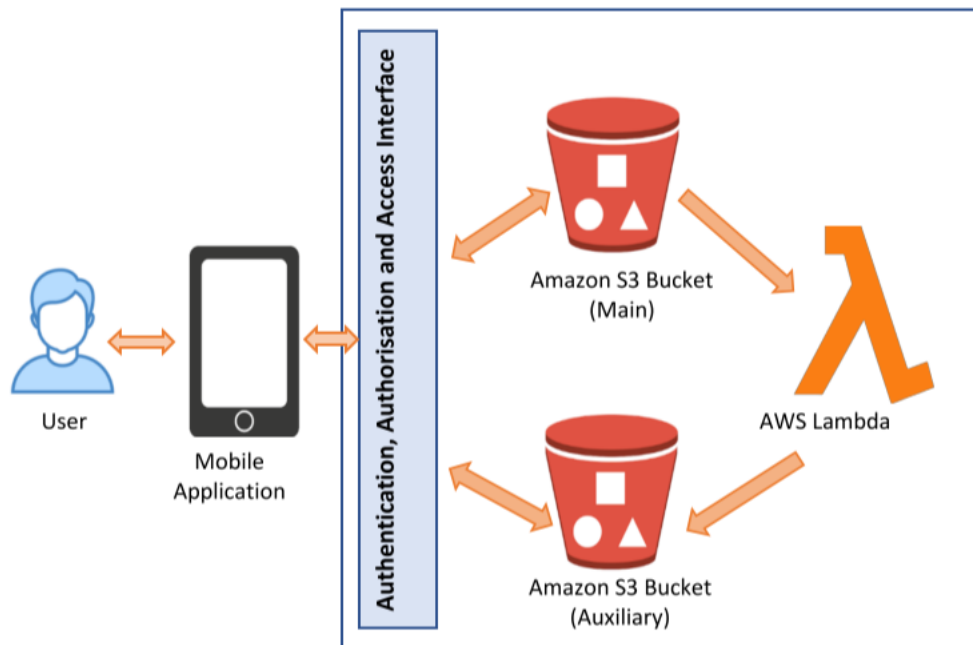


Designing a cloud-based infrastructure for object storage and serverless computing



Please note that this guide does not include the configuration for the authentication, authorisation and access interface.

1. Create S3 buckets and upload a sample image in the bucket

- Navigate to **Amazon S3**, click **Create bucket**
- Bucket 1
 - Bucket Name: adx-a3-joe
 - AWS Region: US East (Ohio) us-east-2
 - Block Public Access: Block all public access
 - Bucket Versioning: Disable
 - Default encryption: Disable
- Bucket 2
 - Bucket Name: adx-a3-joe-resized
 - AWS Region: US East (Ohio) us-east-2
 - Block Public Access: Block all public access
 - Bucket Versioning: Disable
 - Default encryption: Disable

Buckets (2) Info				
Buckets are containers for data stored in S3. Learn more				
<input type="text" value="Find buckets by name"/>				
Name	AWS Region	Access	Creation date	
<input type="radio"/> adx-a3-joe	US East (Ohio) us-east-2	Bucket and objects not public	October 13, 2021, 15:39:14 (UTC+11:00)	
<input type="radio"/> adx-a3-joe-resized	US East (Ohio) us-east-2	Bucket and objects not public	October 13, 2021, 15:47:37 (UTC+11:00)	

- Select **adx-a3-joe** and click **Upload**

adx-a3-joe [Info](#)

Objects | Properties | Permissions | Metrics | Management | Access Points

Objects (1)

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](#)

< 1 >

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	frenchy06.jpg	jpg	October 13, 2021, 15:51:12 (UTC+11:00)	273.4 KB	Standard

2. Create the IAM policy for granting permissions to the Lambda function

- Navigate to **IAM Dashboard**, click **Policies**, **Create Policy**, **JSON**

[IAM](#) > Policies

Policies (869) [Info](#)

A policy is an object in AWS that defines permissions.

< 1 2 3 4 5 6 7 ... 44 >

<input type="checkbox"/>	Policy Name	Type	Use...	Description
<input type="radio"/>	cse2adxLambdaS3Policy	Customer managed	None	This policy delegate the following permission...

[Policies](#) > [cse2adxLambdaS3Policy](#)

Summary

Policy ARN `arn:aws:iam::888742301715:policy/cse2adxLambdaS3Policy`

Description This policy delegate the following permissions to the Lambda function 1. Get the object from the source S3 bucket. 2. Put the resized object into the target S3 bucket. 3. Write logs to Amazon CloudWatch Logs.

Permissions | Policy usage | Tags | Policy versions | Access Advisor

Service	Access level	Resource	Request condition
Allow (2 of 297 services) Show remaining 295			
CloudWatch Logs	Limited: Write	arn:aws:logs:*:*:	None
S3	Limited: Read, Write	Multiple	None

```

1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": [
7         "logs:PutLogEvents",
8         "logs:CreateLogGroup",
9         "logs:CreateLogStream"
10      ],
11      "Resource": "arn:aws:logs:*:*:*"
12    },
13    {
14      "Effect": "Allow",
15      "Action": [
16        "s3:GetObject"
17      ],
18      "Resource": "arn:aws:s3:::adx-a3-joe/*"
19    },
20    {
21      "Effect": "Allow",
22      "Action": [
23        "s3:PutObject"
24      ],
25      "Resource": "arn:aws:s3:::adx-a3-joe-resized/*"
26    }
27  ]
28 }

```

3. Create the execution role

- Navigate to **IAM Dashboard**, click **Roles**, **Create role**
 - Select type of trusted entity: AWS service
 - Choose a use case: Lambda
 - Attach permissions policies: cse2adxLambdaS3Policy
 - Role name: cse2adx-lambda-s3-role
 - Role description: Allows Lambda functions to access Amazon S3 and Amazon CloudWatch

Roles > cse2adx-lambda-s3-role

Summary Delete role

Role ARN	arn:aws:iam::888742301715:role/cse2adx-lambda-s3-role
Role description	Allows Lambda functions to access Amazon S3 and Amazon CloudWatch Edit
Instance Profile ARNs	
Path	/
Creation time	2021-10-13 16:28 UTC+1100
Last activity	Not accessed in the tracking period
Maximum session duration	1 hour Edit

Permissions Trust relationships Tags (1) Access Advisor Revoke sessions

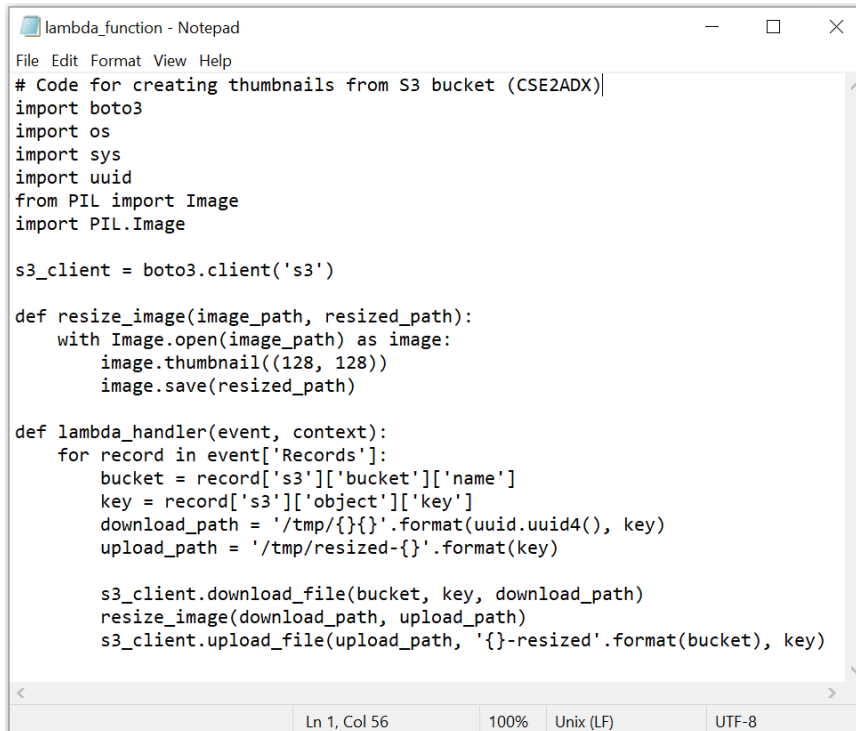
▼ Permissions policies (1 policy applied)

[Attach policies](#) + Add inline policy

Policy name	Policy type
cse2adxLambdaS3Policy	Managed policy

4. Create the function code

- Create a python code in a text editor



```
lambda_function - Notepad
File Edit Format View Help
# Code for creating thumbnails from S3 bucket (CSE2ADX)
import boto3
import os
import sys
import uuid
from PIL import Image
import PIL.Image

s3_client = boto3.client('s3')

def resize_image(image_path, resized_path):
    with Image.open(image_path) as image:
        image.thumbnail((128, 128))
        image.save(resized_path)

def lambda_handler(event, context):
    for record in event['Records']:
        bucket = record['s3']['bucket']['name']
        key = record['s3']['object']['key']
        download_path = '/tmp/{}'.format(uuid.uuid4(), key)
        upload_path = '/tmp/resized-{}'.format(key)

        s3_client.download_file(bucket, key, download_path)
        resize_image(download_path, upload_path)
        s3_client.upload_file(upload_path, '{}-resized'.format(bucket), key)
```

5. Create the deployment package

- I launched an EC2 instance and access it via SSH PuTTY.

```
yum install pip // install pip
sudo rm /usr/bin/python
sudo ln -s /usr/bin/python3.7 /usr/bin/python // run Python3.7
pip install virtualenv // install the virtualenv package
python3 -m venv a3lambda // create a virtual environment
cd a3lambda // go into a3lambda
dir // check directories
```

```
[root@ip-172-31-34-152 ec2-user]# python3 -m venv a3lambda
[root@ip-172-31-34-152 ec2-user]# cd a3lambda
[root@ip-172-31-34-152 a3lambda]# ls
bin include lib lib64 pyvenv.cfg
[root@ip-172-31-34-152 a3lambda]#
```

```
pip install boto3 // install boto3
pip install docutils // install docutils
pip install pillow // install pillow
```

```
sudo find / -type d -iname "xxx" // find where the packages are
yes | cp -ar /source/* /destination/ // copy the packages to a3lambda
```

```
/usr/local/lib/python3.7/site-packages/* // path for the boto3 package
```

```

/usr/local/lib64/python3.7/site-packages/* // path for the PIL package
/usr/lib/python3.7/site-packages/* // path for the docutils package

```

```

/home/ec2-user/a3lambda/lib/python3.7/site-packages // path for the destination

```

```

touch lambda_function.py // create lambda_function.py
cat << EOF > lambda_function.py // write code to lambda_function.py
// Code from Task 2.4 //
EOF

```

```

find . -exec zip lambda-package.zip {} + // create lambda-package.zip

```

```

[root@ip-172-31-34-152 site-packages]# ls
aws_cfn_bootstrap-2.0-py3.7.egg-info  lambda_function.py          pystache-0.5.4-py3.7.egg-info
boto3                                  lambda-package.zip          python_daemon-2.2.3-py3.7.egg-info
boto3-1.18.62.dist-info               lockfile                   python_dateutil-2.8.2.dist-info
botocore                              lockfile-0.11.0-py3.7.egg-info s3transfer
botocore-1.21.62.dist-info            PIL                        s3transfer-0.5.0.dist-info
cfnbootstrap                          Pillow-8.3.2.dist-info     setuptools
daemon                                Pillow.libs               setuptools-47.1.0.dist-info
dateutil                              pip                       setuptools-49.1.3.dist-info
docutils                              pip-20.1.1.dist-info       six-1.16.0.dist-info
docutils-0.14-py3.7.egg-info          pip-20.2.2.dist-info      six.py
easy_install.py                       pkg_resources             urllib3
jmespath                              __pycache__               urllib3-1.26.7.dist-info
jmespath-0.10.0.dist-info             pystache
[root@ip-172-31-34-152 site-packages]#

```

```

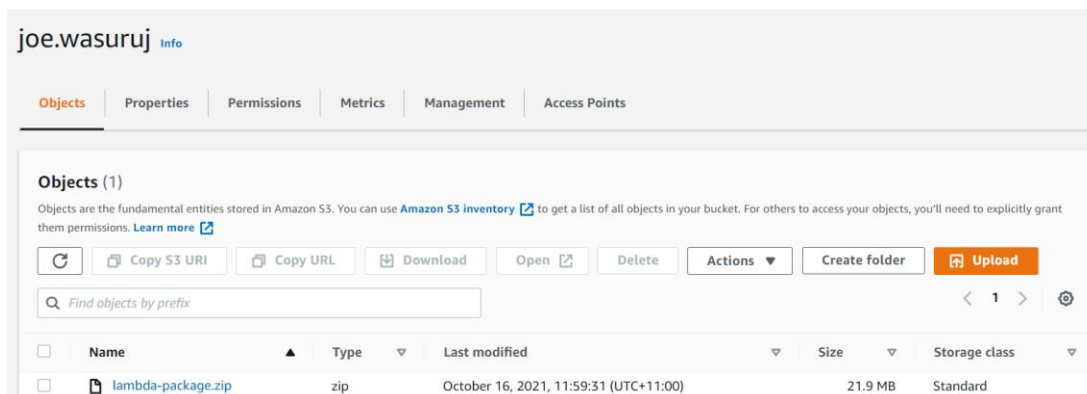
zip -sf lambda-package.zip // check content of lambda-package.zip

```

```

pip install awscli // install AWS CLI
aws configure // enter Access Key ID & Secret Access Key
aws s3 cp /home/ec2-user/a3lambda/lib/python3.7/site-packages/lambda-package.zip
s3://joe.wasuruj/ --acl public-read // upload lambda-package.zip to S3

```



6. Create the Lambda function

- Navigate to **Lambda Dashboard**, click **Functions**, **Create function**
- Function name: **adx-a3-lambda-function**
- Runtime: **Python 3.7**
- Architecture: **x86_64**
- Permissions: Use an existing role: **cse2adx-lambda-s3-role**
- In **Code source**, Upload from: Amazon S3 location
<https://s3.us-east-2.amazonaws.com/joe.wasuruj/lambda-package.zip>

<https://github.com/joe-wasuruj/aws-step-by-step-guides>

adx-a3-lambda-function Throttle Copy ARN Actions

▼ **Function overview** [Info](#)

adx-a3-lambda-function
 Layers (0)

Description

-

Last modified
6 minutes ago

Function ARN
[arn:aws:lambda:us-east-2:888742301715:function:adx-a3-lambda-function](#)

[+ Add trigger](#) [+ Add destination](#)

Code | [Test](#) | [Monitor](#) | [Configuration](#) | [Aliases](#) | [Versions](#)

Code source [Info](#) Upload from

The deployment package of your Lambda function "adx-a3-lambda-function" is too large to enable inline code editing. However, you can still invoke your function.

Code properties

Package size 21.9 MB	SHA256 hash IfdPMkqQUwtUZDDs6UleBwrlTB2rDPc70VnIDfzG6Q=	Last modified October 15, 2021, 12:24 PM GMT+11
-------------------------	--	--

Runtime settings [Info](#) Edit

Runtime Python 3.7	Handler Info lambda_function.lambda_handler	Architecture Info x86_64
-----------------------	--	---

Code | [Test](#) | [Monitor](#) | **[Configuration](#)** | [Aliases](#) | [Versions](#)

[General configuration](#)
[Triggers](#)
[Permissions](#)
[Destinations](#)
[Environment variables](#)
[Tags](#)
[VPC](#)
[Monitoring and operations tools](#)
[Concurrency](#)
[Asynchronous invocation](#)
[Code signing](#)
[Database proxies](#)
[File systems](#)
[State machines](#)

Execution role Edit

Role name
[cse2adx-lambda-s3-role](#)

Resource summary View role document

Amazon CloudWatch Logs
3 actions, 1 resource

To view the resources and actions that your function has permission to access, choose a service.

[By action](#) | **[By resource](#)**

Resource	Actions
arn:aws:logs:*:*	Allow: logs:PutLogEvents Allow: logs:CreateLogGroup Allow: logs:CreateLogStream

Lambda obtained this information from the following policy statements:

- Managed policy [cse2adxLambdaS3Policy](#), statement 0

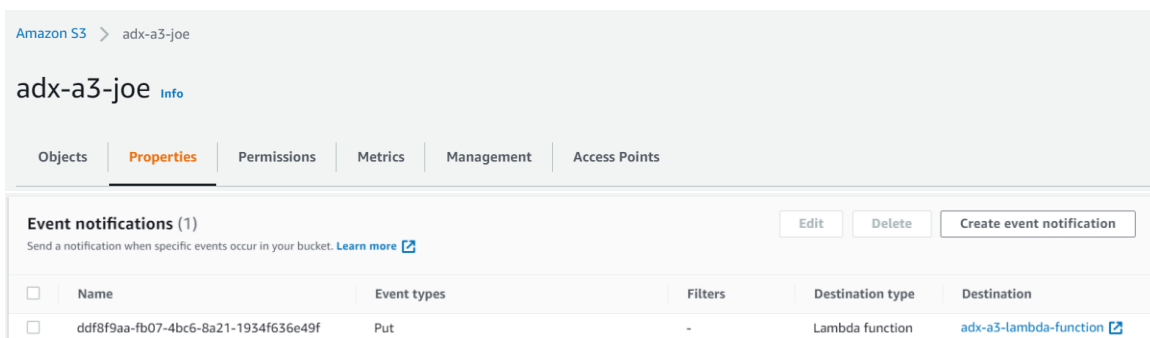
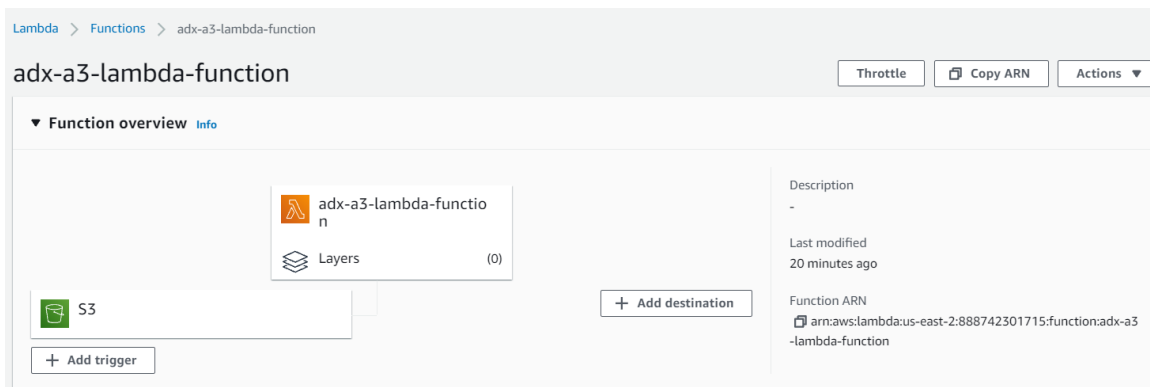
- The OS base that runs Python in AWS does not use the forked libraries such as Pillow that are deprecated. I need to add a layer in Lambda so that it can run Pillow:
- In **adx-a3-lambda-fucntion**, **Code**, **Layers**, click **Add a layer**
- Choose a layer: Specify an ARN

arn:aws:lambda:us-east-2:113088814899:layer:Klayers-python37-Pillow:11

Layers Info						Edit	Add a layer
Merge order	Name	Layer version	Compatible runtimes	Compatible architectures	Version ARN		
1	Klayers-python37-Pillow	11	python3.6, python3.7	-	arn:aws:lambda:us-east-2:113088814899:layer:Klayers-python37-Pillow:11		

7. Configure Amazon S3 to publish events and invoke the Lambda function

- Creating a trigger for the Lambda Function from **AWS Console** can be done in two ways:
- [1] In **Lambda Dashboard**, click **Functions**, **adx-a3-lambda-function**
 - In **Configurations**, click **Triggers**, **Add trigger**
 - Service: S3
 - Bucket: adx-a3-joe
 - Event type: PUT
- [2] In **S3 Dashboard**, click **Buckets**, **adx-a3-joe**, **Properties**
 - In **Event notifications**, click **Create event notification**
 - Event name: S3-trigger
 - Event type: PUT
 - Destination: Lambda function: adx-a3-lambda-function
- I only needed to do one of the two methods. The trigger will automatically appear in both the Lambda Dashboard and the S3 Dashboard.



- Add permissions to the function access policy to allow S3 to invoke the function

```
aws lambda add-permission --function-name adx-a3-lambda-function --principal s3.amazonaws.com
--statement-id s3invoke --action "lambda:InvokeFunction" --source-arn arn:aws:s3:::adx-a3-joe --
source-account *****
```

```
[ec2-user@ip-172-31-34-152 ~]$ aws lambda add-permission --function-name adx-a3-lambda-function
--principal s3.amazonaws.com --statement-id s3invoke --action "lambda:InvokeFunction" --source-a
rn arn:aws:s3:::adx-a3-joe --source-account [REDACTED]
{"Sid": "s3invoke", "Effect": "Allow", "Principal": {"Service": "s3.amazonaws.com"}, "Action": "lambda:I
nvokeFunction", "Resource": "arn:aws:lambda:us-east-2:[REDACTED]:function:adx-a3-lambda-function",
"Condition": {"StringEquals": {"AWS:SourceAccount": "[REDACTED]"}, "ArnLike": {"AWS:SourceArn": "a
rn:aws:s3:::adx-a3-joe"}}}
[ec2-user@ip-172-31-34-152 ~]$
```

Resource-based policy Info					View policy document
A resource-based policy lets you grant permissions to other AWS accounts or services on a per-resource basis.					
Policy statements (2)					Edit Delete Add permissions
<input type="text" value="Find policy statements"/> < 1 >					
	Statement ID	Principal	Conditions	Action	
<input type="radio"/>	lambda-cbab3e49-b798-4eda-b8d8-45983e4356ed	s3.amazonaws.com	StringEquals, ArnLike	lambda:InvokeFunction	
<input type="radio"/>	s3invoke	s3.amazonaws.com	StringEquals, ArnLike	lambda:InvokeFunction	

8. Test using the S3 trigger

- I uploaded 3 photos in the source bucket **adx-a3-joe**
 - chihuahua01.jpg
 - frenchy01.jpg
 - pug01.jpg

adx-a3-joe Info						
Objects	Properties	Permissions	Metrics	Management	Access Points	
Objects (4) 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						
<input type="button" value="Refresh"/> <input type="button" value="Copy S3 URI"/> <input type="button" value="Copy URL"/> <input type="button" value="Download"/> <input type="button" value="Open"/> <input type="button" value="Delete"/> <input type="button" value="Actions"/> <input type="button" value="Create folder"/> <input type="button" value="Upload"/>						
<input type="text" value="Find objects by prefix"/> < 1 >						
<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class	
<input type="checkbox"/>	frenchy06.jpg	jpg	October 13, 2021, 15:51:12 (UTC+11:00)	273.4 KB	Standard	
<input type="checkbox"/>	chihuahua01.jpg	jpg	October 16, 2021, 12:06:14 (UTC+11:00)	63.8 KB	Standard	
<input type="checkbox"/>	frenchy01.jpg	jpg	October 16, 2021, 12:06:15 (UTC+11:00)	99.0 KB	Standard	
<input type="checkbox"/>	pug01.jpg	jpg	October 16, 2021, 12:06:17 (UTC+11:00)	61.6 KB	Standard	

adx-a3-joe-resized Info						
Objects	Properties	Permissions	Metrics	Management	Access Points	
Objects (3) 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						
<input type="button" value="Refresh"/> <input type="button" value="Copy S3 URI"/> <input type="button" value="Copy URL"/> <input type="button" value="Download"/> <input type="button" value="Open"/> <input type="button" value="Delete"/> <input type="button" value="Actions"/> <input type="button" value="Create folder"/> <input type="button" value="Upload"/>						
<input type="text" value="Find objects by prefix"/> < 1 >						
<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class	
<input type="checkbox"/>	chihuahua01.jpg	jpg	October 16, 2021, 12:06:17 (UTC+11:00)	2.2 KB	Standard	
<input type="checkbox"/>	frenchy01.jpg	jpg	October 16, 2021, 12:06:18 (UTC+11:00)	2.2 KB	Standard	
<input type="checkbox"/>	pug01.jpg	jpg	October 16, 2021, 12:06:19 (UTC+11:00)	2.4 KB	Standard	

