

Auto Scale your Amazon EC2 Instances Ahead of Demand

Explore how the new predictive scaling policy of EC2 Auto Scaling helps you improve availability for your applications.

Learn More

We have launched a new allocation strategy, 'Price capacity optimized', that optimizes for both the lowest price and available capacity for the number of Spot Instances that are launching. For more information, see [Allocation strategies](#).

scaling1 created successfully

EC2 > Auto Scaling groups

Auto Scaling groups (1) Info

Edit

Delete

Create an Auto Scaling group

Search your Auto Scaling groups

< 1 > ⚙

<input type="checkbox"/>	Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max
<input type="checkbox"/>	scaling1	luu Version Default	0	Updating capacity...	2	2	4

0 Auto Scaling groups selected

ec2_s3

Delete

Allows EC2 instances to call AWS services on your behalf.

Summary

Edit

Creation date	ARN	Instance profile ARN
December 28, 2022, 15:52 (UTC+02:00)	arn:aws:iam::248375036086:role/ec2_s3	arn:aws:iam::248375036086:instance-profile/ec2_s3
Last activity	Maximum session duration	
None	1 hour	

Permissions

Trust relationships

Tags

Access Advisor

Revoke sessions

Permissions policies (1) Info

Simulate

Remove

Add permissions

You can attach up to 10 managed policies.

Filter policies by property or policy name and press enter.

< 1 > ⚙

<input type="checkbox"/>	Policy name	Type	Description
<input type="checkbox"/>	AmazonS3FullAccess	AWS managed	Provides full access to all buckets via the AWS Management Console.

Permissions boundary - (not set) Info

aws Services Search [Alt+S] Global Lubna Ibrahim

Upload succeeded
View details below.

Summary

Destination s3://lubna-s3	Succeeded 1 file, 190.0 B (100.00%)	Failed 0 files, 0 B (0%)
------------------------------	--	-----------------------------

Files and folders | Configuration

Files and folders (1 Total, 190.0 B)

Find by name

Name	Folder	Type	Size	Status	Error
index.html	-	text/html	190.0 B	Succeeded	-

aws Services Search [Alt+S] N. Virginia Lubna Ibrahim

Allow tags in metadata Info

Select

User data Info

```
#!/bin/bash
sudo apt update -y
sudo apt install apache2 -y
systemctl enable apache2
systemctl start apache2
sudo cat s3://lubna-s3/index.htm > /var/www/html/index.html
```

Summary

Number of instances Info

1

Software Image (AMI)

Canonical, Ubuntu, 22.04 LTS, ...read more
ami-0574da719dca65348

Virtual server type (instance type)

t2.micro

Firewall (security group)

ssh_http_tcp

Storage (volumes)

1 volume(s) - 8 GiB

Instance: i-0b3d965e4a056fc53 (luu-ap)

Instance ID	Public IPv4 address	Private IPv4 addresses
i-0b3d965e4a056fc53 (luu-ap)	-	10.0.128.157
IPv6 address	Instance state	Public IPv4 DNS
-	Running	-

```
[ 72.513386] cloud-init[1217]: Enabling module status.
[ 72.551462] cloud-init[1217]: Enabling module reqtimeou.
[ 72.590827] cloud-init[1217]: Enabling conf charset.
[ 72.628512] cloud-init[1217]: Enabling conf localized-error-pages.
[ 72.667692] cloud-init[1217]: Enabling conf other-vhosts-access-log.
[ 72.705567] cloud-init[1217]: Enabling conf security.
[ 72.743244] cloud-init[1217]: Enabling conf serve-cgi-bin.
[ 72.781085] cloud-init[1217]: Enabling site 000-default.
[ 73.574023] cloud-init[1217]: Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service â /lib/systemd/system/apache2.service.
Starting @0;1;39mThe Apache HTTP Server@0m...
[0;32m OK @0m Started @0;1;39mThe Apache HTTP Server@0m.
[ 74.384371] cloud-init[1217]: Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service â /lib/systemd/system/apach
Starting @0;1;39mDisk Cache Cleanin@emon for Apache HTTP Server@0m...
[0;32m OK @0m Started @0;1;39mDisk Cache Cleanin@Daemon for Apache HTTP Server@0m.
[ 76.029308] cloud-init[1217]: Processing triggers for ufw (0.36.1-4build1) ...
[ 76.182787] cloud-init[1217]: Processing triggers for man-db (2.10.2-1) ...
[ 76.487704] cloud-init[1217]: Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
[ 77.830478] cloud-init[1217]: Running kernel seems to be up-to-date.
[ 77.839057] cloud-init[1217]: No services need to be restarted.
```

aws

Services

Search

[Alt+S]

N. Virginia

Lubna Ibrahim

Allow tags in metadata

Select

User data

```
#!/bin/bash
sudo apt update -y
sudo apt install apache2 -y
sudo systemctl enable apache2
sudo systemctl start apache2
sudo aws s3 cp s3://lubna-s3/index.html /var/www/html/index.html
sudo systemctl restart apache2
```

☐ User data has already been base64 encoded

▼ Summary

Number of instances

1

Software Image (AMI)

Canonical, Ubuntu, 22.04 LTS, ...read more

ami-0574da719dca65348

Virtual server type (instance type)

t2.micro

Firewall (security group)

ssh_http_tcp

Storage (volumes)

1 volume(s) - 8 GiB

Cancel

Launch instance

Allow tags in metadata

Select

User data

```
#!/bin/bash
sudo yum update -y
sudo amazon-linux-extras install php8.0 mariadb10.5
sudo yum install -y httpd
sudo systemctl start httpd
sudo systemctl enable httpd
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install
sudo aws s3 cp s3://lubna-s3/index.html /var/www/html/index.html
sudo systemctl restart apache2
```

☐ User data has already been base64 encoded

▼ Summary

Number of inst

1

Software Imag

Amazon Linux :

ami-0b5eea7698;

Virtual server t

t2.micro

Firewall (securi

ssh_http_tcp

Storage (volum

1 volume(s) - 8

Cancel

We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose **Provide feedback**.

Provide feedback

Successfully created bucket "targetbucket-lubna"

To upload files and folders, or to configure additional bucket settings choose **View details**.

View details

Earn an AWS Learning Badge to showcase your knowledge of S3.

Start now

Account snapshot

Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

View Storage Lens dashboard

Buckets (1) [Info](#)

Copy ARN

Empty

Delete

Create bucket

Buckets are containers for data stored in S3. [Learn more](#)

Find buckets by name

< 1 >

	Name	AWS Region	Access	Creation date
<input type="radio"/>	targetbucket-lubna	US East (N. Virginia) us-east-1	Bucket and objects not public	December 27, 2022, 09:51:13 (UTC+02:00)

December 27, 2022, 10:08 (UTC+02:00)

arn:aws:iam::248375036086:role/lambda_s3

Maximum session duration

1 hour

Last activity

None

Permissions

Trust relationships

Tags

Access Advisor

Revoke sessions

Permissions policies (1) [Info](#)

Simulate

Remove

Add permissions

You can attach up to 10 managed policies.

Filter policies by property or policy name and press enter.

< 1 >

<input type="checkbox"/>	Policy name	Type	Description
<input type="checkbox"/>	<div><div></div><div>AmazonS3FullAccess</div></div>	AWS managed	Provides full access to all buckets via the AWS Management Console.

Permissions boundary - (not set) [Info](#)

Set a permissions boundary to control the maximum permissions this role can have.

This is not a common setting but can be used to delegate permission management to

S3_Lambda_python

Throttle

Copy ARN

Actions

Function overview



S3_Lambda_python



Layers

(0)

+ Add trigger

+ Add destination

Description

-

Last modified

32 seconds ago

Function ARN

arn:aws:lambda:eu-central-1:248375036086:function:S3_Lambda_python

Function URL

-

Successfully created the function S3_Lambda_python. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

File Edit Find View Go Tools Window Test Deploy Changes not deployed

Go to Anything (Ctrl-P)

Environment

- S3_Lambda_python
 - lambda_function.py
 - vois

lambda_function x

```
1 import json
2
3 def lambda_handler(event, context):
4     # TODO implement
5     return {
6         'statusCode': 200,
7         'body': json.dumps('Hello from Lambda!')}
8
9
```

Successfully updated the function S3_Lambda_python.

File Edit Find View Go Tools Window Test Deploy

Go to Anything (Ctrl-P)

Environment

- S3_Lambda_python
 - lambda_function.py
 - vois

lambda_function x vois x Execution results x

```
1 import json
2 import boto3
3
4 s3 = boto3.resource('s3')
5
6 def lambda_handler(event, context):
7     bucket_name = 'targetbucket-lubna'
8     file_name = 'vois'
9     s3.Bucket(bucket_name).put_object(Key=file_name)
10    print('Done')
```

Go to Anything (Ctrl-P)

Environment

- S3_Lambda_python
 - lambda_function.py
 - vois

lambda_function x vois x Execution result x

Execution results

Status: Succeeded Max memory used: 71 MB Time: 795.73 ms

Test Event Name

test1

Response

null

Function Logs

START RequestId: aaf7d661-a498-4f28-9851-1f758a692221 Version: \$LATEST

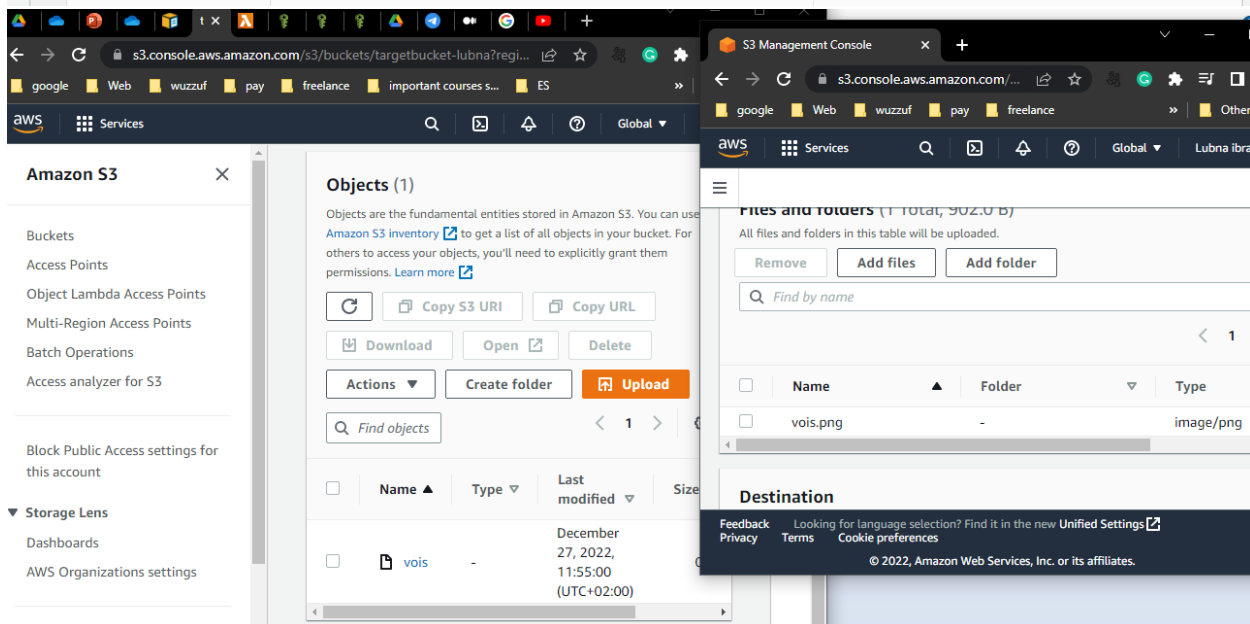
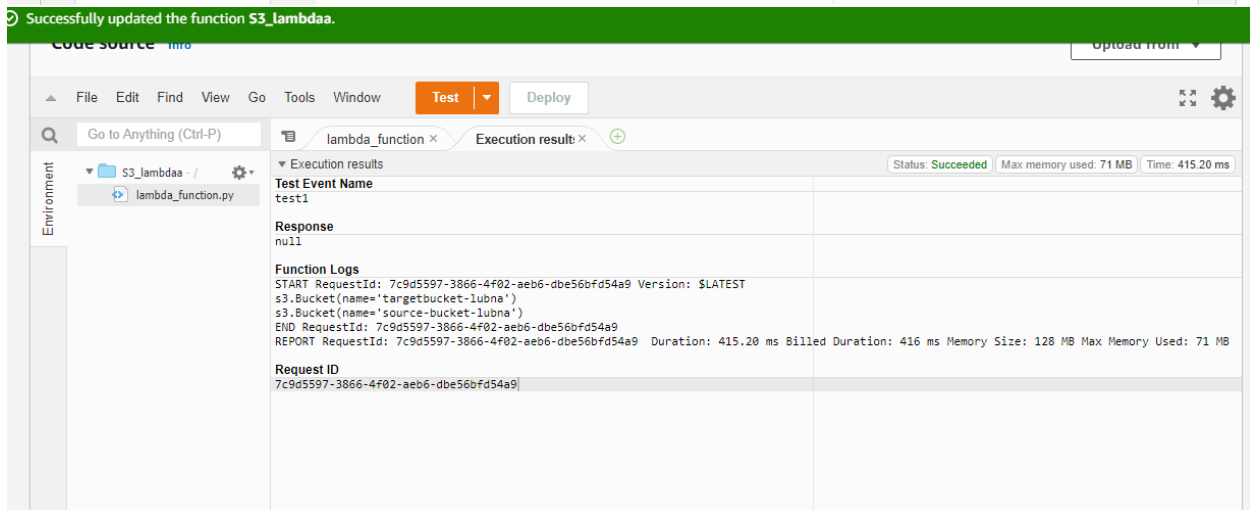
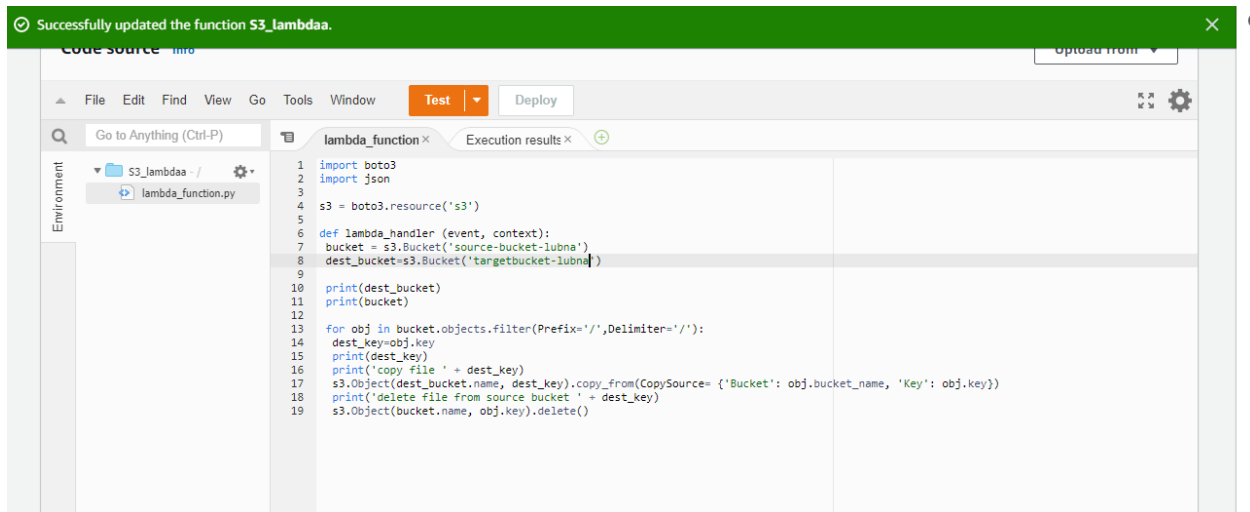
Done

END RequestId: aaf7d661-a498-4f28-9851-1f758a692221

REPORT RequestId: aaf7d661-a498-4f28-9851-1f758a692221 Duration: 795.73 ms Billed Duration: 796 ms Memory Size: 128 MB Max Memory Used: 71 MB

Request ID

aaf7d661-a498-4f28-9851-1f758a692221



Successfully updated the function **S3_lambdaa**.

Code source [Info](#)

Upload from ▼

File Edit Find View Go Tools Window **Test** ▼ Deploy

Go to Anything (Ctrl-P)

Environment

S3_lambdaa - /
lambda_function.py

Execution results

Execution results

Status: Succeeded Max memory used: 71 MB Time: 404.56 ms

Test Event Name

test1

Response

null

Function Logs

START RequestId: 7a5a9ac7-864e-4ba1-b356-b0b412ea2f7b Version: \$LATEST
s3.Bucket(name='targetbucket-lubna')
s3.Bucket(name='source-bucket-lubna')
END RequestId: 7a5a9ac7-864e-4ba1-b356-b0b412ea2f7b
REPORT RequestId: 7a5a9ac7-864e-4ba1-b356-b0b412ea2f7b Duration: 404.56 ms Billed Duration: 405 ms Memory Size: 128 MB Max Memory Used: 71 MB

Request ID

7a5a9ac7-864e-4ba1-b356-b0b412ea2f7b

Successfully updated the function **S3_lambdaa**.

Code source [Info](#)

Upload from ▼

File Edit Find View Go Tools Window **Test** ▼ Deploy

Go to Anything (Ctrl-P)

Environment

S3_lambdaa - /
lambda_function.py

lambda_function.py

1 import boto3
2 import json
3
4 s3 = boto3.resource('s3')
5
6 def lambda_handler(event, context):
7 bucket = s3.Bucket('source-bucket-lubna')
8 dest_bucket=s3.Bucket('targetbucket-lubna')
9
10 print(dest_bucket)
11 print(bucket)
12
13 for obj in bucket.objects.filter(Prefix='',Delimiter=''):
14 dest_key=obj.key
15 print(dest_key)
16 print('copy file ' + dest_key)
17 s3.Object(dest_bucket.name, dest_key).copy_from(CopySource= {'Bucket': obj.bucket_name, 'Key': obj.key})
18 print('delete file from source bucket ' + dest_key)
19 s3.Object(bucket.name, obj.key).delete()