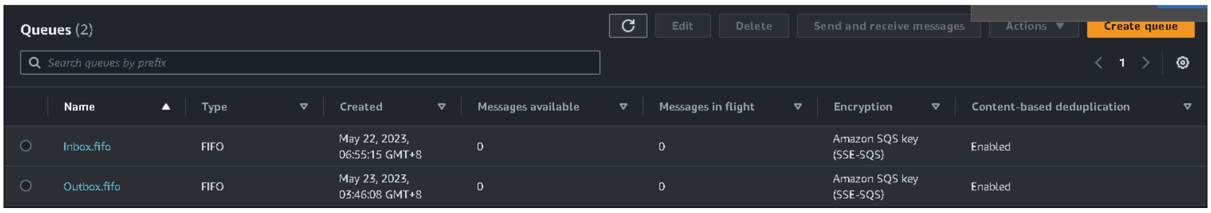
Objects (0)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class

No objects

You don't have any objects in this bucket.



Queues (2)

Name	Type	Created	Messages available	Messages in flight	Encryption	Content-based deduplication
Inbox.fifo	FIFO	May 22, 2023, 06:55:15 GMT+8	0	0	Amazon SQS key (SSE-SQS)	Enabled
Outbox.fifo	FIFO	May 23, 2023, 03:46:08 GMT+8	0	0	Amazon SQS key (SSE-SQS)	Enabled

Code description
S3 bucket name: “images3358”
Inbox.fifo URL: “https://sqs.ap-northeast-1.amazonaws.com/713581367265/Inbox.fifo”
Outbox.fifo URL: “https://sqs.ap-northeast-1.amazonaws.com/713581367265/Outbox.fifo”

The flow of the local code (Main.jar) is as follows:

1. Initialize S3 and SQS classes
2. Send an image with filename as key to S3 and a message with the file name to the inbox SQS queue
3. Poll messages from the outbox SQS queue and look for the “resize_”+fileName message.
4. Once consumed the “resize_”+fileName message, get and delete the resized image with “resize_”+fileName key from S3.

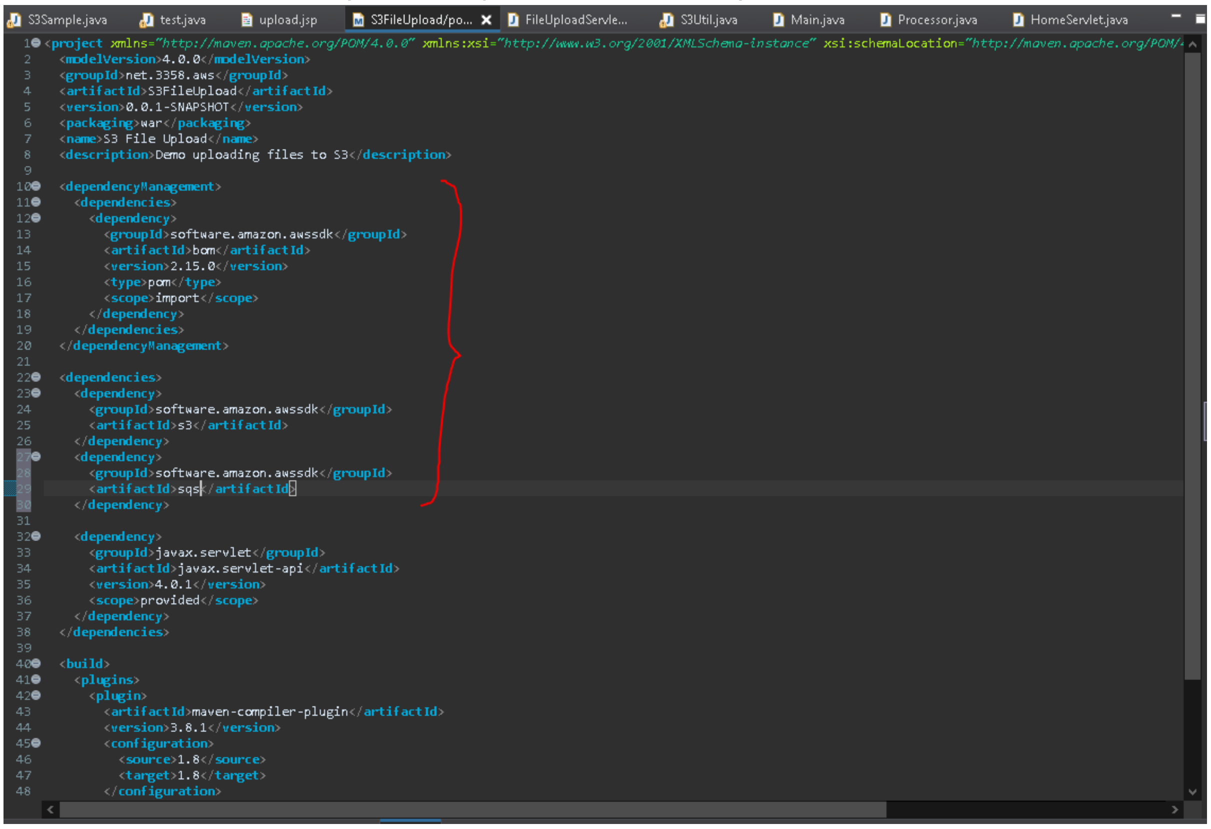
The flow of the EC2 code (Processor.jar) is as follows:

1. Initialize S3 and SQS classes
2. Poll messages from the inbox SQS queue and look for the fileName message.
3. Once consumed the “resize_”+fileName message, download locally and delete the image with the fileName key.
4. Resize the image. This is done by running the linux “convert resize” command in Processor.jar
5. Send and rename the resized image to S3.
6. Send a message with body “resize_”+fileName to the outbox SQS queue..

The flow of the servlet is similar to that of Main.jar, but it uses S3Client and SqsClient in software.amazon.awssdk, S3Client and SqsClient to send requests/response instead of AmazonS3 and AmazonSQS in com.amazonaws. The web project is made in Eclipse and is exported as a .war file.

The page is coded in upload.jsp. Upon uploading a file and clicking the submit button, it calls the servlet FileUploadServlet.java (which then calls a helper class S3Util.java to send upload requests) to send upload and download requests. The file submission is done with

InputStream and file download is done with FileWriter (together with response.setHeader("Content-Disposition", "attachment; filename=\""+ fileName + "\"")). Moreover, the pom.xml needs to be configured to enable importing from software.amazon.awssdk (see below). For more details, please refer to the source code.



```xml
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/
  <modelVersion>4.0.0</modelVersion>
  <groupId>net.3358.aws</groupId>
  <artifactId>S3FileUpload</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>war</packaging>
  <name>S3 File Upload</name>
  <description>Demo uploading files to S3</description>

  <dependencyManagement>
    <dependencies>
      <dependency>
        <groupId>software.amazon.awssdk</groupId>
        <artifactId>bom</artifactId>
        <version>2.15.0</version>
        <type>pom</type>
        <scope>import</scope>
      </dependency>
    </dependencies>
  </dependencyManagement>

  <dependencies>
    <dependency>
      <groupId>software.amazon.awssdk</groupId>
      <artifactId>s3</artifactId>
    </dependency>
    <dependency>
      <groupId>software.amazon.awssdk</groupId>
      <artifactId>sqs</artifactId>
    </dependency>

    <dependency>
      <groupId>javax.servlet</groupId>
      <artifactId>javax.servlet-api</artifactId>
      <version>4.0.1</version>
      <scope>provided</scope>
    </dependency>
  </dependencies>

  <build>
    <plugins>
      <plugin>
        <artifactId>maven-compiler-plugin</artifactId>
        <version>3.8.1</version>
        <configuration>
          <source>1.8</source>
          <target>1.8</target>
        </configuration>
```

The source code of S3FileUpload.war can be viewed in the S3FileUpload folder.

.