Part 3

AWS Cloud

Matthieu Le Lain, Xavier Roirand

BUT3 INFO, 2023-2024

IUT de Vannes, Université Bretagne Sud

matthieu.le-lain@univ-ubs.fr

Planning

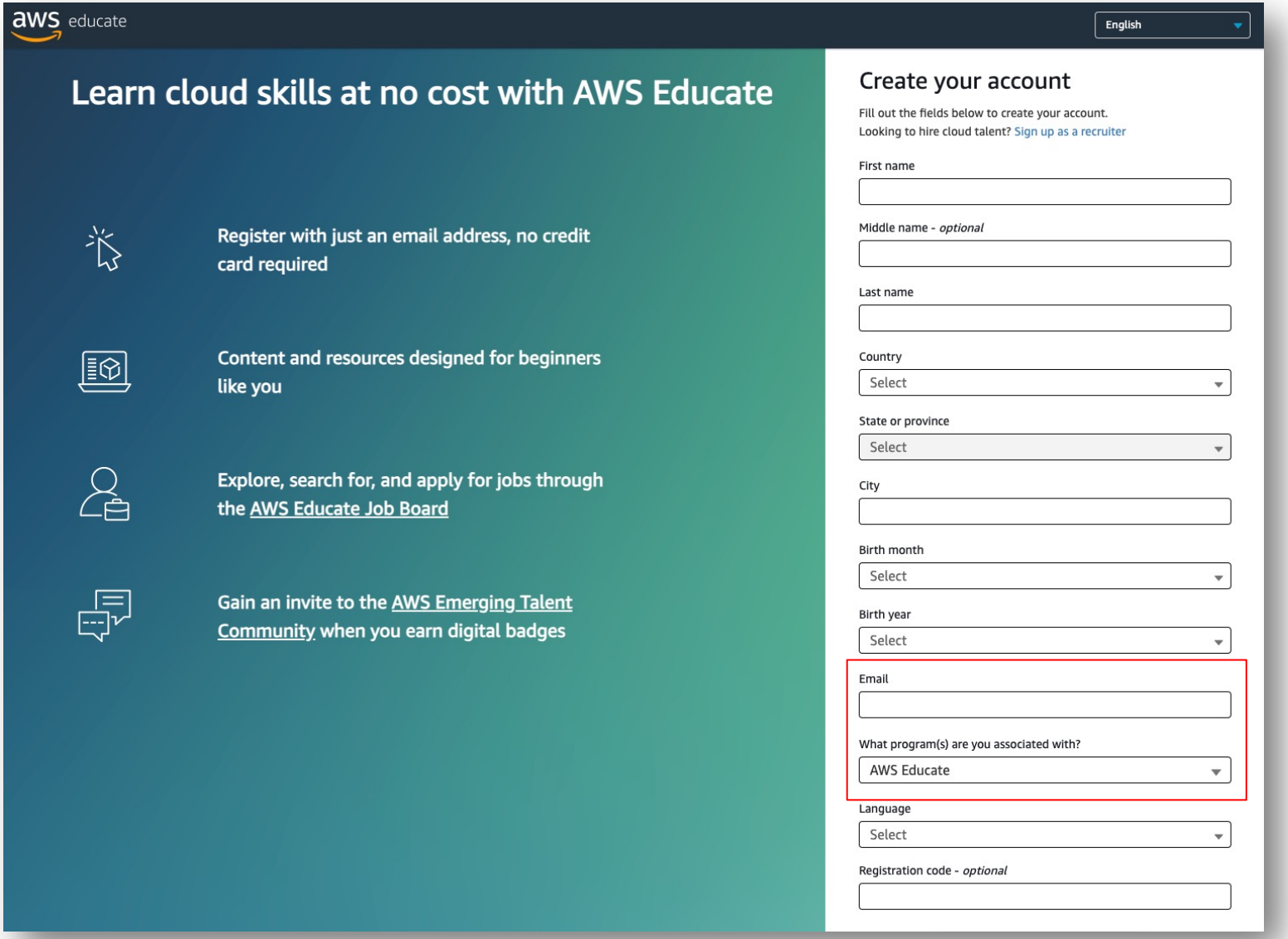
- AWS Educate
- Modules 101
- Ec2 & Container

AWS Educate

Compte Apprenant

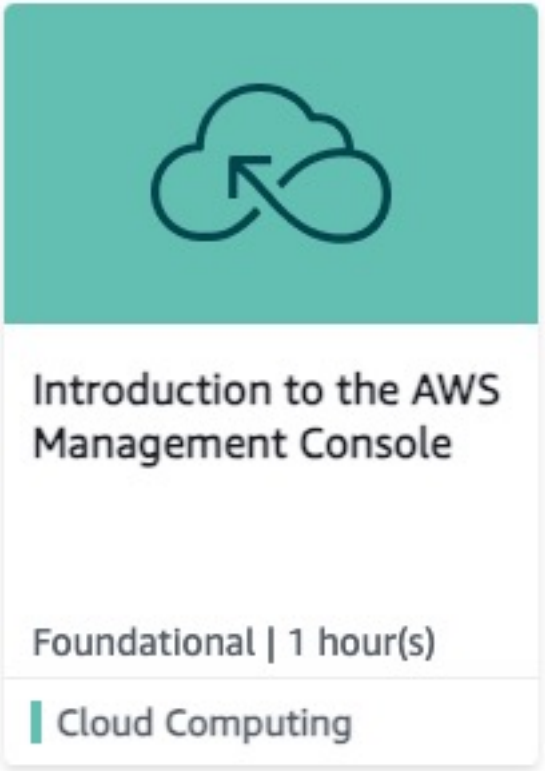
- **Inscription**
  Accès avec votre compte étudiant

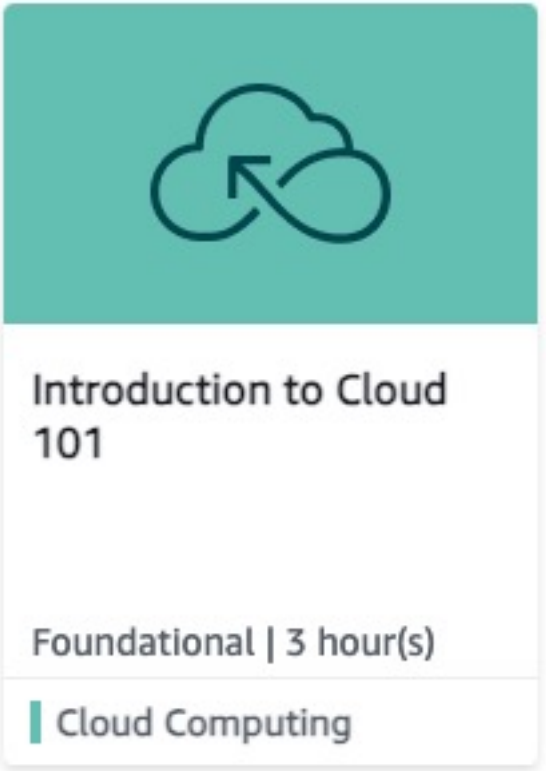
- **Certification / Initiation :**
  Modules de prise en main



AWS Modules

Modules d'initiation



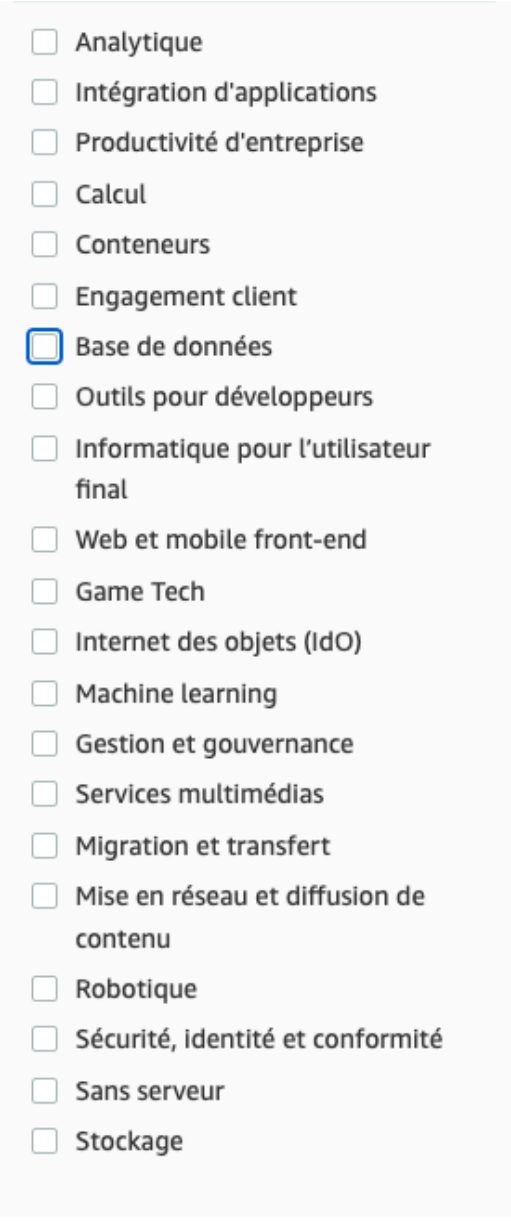


- **Module de démarrage :** AWS 101

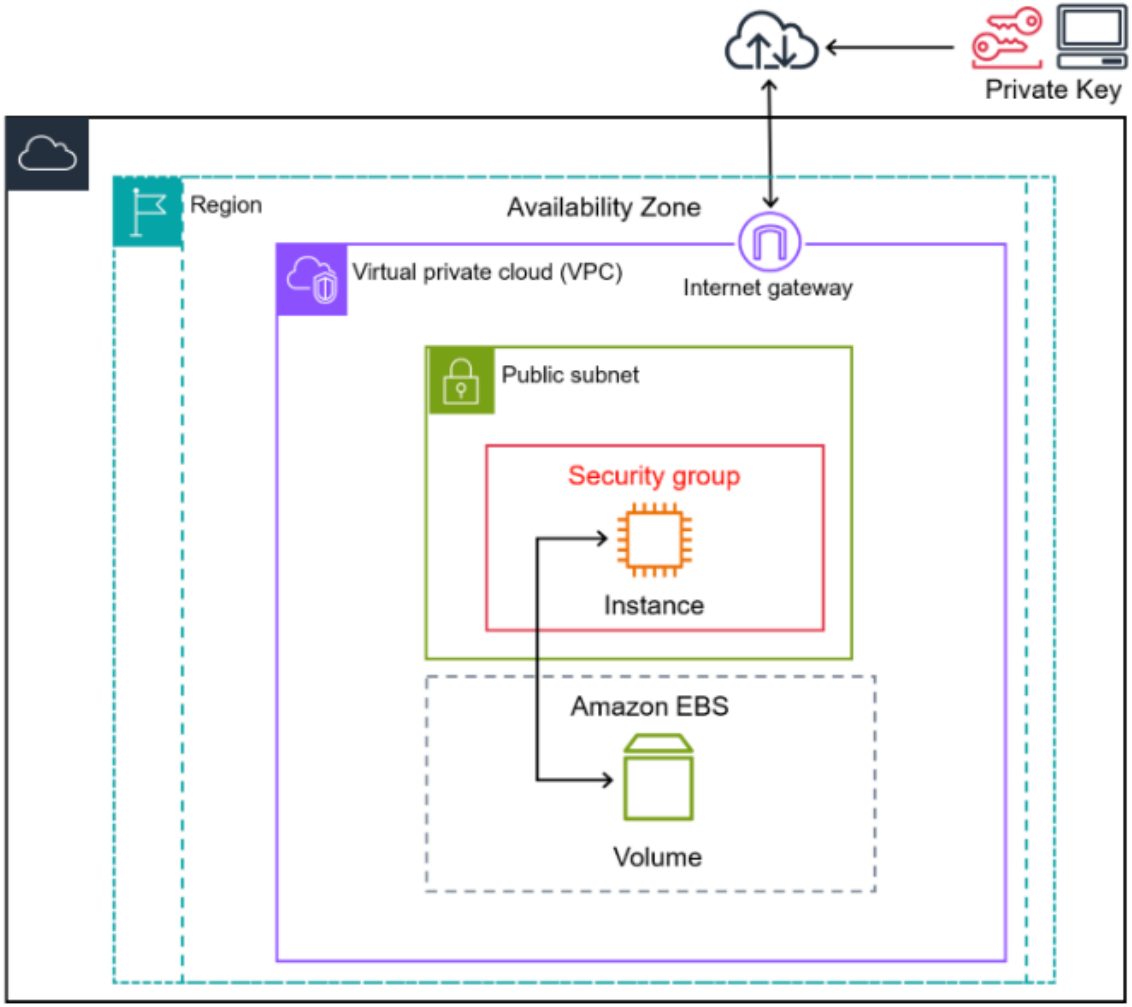
- **Pool de calcul gratuit :** Instance EC2, Elastic Container, BDD…

Environnement riche

- **Eco-système cloud complet**
- **Coût à l'usage =>** Scalabilité



AWS – EC2



EC2 – Linux Instances

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html

AWS – EC2

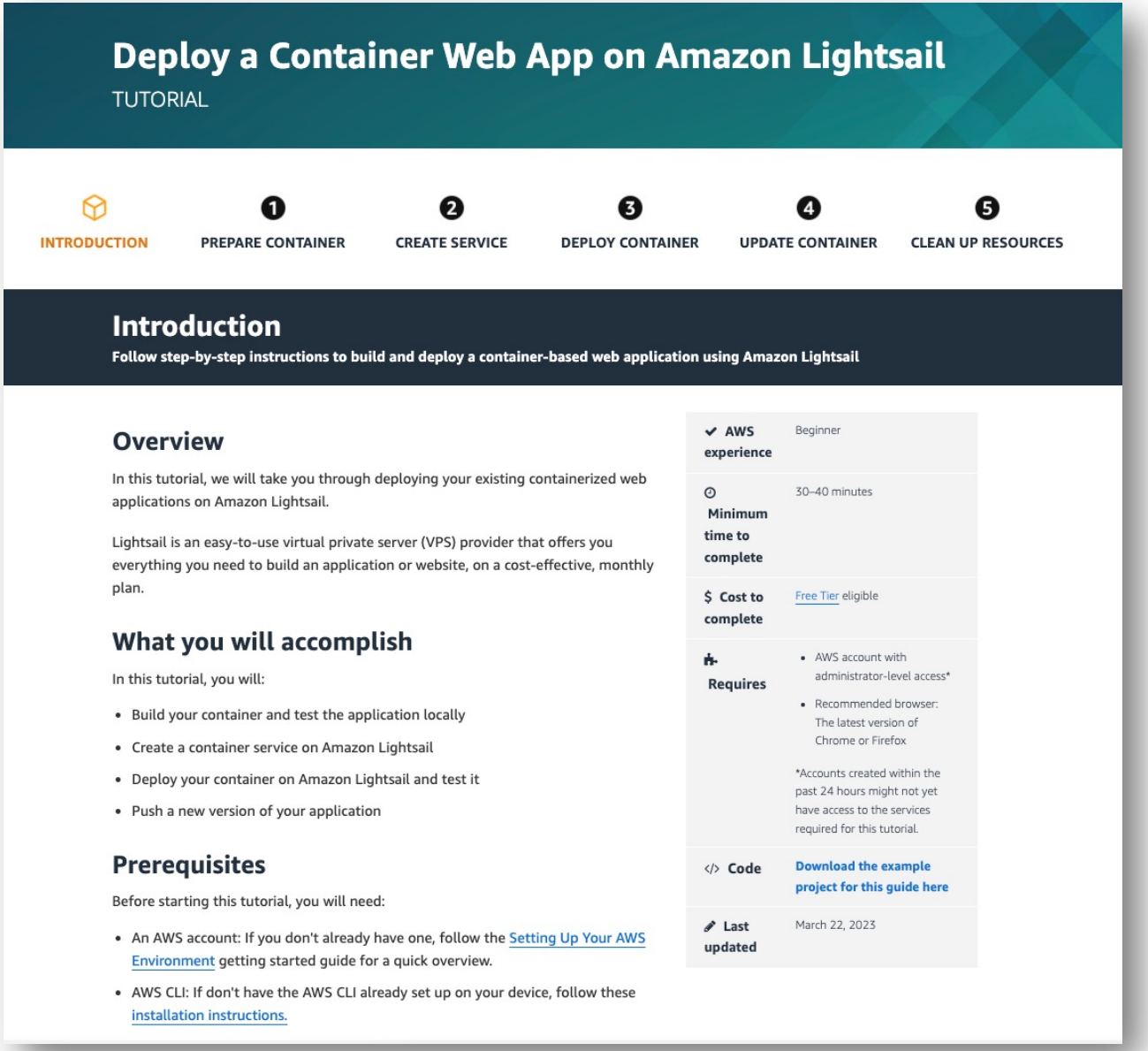
Amazon EC2 Basics & Instances

https://www.youtube.com/watch?v=iHX-jtKIVNA

Autres ressources

AWS Fr : https://www.youtube.com/watch?v=EgLGZ0KH3Hk
https://www.youtube.com/watch?v=8TlukLu11Yo

Container ?



Deploy a Container Web App on Amazon Lightsail

TUTORIAL

INTRODUCTION | 1 PREPARE CONTAINER | 2 CREATE SERVICE | 3 DEPLOY CONTAINER | 4 UPDATE CONTAINER | 5 CLEAN UP RESOURCES

Introduction

Follow step-by-step instructions to build and deploy a container-based web application using Amazon Lightsail

Overview

In this tutorial, we will take you through deploying your existing containerized web applications on Amazon Lightsail.

Lightsail is an easy-to-use virtual private server (VPS) provider that offers you everything you need to build an application or website, on a cost-effective, monthly plan.

What you will accomplish

In this tutorial, you will:

- Build your container and test the application locally
- Create a container service on Amazon Lightsail
- Deploy your container on Amazon Lightsail and test it
- Push a new version of your application

Prerequisites

Before starting this tutorial, you will need:

- An AWS account: If you don't already have one, follow the Setting Up Your AWS Environment getting started guide for a quick overview.
- AWS CLI: If don't have the AWS CLI already set up on your device, follow these installation instructions.

AWS experience	Beginner
Minimum time to complete	30–40 minutes
Cost to complete	Free Tier eligible
Requires	• AWS account with administrator-level access* • Recommended browser: The latest version of Chrome or Firefox *Accounts created within the past 24 hours might not yet have access to the services required for this tutorial.
Code	Download the example project for this guide here
Last updated	March 22, 2023

Container ?

Module 3: Deploy a Container

In this module, we will deploy our container to the cloud

Overview

Now that your Amazon Lightsail container service is ready, the next step is to deploy your container.

Amazon Lightsail is able to deploy containers from public container image repositories such as Docker Hub, Amazon ECR Public Gallery, or your local machine.

In this tutorial, we will deploy directly from your local computer.

What you will accomplish

In this module, you will:

- Push a local container image to Amazon Lightsail
- Deploy a version of your container image on a container service

Minimum time to complete	10 minutes
Module prerequisites	• AWS account with administrator-level access* • Recommended browser: The latest version of Chrome or Firefox *Accounts created within the past 24 hours might not yet have access to the services required for this tutorial.

Container ?

Implementation

Push a local container image

The first step to deploy a container is to push a container image to Amazon Lightsail.

Open a terminal in the directory in which you created your container and enter the following command. Make sure to use the same Region in which you created your Lightsail container service.

Bash

```
aws lightsail push-container-image    \
    --region eu-west-3                \
    --service-name signup-application \
    --label latest                    \
    --image demo-flask-signup:latest

# the command outputs the following lines
f017a6ddb209: Pushed
b94dee417b5e: Pushed
37d77b23a488: Pushed
8e77a3b871e7: Pushed
4bc5d67b6427: Pushed
ce8ffb5c560e: Pushed
4226d5d0360f: Pushed
9d1af766c818: Pushed
d97733c0a3b6: Pushed
c553c6ba5f13: Pushed
48b4a40de359: Pushed
ace9ed9bcfaf: Pushed
764055ebc9a7: Pushed
Digest: sha256:128f84907d30a1fb47c1888720485fa8050cc99bc1034e0cfd1f46d3b6e57e19
Image "demo-flask-signup:latest" registered.
Refer to this image as ":signup-application.latest.1" in deployments.
```

Copy

Container ?

The command invokes docker push to upload your image to Amazon Lightsail. It might take a couple of minutes to complete, depending on your network bandwidth.

The last line of the output is giving the internal name of your container; in our case, it's **:signup-application.latest.1**. Note the name, because we will need it to deploy the container onto the container service. In case you want to access the container name at a later stage, you can enter the following command:

Bash

```
aws lightsail get-container-images    \
    --region eu-west-3                \
    --service-name signup-application

# the command outputs the following lines
{
    "containerImages": [
        {
            "image": ":signup-application.latest.1",
            "digest": "sha256:128f84907d30a1fb47c1888720485fa8050cc99bc1034e0cfd1f46d3b6e57e19",
            "createdAt": "2021-07-17T15:11:49+02:00"
        }
    ]
}
```

Container ?

Push a local container image to LightSail

Now that the container image is stored on Amazon Lightsail, we can deploy that image to the container service.

Lightsail requires the following information to create a deployment:

- The name of the container image to deploy (**:signup-application.latest.1** as returned by the previous command)
- The network port exposed by the container (as described in the Dockerfile configuration)
- (optional) Details of a publicly accessible network endpoint: a TCP port number and protocol; the IP will be assigned automatically.

All this information is stored in a JSON file that you need to create before the deployment.

Create a JSON file named lc.json. If you are working from the downloaded code, you will modify the existing lc.json file which is in the same directory as your container. Copy the following json into the file:

JSON

```json
{
    "serviceName": "signup-application",
    "containers": {
        "signup-app-container": {
            "image": ":signup-application.latest.1",
            "ports": {
                "80": "HTTP"
            }
        }
    },
    "publicEndpoint": {
        "containerName": "signup-app-container",
        "containerPort": 80
    }
}
```

Copy

Déployer

Then, deploy the container by entering the following command:

Bash

```bash
aws lightsail create-container-service-deployment \
    --region eu-west-3                            \
    --cli-input-json file://lc.json

```

Container ?

Test your deployment

To test your deployment, first retrieve the URL Lightsail created for you. Open a terminal and enter the following command:

Bash

```
aws lightsail get-container-services        \
    --region eu-west-3                      \
    --query "containerServices[].url"

# the command outputs something like the below
[
    "https://signup-application.me04fvc6dbk4e.eu-west-3.cs.amazonlightsail.com/"
]
```

Copy