

# LAB: Create an S3 Bucket from a Fargate Cluster Task via IAM policies

## You need:

- An AWS Account
- Configured AWS Cli locally

**Duration of the Lab:** 30 Minutes.

**Difficulty:** medium

## Try to list S3 Buckets from your local machine.

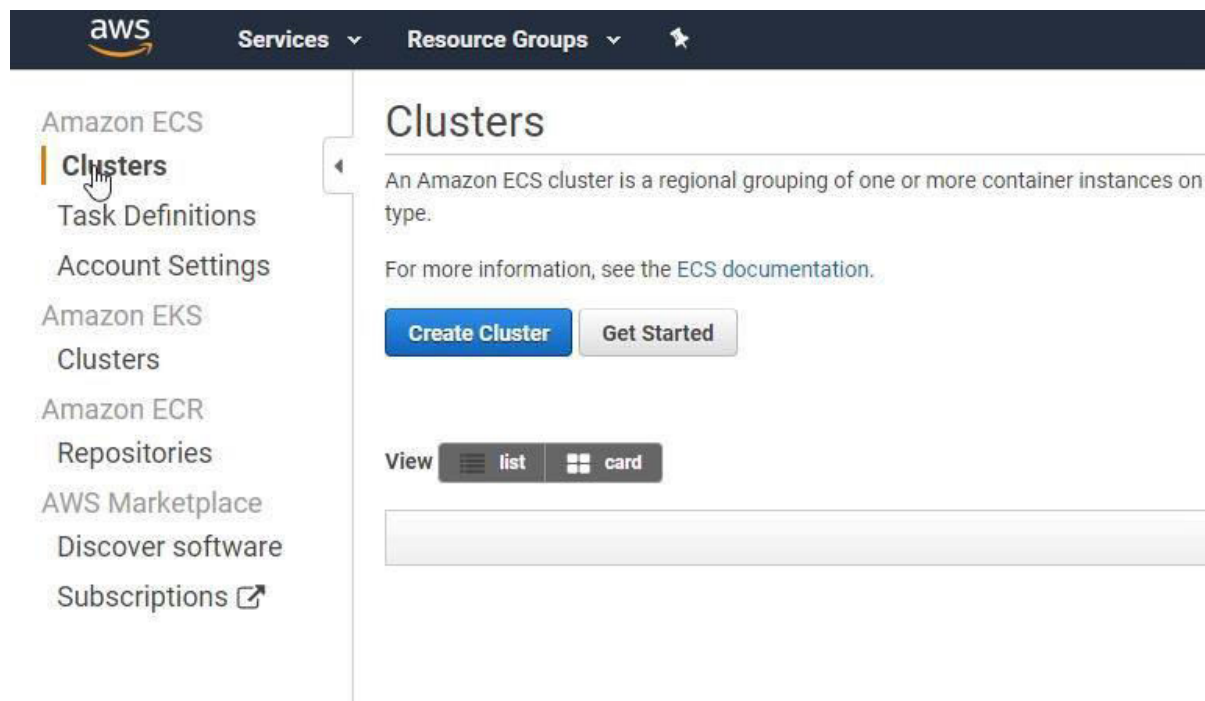
Open a new Terminal/PowerShell and see if you can run the image banst/awscli locally and mount your local aws credentials into the container:

```
docker run --rm -it -v ~/.aws:/root/.aws banst/awscli s3 ls
```

It should, ideally, output nothing (no error message), or S3 buckets, if you still have some. So we know the awscli works, let's use this in a container in the aws ecosystem!

## Create a new Fargate Cluster

If you deleted your cluster in the previous lab then create a new one:



## Complete AWS ECS DevOps Masterclass for Beginners

### Select cluster template

The following cluster templates are available to simplify cluster creation. Additional configuration and integrations can be added later.

#### Networking only

Resources to be created:

Cluster  
VPC (optional)  
Subnets (optional)

**Powered by AWS Fargate**

#### EC2 Linux + Networking

Resources to be created:

Cluster  
VPC  
Subnets  
Auto Scaling group with Linux AMI

#### EC2 Windows + Networking

Resources to be created:

Cluster  
VPC  
Subnets  
Auto Scaling group with Windows AMI

\*Required

Cancel

Next step

## Complete AWS ECS DevOps Masterclass for Beginners

### Configure cluster

Cluster name\*  ⓘ

### Networking

Create a new VPC for your cluster to use. A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Fargate tasks.

Create VPC ☐ Create a new VPC for this cluster

### Tags

Key	Value
<input type="text" value="Add key"/>	<input type="text" value="Add value"/>

### CloudWatch Container Insights

CloudWatch Container Insights is a monitoring and troubleshooting solution for containerized applications and microservices. It collects, aggregates, and summarizes compute utilization such as CPU, memory, disk, and network; and diagnostic information such as container restart failures to help you isolate issues with your clusters and resolve them quickly. [Learn more](#)

CloudWatch Container Insights ☐ Enable Container Insights

\*Required

Cancel

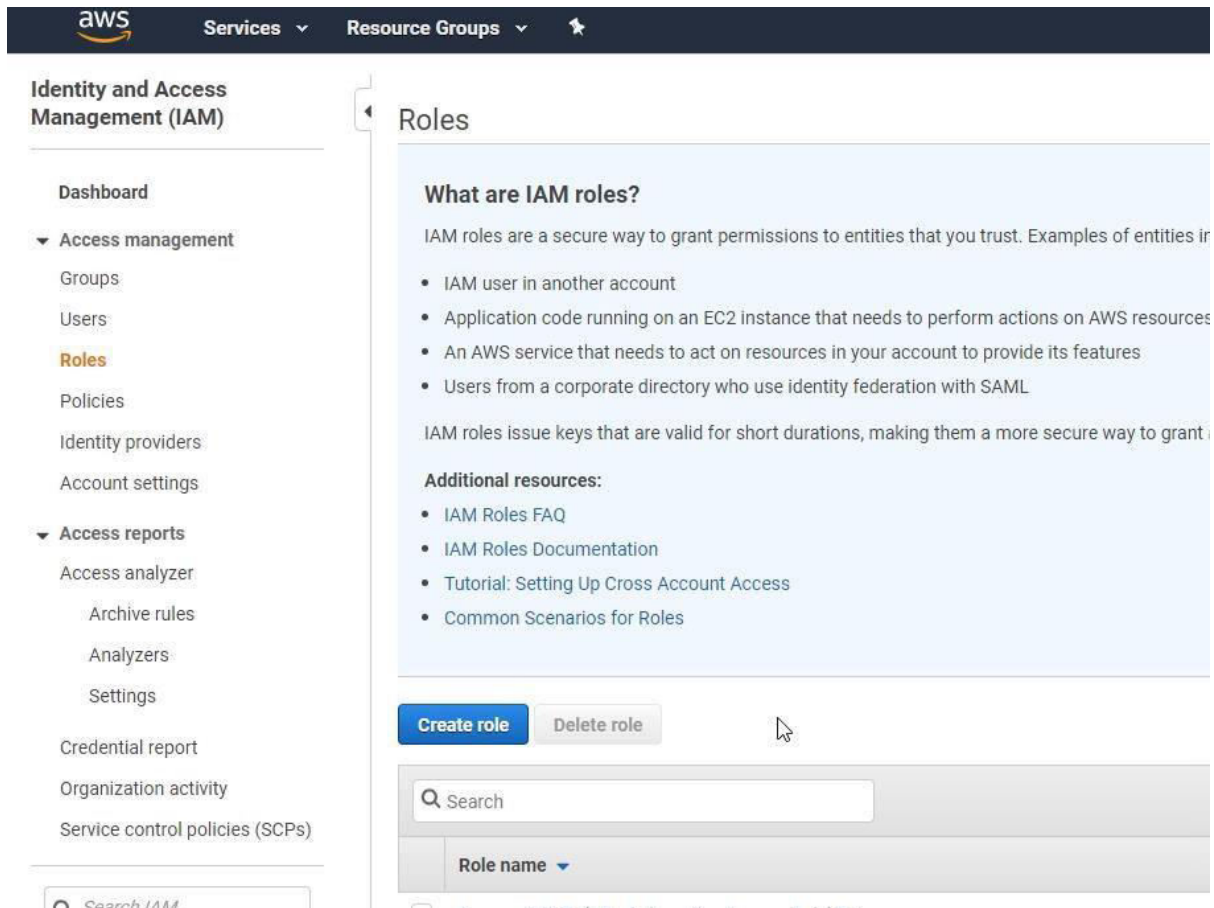
Previous

Create

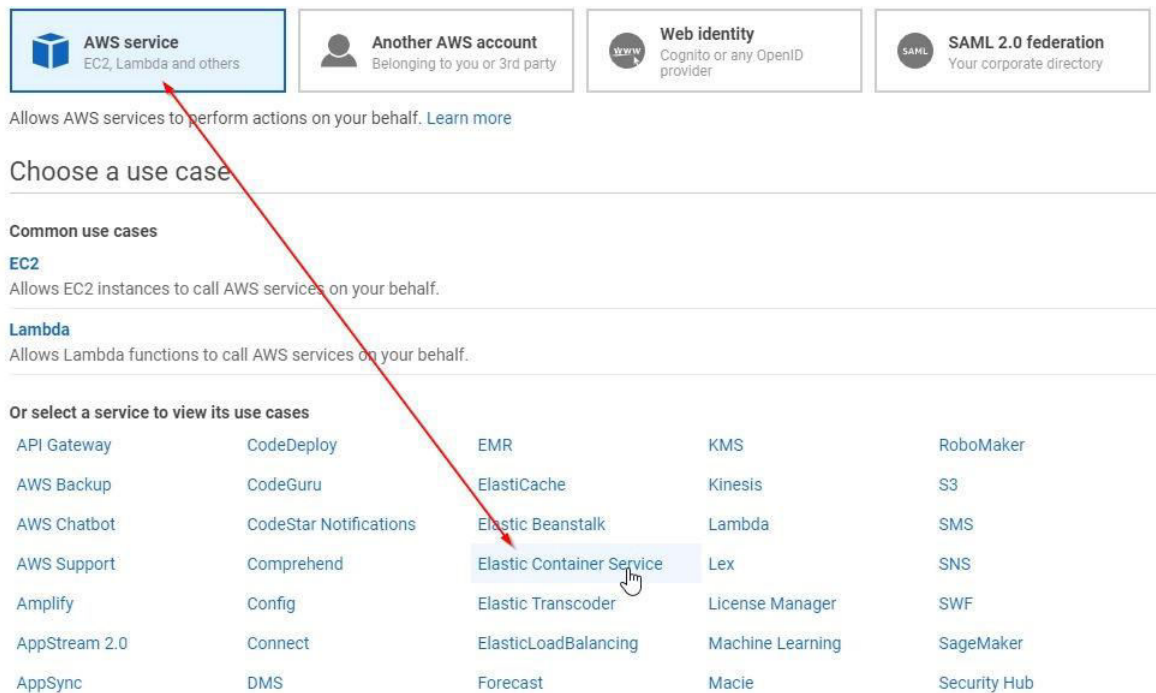
## Create a new IAM Role for your Task

Open the IAM Dashboard -> Roles -> Create role

## Complete AWS ECS DevOps Masterclass for Beginners



This time select “Elastic Container Service” as AWS Service:



Select the Elastic Container Service Task, because our Task will talk to other AWS Services:

## Complete AWS ECS DevOps Masterclass for Beginners

### Select your use case

#### EC2 Role for Elastic Container Service

Allows EC2 instances in an ECS cluster to access ECS.

#### Elastic Container Service

Allows ECS to create and manage AWS resources on your behalf.

#### Elastic Container Service Autoscale

Allows Auto Scaling to access and update ECS services.


#### Elastic Container Service Task

Allows ECS tasks to call AWS services on your behalf.

### Give S3 Full Access Permissions:





#### ▼ Attach permissions policies

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

Create policy

Filter policies ▼

Showing 4 results

	Policy name ▼	Used as
<input type="checkbox"/>	 AmazonDMSRedshiftS3Role	None
<input checked="" type="checkbox"/>	 AmazonS3FullAccess	Permissions policy (1)
<input type="checkbox"/>	 AmazonS3ReadOnlyAccess	None
<input type="checkbox"/>	 QuickSightAccessForS3StorageManagementAnalyticsReadOnly	None

### Give the Role a name, for example “iams3allaccess”:

#### Create role



#### Review

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

**Role name\***   
Use alphanumeric and '+', '@', '-' characters. Maximum 64 characters.

**Role description**   
Maximum 1000 characters. Use alphanumeric and '+', '@', '-' characters.

**Trusted entities** AWS service: ecs-tasks.amazonaws.com

**Policies**  AmazonS3FullAccess 

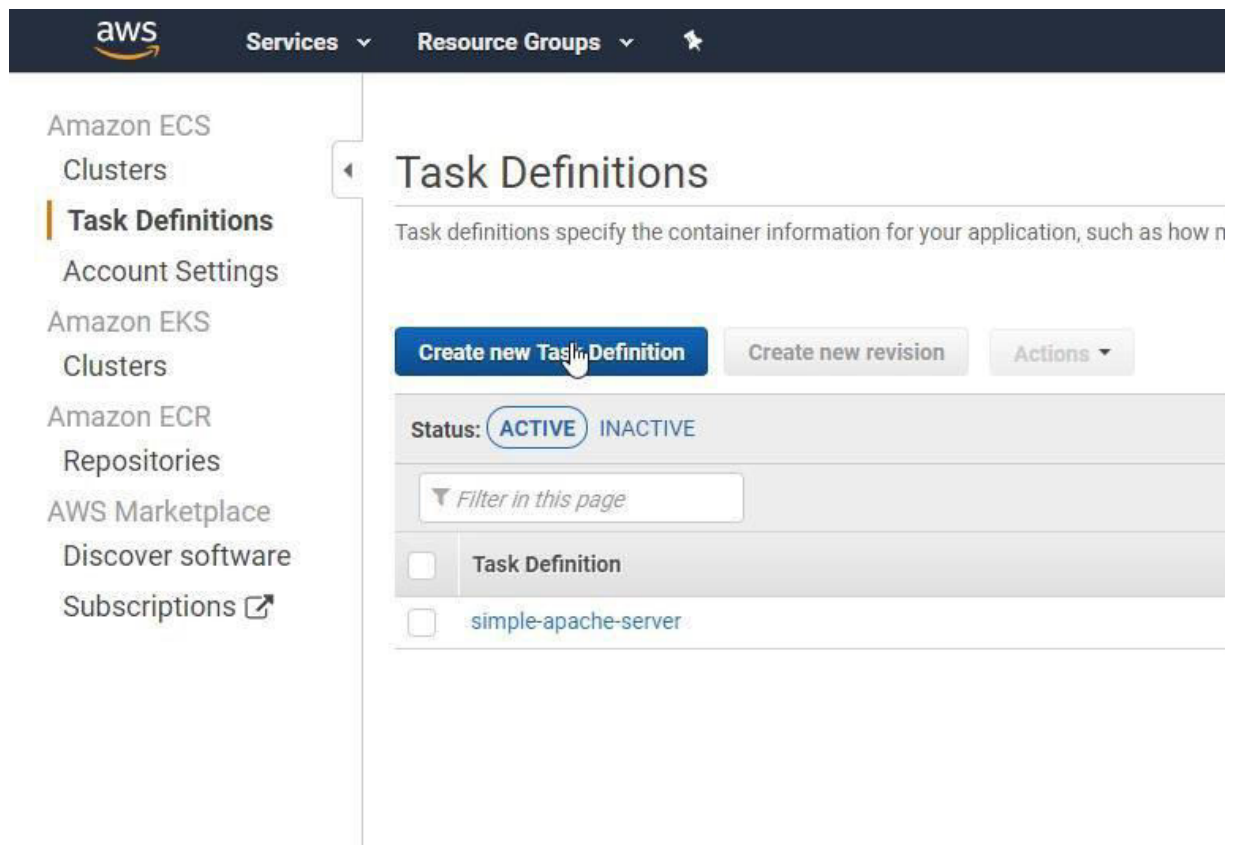
**Permissions boundary** Permissions boundary is not set

*No tags were added.*

### Then Create the Role

## Create a Task Definition

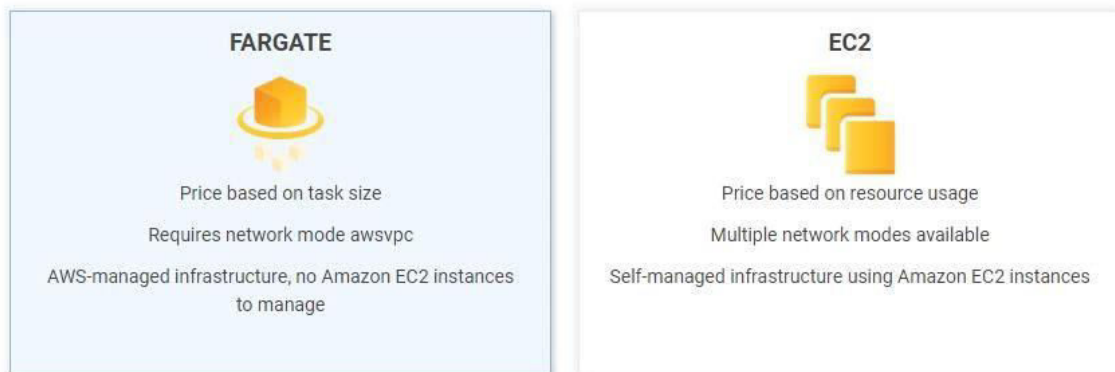
Now we need a Task definition for our task



The screenshot shows the AWS Management Console interface. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', and a search icon. The left sidebar lists navigation options: Amazon ECS Clusters, Task Definitions (highlighted), Account Settings, Amazon EKS Clusters, Amazon ECR Repositories, AWS Marketplace Discover software, and Subscriptions. The main content area is titled 'Task Definitions' and includes a subtitle: 'Task definitions specify the container information for your application, such as how n'. Below the subtitle are three buttons: 'Create new Task Definition' (highlighted with a mouse cursor), 'Create new revision', and 'Actions'. A status filter shows 'ACTIVE' selected and 'INACTIVE' as an option. A search bar contains the text 'Filter in this page'. A table lists task definitions, with the first entry 'Task Definition' having a checkbox and the second entry 'simple-apache-server' also having a checkbox.

### Select launch type compatibility

Select which launch type you want your task definition to be compatible with based on where you want to launch your task.



The comparison shows two launch type options: FARGATE and EC2. FARGATE is highlighted with a blue background and features an icon of a cube on a platform. Its characteristics are: 'Price based on task size', 'Requires network mode awsvpc', and 'AWS-managed infrastructure, no Amazon EC2 instances to manage'. EC2 features an icon of three stacked squares. Its characteristics are: 'Price based on resource usage', 'Multiple network modes available', and 'Self-managed infrastructure using Amazon EC2 instances'.

\*Required

Cancel

Next step

Select the Role you created in the previous IAM Step:

## Complete AWS ECS DevOps Masterclass for Beginners

### Configure task and container definitions

A task definition specifies which containers are included in your task and how they interact with each other. You can also specify data volumes for your containers to use. [Learn more](#)

Task Definition Name\* awslis3create ⓘ

Requires Compatibilities\* FARGATE

Task Role Select a role... ⓘ

- None
- iam ec2 task
- ecsTaskExecutionRole
- iam s3 all access

Network Mode awsvpc ⓘ

If you choose <default>, ECS will start your container using Docker's default networking mode, which is Bridge on Linux and NAT on Windows. <default> is the only supported mode on Windows.

1.5GB of RAM and 0.25 vCPU is enough.

Add containers:

Task execution IAM role ⓘ

This role is required by tasks to pull containers from Amazon ECR. If you already have the `ecsTaskExecutionRole` already, you can reuse it.

Task execution role ⓘ

Task size ⓘ

The task size allows you to specify a fixed task size for the EC2 launch type. Container level task sizes are not supported.

Task memory ⓘ

Task CPU (vCPU) ⓘ

Task memory maximum allocation for container ⓘ

Task CPU maximum allocation for container ⓘ

Container Definitions ⓘ

Add container

Add container

Standard

Container name\* awslis3container ⓘ

Image\* banst/awslis ⓘ

Private repository authentication\* ⓘ

Memory Limits (MiB) Soft limit 128 ⓘ

Add Hard limit ⓘ

Define hard and/or soft memory limits in MiB for your container. Hard and soft limits correspond to the `memory` and `memoryReservation` parameters, respectively, in task definitions. ECS recommends 300-500 MiB as a starting point for web applications.

Port mappings ⓘ

Container port Protocol tcp ⓘ

Add port mapping ⓘ

Image: banst/awslis

And as a command enter `"s3,mb,s3://sometestbucket-123-123-234555"`, so that bucket name should be pretty unique – hopefully. Otherwise change the bucket name to your own pattern/namespace or choose something random



## Complete AWS ECS DevOps Masterclass for Beginners

**ENVIRONMENT**

CPU units

GPUs

Essential

☒

Entry point

comma delimited: sh -c

Command

s3.mb.s3://testbucket-123-12543253

Working directory

/usr/app

Environment variables

You may also designate AWS Systems Manager Parameter Store keys or ARNs using the 'valueFrom' field. ECS will inject the value into containers at run-time.

Key

Add key

Value

Add value

Then create the Task Definition.

## Run the Task

Open your Cluster, but this time go to the Task Tab, not the Service tab. Hit “Run new Task”:

### Cluster : myfargate

Get a detailed view of the resources on your cluster.

Cluster ARN

arn:aws:ecs:eu-central-1:161952721022:cluster/myfargate

Status

ACTIVE

Registered container instances

0

Pending tasks count

0 Fargate, 0 EC2

Running tasks count

0 Fargate, 0 EC2

Active service count

0 Fargate, 0 EC2

Draining service count

0 Fargate, 0 EC2

Services

**Tasks**

ECS Instances

Metrics

Scheduled Tasks

Tags

Capacity Providers

Run new Task

Stop

Stop All

Actions

Desired task status:

Running

Stopped

Filter in this page

Launch type

ALL

	Task	Task definition	Container instance	Last status	Desired status
	No results				

Select your Task Definition, choose Fargate as Launch type, Choose your cluster and give the task group a name:



## Run Task

Select the cluster to run your task definition on and the number of copies of that task to run. To apply container

**Launch type** ☒ FARGATE ☐ EC2 ⓘ

[Switch to capacity provider strategy](#) ⓘ

**Task Definition**  ⓘ

**Platform version**  ⓘ

**Cluster**  ⓘ

**Number of tasks**

**Task Group**  ⓘ

Choose your VPC and choose one Subnet where this task will be placed:

### VPC and security groups

VPC and security groups are configurable when your task definition uses the awsvpc network mode.

**Cluster VPC\***  ⓘ

**Subnets\***

subnet-cfd47ba5  
(172.31.16.0/20) - eu-central-1a  
assign ipv6 on creation: Disabled

ⓧ ⓘ

**Security groups\***  Edit ⓘ

**Auto-assign public IP**  ⓘ

Then Run the Task.

Observe the Task going from Pending to Running to Stopped, because it's not a long running task:

## Complete AWS ECS DevOps Masterclass for Beginners

Clusters > myfargate > Task: 5b5c166e-350b-4983-98fe-9273d17c00f0

Task : 5b5c166e-350b-4983-98fe-9273d17c00f0 Run more like this Stop

Details Tags Logs

Cluster myfargate  
Launch type FARGATE  
Platform version 1.3.0  
Task definition awsclics3create:1  
Group awsclics3create  
Task role iams3allaccess  
Last status RUNNING  
Desired status RUNNING  
Created at 2020-04-07 10:56:54 +0200  
Started at 2020-04-07 10:57:24 +0200

Network

Network mode awsvpc  
ENI Id eni-0ff565e5e1d193345  
Subnet Id subnet-cfd47ba5  
Private IP 172.31.21.253  
Public IP 3.120.30.92  
Mac address 02:0a:5b:f3:1b:72

Containers

Last updated on April 7, 2020 10:57:30 AM (0m ago) Refresh Close

Name	Container Runtime ID ...	Status	Image	Image Digest	CPU Units	Hard/Soft memory li...	Essential	Resource ID
awsclicont...	2aacf832cf402adb696...	RUNNING	banst/awsclic		0	~/128	true	009920f6-bf3c-4e6b...

Head over to the “Logs” tab and observe it created the bucket:

Clusters > myfargate > Task: 5b5c166e-350b-4983-98fe-9273d17c00f0

Task : 5b5c166e-350b-4983-98fe-9273d17c00f0

Details Tags Logs

Filter logs × All 30s 5m 1h

Timestamp (UTC+00:00) ▼	Message
2020-04-07 10:57:27	make_bucket: testbucket-123-12543253

If you check your S3 Dashboard, you’ll see your new Bucket:

aws Services Resource Groups

Amazon S3

Buckets

Batch Operations

Access analyzer for S3

Block public access (account settings)

Feature spotlight 2

Amazon S3

Buckets (1)

Find bucket by name

Name	Region
testbucket-123-12543253	EU (Frankfurt) eu-central-1

## Clean Up

- Delete the Bucket

Lab End