

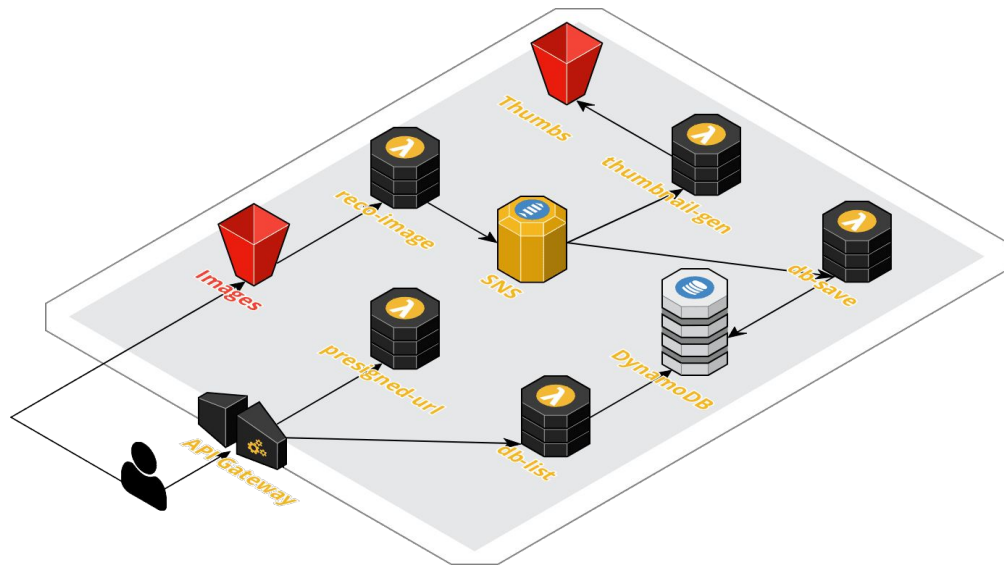
# AWS training - Serverless

Hands On #1 - S3 | IAM - Image bucket

# Overview

This Hands-on is composed of 2 parts:

1. [S3 Part](#) : new S3 bucket to store images to analyze
2. [IAM Part](#) : new IAM role shared by all Lambda function created over the training



# Let's go! | S3 Part

---

Go to Virginia region

N. Virginia ▾

Create a new S3 bucket having these properties:

- **Bucket Name:** serverless-training-img-<xxx>
- **Region:** US East (N.Virginia)

In order to upload images directly from the front-end web-ui, CORS has to be enabled on this bucket:

- Under **Permissions > CORS Configuration**, set the xml setting (see Hint 2 for xml template to use)

Once done -> Go to [IAM Part](#)

## Context:

In this part we create the S3 bucket storing the images to be analyzed by the recognition function.

## Documentation:

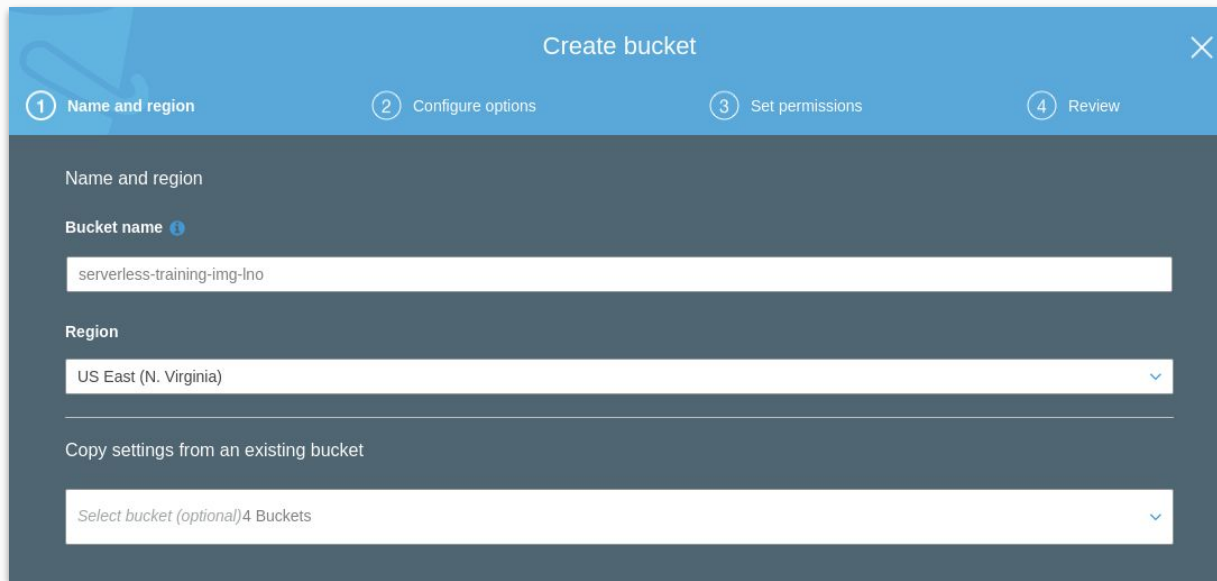
<https://docs.aws.amazon.com/AmazonS3/latest/dev/Welcome.html>

# Hint 1

**S3** - Create a new bucket

“serverless-training-img-**<xxx>**”

(replace **<xxx>** by a unique id)



The screenshot shows the 'Create bucket' wizard in the AWS Management Console. The title bar is blue with a close button (X) on the right. Below the title bar is a progress bar with four steps: 1. Name and region (active), 2. Configure options, 3. Set permissions, and 4. Review. The main content area is dark grey. It has a section titled 'Name and region' with a sub-label 'Bucket name' and a help icon. A text input field contains 'serverless-training-img-lno'. Below this is a 'Region' section with a dropdown menu showing 'US East (N. Virginia)'. At the bottom, there is a section 'Copy settings from an existing bucket' with a dropdown menu showing 'Select bucket (optional)' and '4 Buckets'.

Create bucket

1 Name and region 2 Configure options 3 Set permissions 4 Review

Name and region

Bucket name ⓘ

serverless-training-img-lno

Region

US East (N. Virginia) ▼

Copy settings from an existing bucket

Select bucket (optional) 4 Buckets ▼

# Hint 2

**S3** - Enable CORS from  
"Permissions" > "CORS  
Configuration"

```
<?xml version="1.0" encoding="UTF-8"?>
<CORSConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <CORSRule>
    <AllowedOrigin>*</AllowedOrigin>
    <AllowedMethod>POST</AllowedMethod>
    <MaxAgeSeconds>3000</MaxAgeSeconds>
    <AllowedHeader>*</AllowedHeader>
  </CORSRule>
</CORSConfiguration>
```

# Let's go! | IAM Part

---

Under IAM service, create a new role:

- **Type of trusted entity:** AWS Service
- **Service:** Lambda
- Create a **New Policy** using the **JSON editor** (see Hint 5 for JSON template to use)
- **Policy Name:** serverless\_lambda\_policy
- **Role Name:** serverless\_lambda\_role

## Context:

In this part we create the IAM role which will be used by all lambda functions created over further Hands-on.

*Remark: this is not a best practice to create a unique role gathering all permissions. In actual production applications, we'd rather create a new role for each Lambda function.*


## Documentation:


[https://docs.aws.amazon.com/fr\\_fr/IAM/latest/UserGuide/introduction.html](https://docs.aws.amazon.com/fr_fr/IAM/latest/UserGuide/introduction.html)


# Hint 3


**IAM Role Creation** - Create a new  
AWS Service Lambda role

Select type of trusted entity

**AWS service**  
EC2, Lambda and others

**Another AWS account**  
Belonging to you or 3rd party

**Web identity**  
Cognito or any OpenID provider

**SAML 2.0 federation**  
Your corporate directory

Allows AWS services to perform actions on your behalf. [Learn more](#)

Choose the service that will use this role

**EC2**  
Allows EC2 instances to call AWS services on your behalf.

**Lambda**  
Allows Lambda functions to call AWS services on your behalf.

<a href="#">API Gateway</a>	<a href="#">CodeBuild</a>	<a href="#">EKS</a>	<a href="#">Lambda</a>	<a href="#">SMS</a>
<a href="#">AWS Backup</a>	<a href="#">CodeDeploy</a>	<a href="#">EMR</a>	<a href="#">Lex</a>	<a href="#">SNS</a>

# Hint 4

**IAM Role Creation** - Create a new policy

## Create role

1 2 3 4

▼ Attach permissions policies

Choose one or more policies to attach to your new role.

Create policy

Filter policies ▼ Search Showing 494 results

	Policy name ▼	Used as	Description
<input type="checkbox"/>	▶ AdministratorAccess	Permissions policy (1)	Provides full access to AWS services ...
<input type="checkbox"/>	▶ AlexaForBusinessDeviceSetup	None	Provide device setup access to Alexa...
<input type="checkbox"/>	▶ AlexaForBusinessFullAccess	None	Grants full access to AlexaForBusines...
<input type="checkbox"/>	▶ AlexaForBusinessGatewayExecution	None	Provide gateway execution access to ...
<input type="checkbox"/>	▶ AlexaForBusinessReadOnlyAccess	None	Provide read only access to AlexaFor...
<input type="checkbox"/>	▶ AmazonAPIGatewayAdministrator	None	Provides full access to create/edit/dele...
<input type="checkbox"/>	▶ AmazonAPIGatewayInvokeFullAccess	None	Provides full access to invoke APIs in ...
<input type="checkbox"/>	▶ AmazonAPIGatewayPushToCloudWatchLogs	None	Allows API Gateway to push logs to u...



# Hint 5

**IAM Policy Creation** - Open the "JSON" tab editor and paste the following policy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "lambda:GetLayerVersion",
        "lambda:GetLayerVersionPolicy",
        "s3:*",
        "sns:*",
        "dynamodb:*"
      ],
      "Resource": "*"
    }
  ]
}
```

# Hint 6

**IAM Policy Creation** - Review and  
create policy  
“serverless\_lambda\_policy”

### Review policy

**Name\***

Use alphanumeric and '+=, @\_-' characters. Maximum 128 characters.

**Description**

Maximum 1000 characters. Use alphanumeric and '+=, @\_-' characters.

**Summary**

Service ▼	Access level	Resource	Request condition
Allow (5 of 170 services) <a href="#">Show remaining 165</a>			
<a href="#">CloudWatch Logs</a>	Limited: Write	All resources	None
<a href="#">DynamoDB</a>	Full access	All resources	None
<a href="#">Lambda</a>	Limited: Read	All resources	None
<a href="#">S3</a>	Full access	All resources	None
<a href="#">SNS</a>	Full access	All resources	None

# Hint 7

**IAM Role Creation** - Go back on role creation page, refresh policy list and select the `serverless_lambda_policy` just created

Create role

1234

▼ Attach permissions policies

Choose one or more policies to attach to your new role.

Create policy

Filter policies ▼

Q serverless\_lambda\_policy

Showing 1 result

	Policy name ▼	Used as	Description
<input checked="" type="checkbox"/>	serverless_lambda_policy	None	

# Hint 8

**IAM Role Creation** - Assign the role name `serverless_lambda_role`, review and create it

Create role

1234

Review

Provide the required information below and review this role before you create it.

Role name\*

serverless\_lambda\_role

Use alphanumeric and '+=, @-\_' characters. Maximum 64 characters.

Role description

Allows Lambda functions to call AWS services on your behalf.

Maximum 1000 characters. Use alphanumeric and '+=, @-\_' characters.

Trusted entities

AWS service: lambda.amazonaws.com

Policies

[serverless\\_lambda\\_policy](#)

Permissions boundary

Permissions boundary is not set

# Done !

No test to perform for this Hands-On !