

GETTING TO KNOW THE **AMAZING** AMAZON WEB SERVICES

Luigi Libero Lucio Starace

`luigi.starace@gmail.com`

June 4, 2018

University of Naples, Federico II

1/187

OUTLINE

- 1 A little bit of context
- 2 An AWS bestiary
- 3 Serverless architectures
- 4 Demo: a serverless web application
- 5 Take Home Messages

2/187

A LITTLE BIT OF CONTEXT

CLOUD COMPUTING

Cloud computing is the **on-demand delivery** of computing resources through a cloud services platform via the internet with **pay-as-you-go** pricing.

■ Software as a Service (SaaS)

The service vendor provides the user with a completed product that is run and managed by the service provider.

■ Platform as a Service (PaaS)

The service vendor provides the user with a set of API which can be used to build, test and deploy applications.

■ Infrastructure as a Service (IaaS)

The service vendor provides users access to computing resources such as servers, storage and networking.

4/187

SERVICE MODELS: A VISUAL COMPARISON

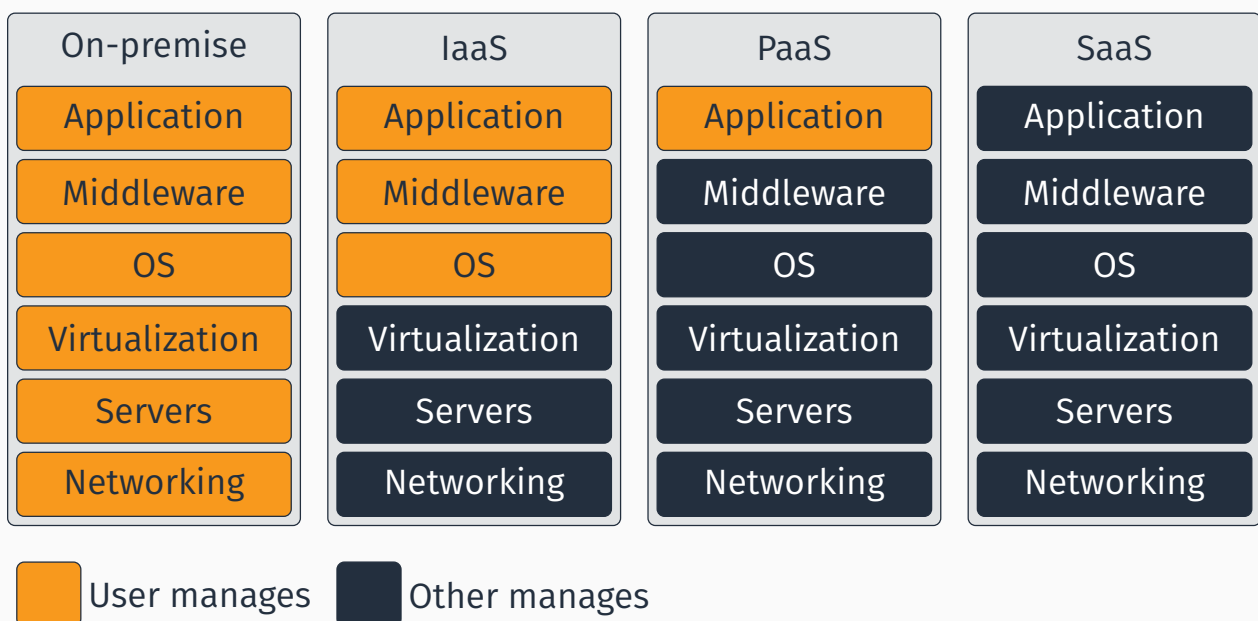


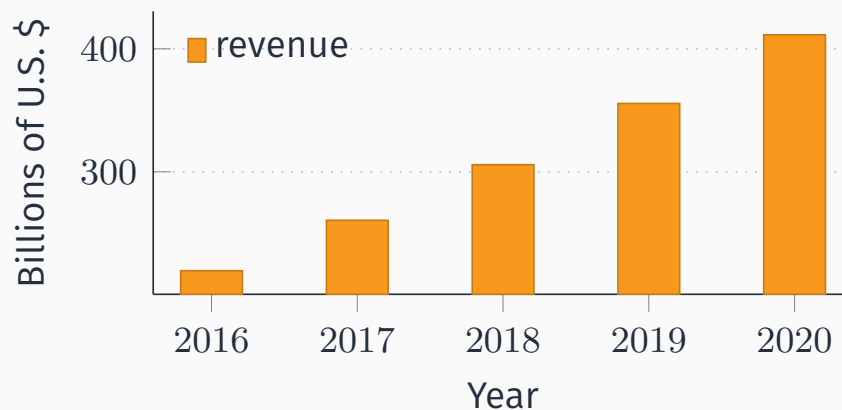
Figure 1: A service models comparison

5/187

SOME STATS

Worldwide Public Cloud Services Revenue Forecast (Billions of U.S. Dollars) [Gar17]

2016	2017	2018	2019	2020
219,6	260,6	305,8	355,6	411,4



6/187

THE BIGWIGS

- Google
- IBM
- Microsoft
- Alibaba
- Oracle
- Amazon



Google Cloud



7/187

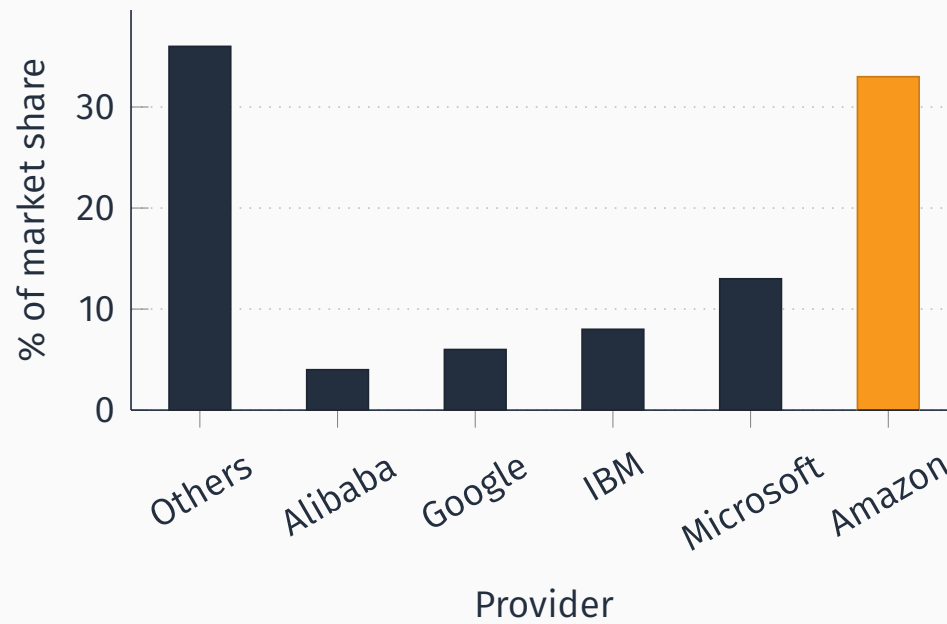


Figure 2: Market share in Q4 2017 (IaaS, PaaS, Hosted Private Cloud)
[Syn18]



Amazon Web Services is a collection of cloud-based services.
A very big one.

9/187

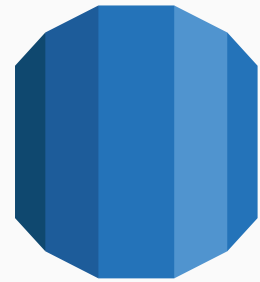
AN AWS BESTIARIUM

DATABASE SERVICES

RELATIONAL DATABASE SERVICE (RDS)



- Set up, operate a relational database in the cloud.
- Takes care of backups, patching.
- Supports:
 - MySQL, PostgreSQL, MariaDB
 - Oracle, MS SQL Server
 - Amazon Aurora



NON RELATIONAL DATABASE SERVICES

- DynamoDB
 - *Fast and flexible NoSQL database service for any scale.*
- ElastiCache
 - In memory data store.
 - Supports memcached, Redis
- Neptune
 - Graph database service
 - Supports RDF, SPARQL, ...



AN AWS BESTIARIUM

CLOUD STORAGE

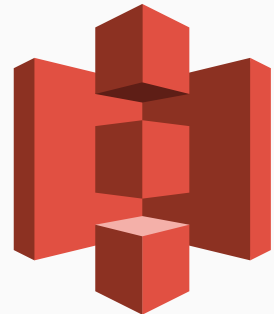
CLOUD STORAGE PRODUCTS

- Elastic Block Storage (EBS)
 - Persistent local storage for EC2 instances.
- Elastic File System (EFS)
 - File system interface to share data between EC2 instances.
- Simple Storage Service (S3)
- Glacier
 - Durable and cheap long-term storage.



AMAZON SIMPLE STORAGE SERVICE (S3)

- *store and retrieve any amount of data from anywhere*
- 99.999999999% durability (nine nines!)
- Data is distributed across a *minimum* of three availability zones
- A logical unit of storage is a *bucket*
- Multiple storage classes
 - Standard
 - Infrequent Access
 - One zone-Infrequent Access
 - Amazon Glacier



13/187

AMAZON SIMPLE STORAGE SERVICE (S3) - MORE

Multiple storage classes

Storage class	Storage	Retrieval (per 1K req.)
Standard	\$0.022 per GB	\$0.0004
Infrequent access	\$0.0125 per GB	\$0.001
IA single zone	\$0.01 per GB	\$0.001

Table 1: S3 pricing (Ireland)

14/187

- Well-integrated with other services
 - Machine Learning
 - Big Data Analysis
- REST API
- Can be used to host static websites



AN AWS BESTIARIUM

DEVELOPER TOOLS

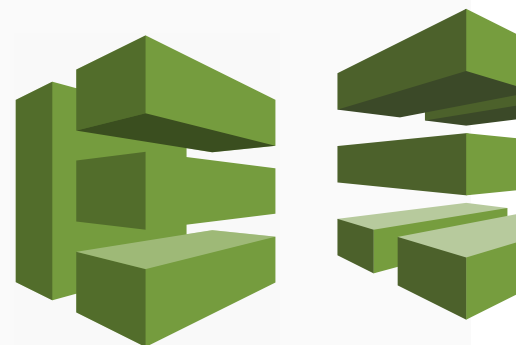
- CodeCommit
 - Managed, scalable, private git server
 - Pricing based on active users (5 free, 1\$ for each additional user)
- CodeBuild
 - Managed, scalable build server
 - Pay-per-minute spent building your code



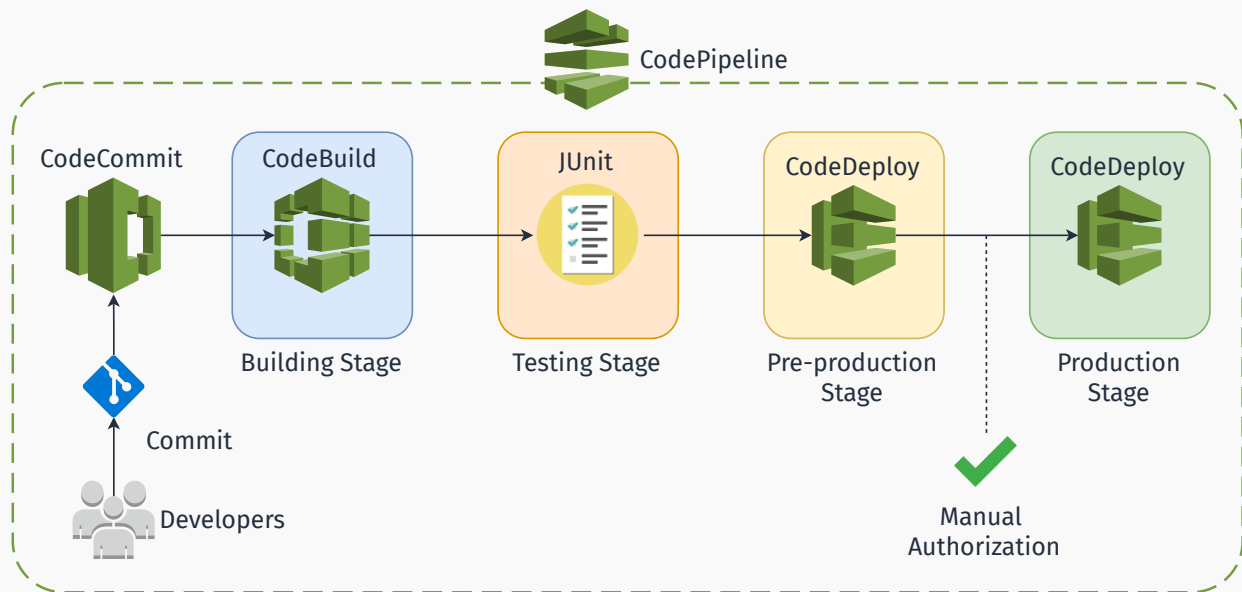
16/187

DEVELOPER TOOLS - MORE

- CodeDeploy
 - Automates deployment to computing services (also to instances running on-premise)
 - Tries to avoid downtime
 - 0.02\$ per-on-premise deployment
- CodePipeline
 - Continuous integration e continuous delivery
 - Define your own workflow and stages
 - 1\$ per-month per active pipeline



17/187



18/187

CODEPIPELINE - MORE

Interested in CI/CD on AWS? Check these out:

- *Practicing Continuous Integration and Continuous Delivery on AWS* (whitepaper) [[Ama17](#)]
- *Set up a Continuous Deployment Pipeline using AWS CodePipeline* [[Amab](#)]
- *Tutorial: Create a Four-Stage Pipeline* [[Amac](#)]

19/187

- CodeStar
 - Wrapper around developer tools to simplify setup
 - Templates
 - Team Management
 - Central Project Dashboard
 - Free of charge
- Cloud9
 - Cloud-based full-fledged IDE
 - Runs in a web browser
 - Collaborative editing and chat
 - Greatly-integrated with AWS
 - Free of charge

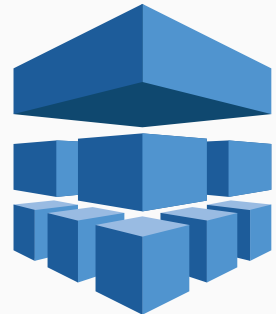


20/187

AN AWS BESTIARIUM

MACHINE LEARNING

- Amazon SageMaker
 - Preconfigured for Tensorflow, MXNet...
 - Build, Train and Deploy phases
 - Pay based on build time, train time and hosting time



21/187

- Comprehend (for NLP) [Dashboard](#)
- Rekognition (Visual Analysis) [Dashboard](#)
- Translate
- Polly (text-to-speech)
- Transcribe (speech-to-text)



22/187

AN AWS BESTIARIUM

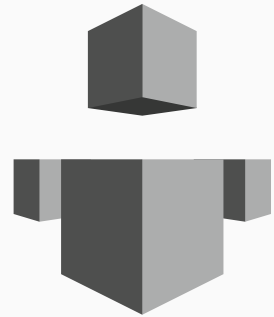
MISCELLANEA

MISCELLANEA

- Cognito
 - Sign-up and authentication
 - Federated identities
- CloudFront
 - Content Delivery Network
 - 116 Points of Presence in 56 cities across 24 countries
- Mechanical Turk



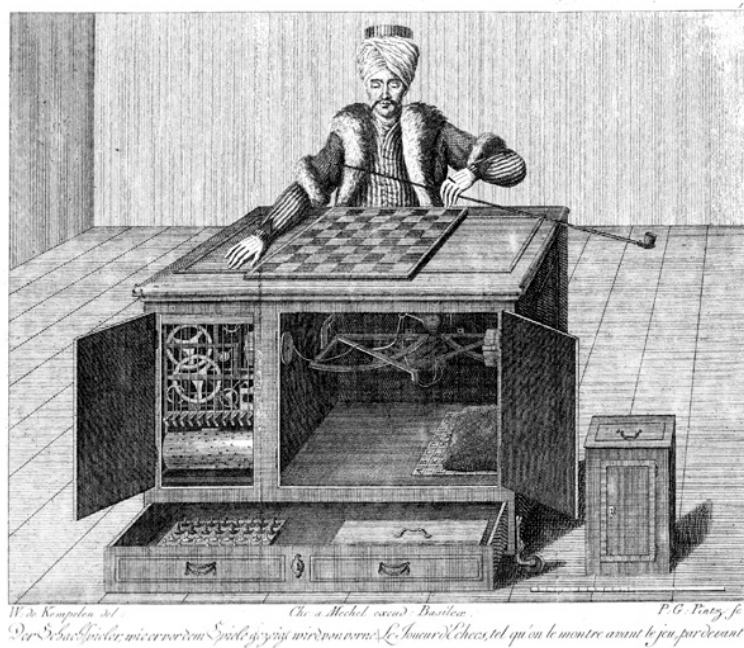
■ ???



24/187

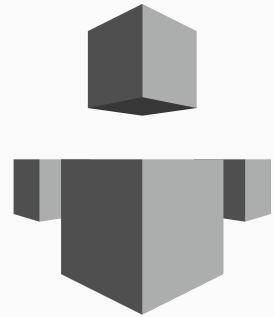
THE TURK

The Turk was a chess-playing automaton built in 1770. Obviously it was a fraud.



25/187


- ???
- Human Intelligence through an API
- Create HIT (Human Intelligence Task)
- Elastic, on-demand workforce
- Available 24/7



26/187

AN AWS BESTIARIUM

COMPUTING

- (Virtual) Servers on demand
- Different types of instances to suit computing needs
- Per-second (or per-hour) billing
- Data transfer **not** included!
- Persistent storage **not** included!
 -  EBS/EFS
- Scaling **not** included!



Azure: Virtual Machines 

Google Cloud: Compute Engine 

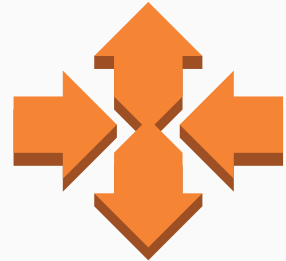
27/187

AMAZON ELASTIC COMPUTE CLOUD (EC2)

NOTES

- Need more space? Amazon Elastic Block Storage provides persistent block storage volumes for use with Amazon EC2 instances in the AWS Cloud

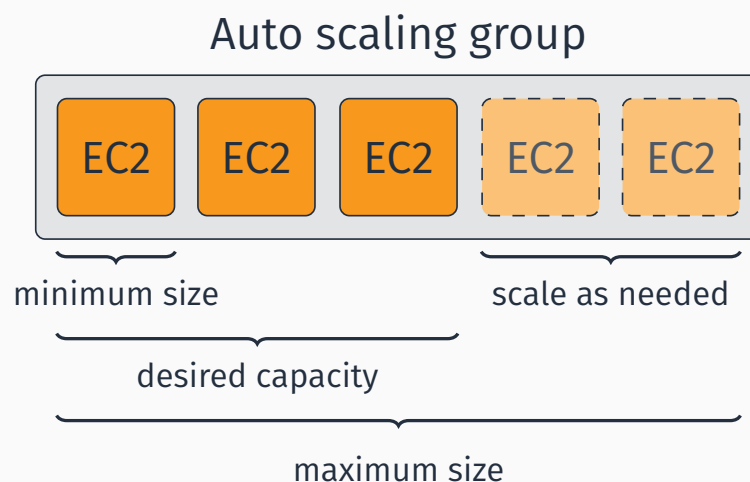
- *Scaling is the ability to increase or decrease the compute capacity of your application*
- Scale your application manually, on a scheduled basis or on demand



Azure: Virtual Machine Scale Sets [web](#)
Google Cloud: Load Balancing [web](#)

28/187

AMAZON EC2 AUTO SCALING: DETAILS



29/187

AMAZON ELASTIC LOAD BALANCING (ELB)

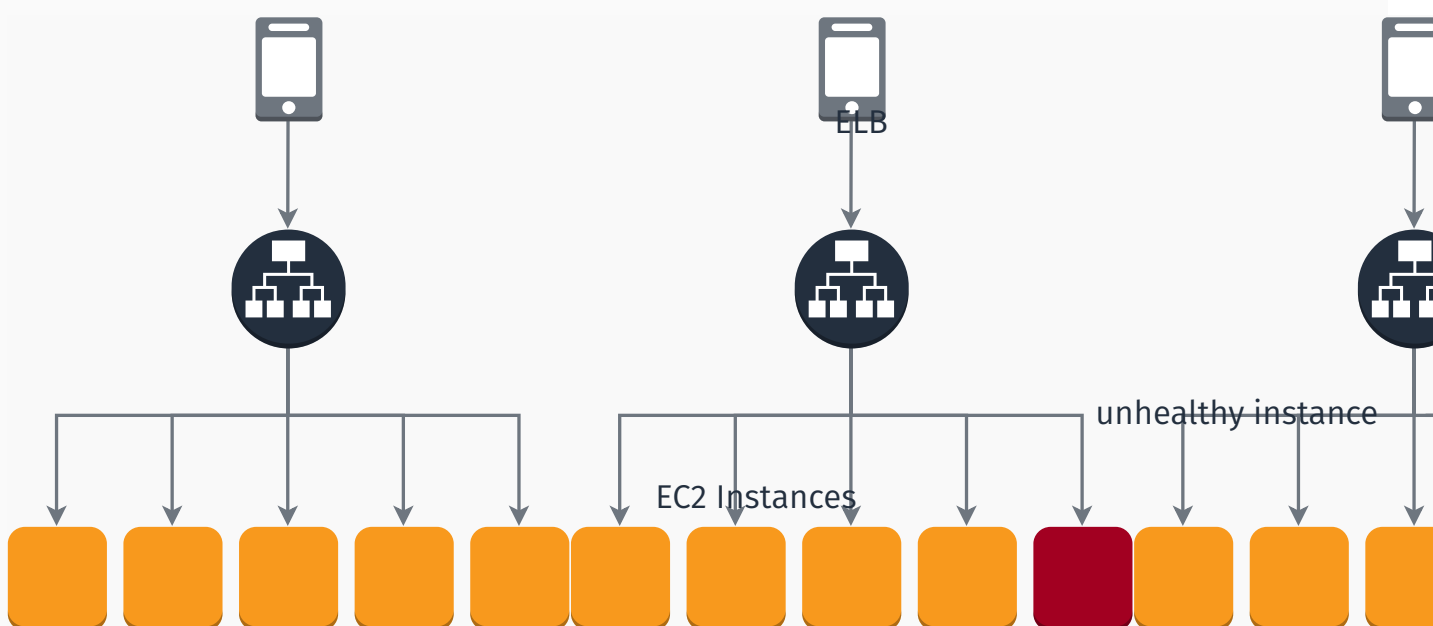
- Distributes incoming traffic across multiple EC2 instances
- Pay-per-use billing
 - Execution time
 - Number of requests / traffic



Azure: Load Balancer  web

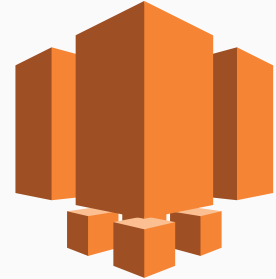
30/187

AMAZON ELASTIC LOAD BALANCING (ELB)



31/187

- A lightweight, simplified offer
- Bundles computing, storage, and networking capacity
- Preconfigured instances for
 - Debian, Windows Server, ...
 - Wordpress, Magento, Redmine, ...
 - LAMP stack, Nginx, ...
- Low and **predictable** monthly costs



Websites:  EC2  Lightsail

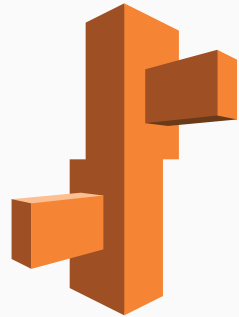
32/187

AMAZON LIGHTSAIL

NOTES

- Stress out how it is simplified
- Show the websites:  EC2  Lightsail
- Point out how it's not immediate to predict monthly cost for EC2 + Storage (Elastic Block Store) + Autoscaling + ELB

- “Easy to begin, impossible to outgrow”
- Easy-to-use service to deploy web apps
- Supports Apache, Nginx, IIS and more
- Supports Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker
- Manages auto-scaling, load balancing, health monitoring
- Customizable
- Free of charge. Pay only for the AWS resources you use.



33/187

A LITTLE RECAP

So far we've seen:

- Elastic Compute Cloud (EC2)
 - Auto-scaling, Elastic Load Balancing
- Lightsail
- Elastic Beanstalk

We have to (somewhat) care about the infrastructure!

34/187



It's demo time!

35/187

WHAT WE'RE GOING TO DO IN THIS DEMO

- Checkout a very simple web application written in PHP
- Run it locally (optional)
- Deploy it to the cloud using Amazon Elastic Beanstalk
- Doable in 30 minutes at home.

36/187

We'll deploy a very simple website for this very talk. The web app has two pages:

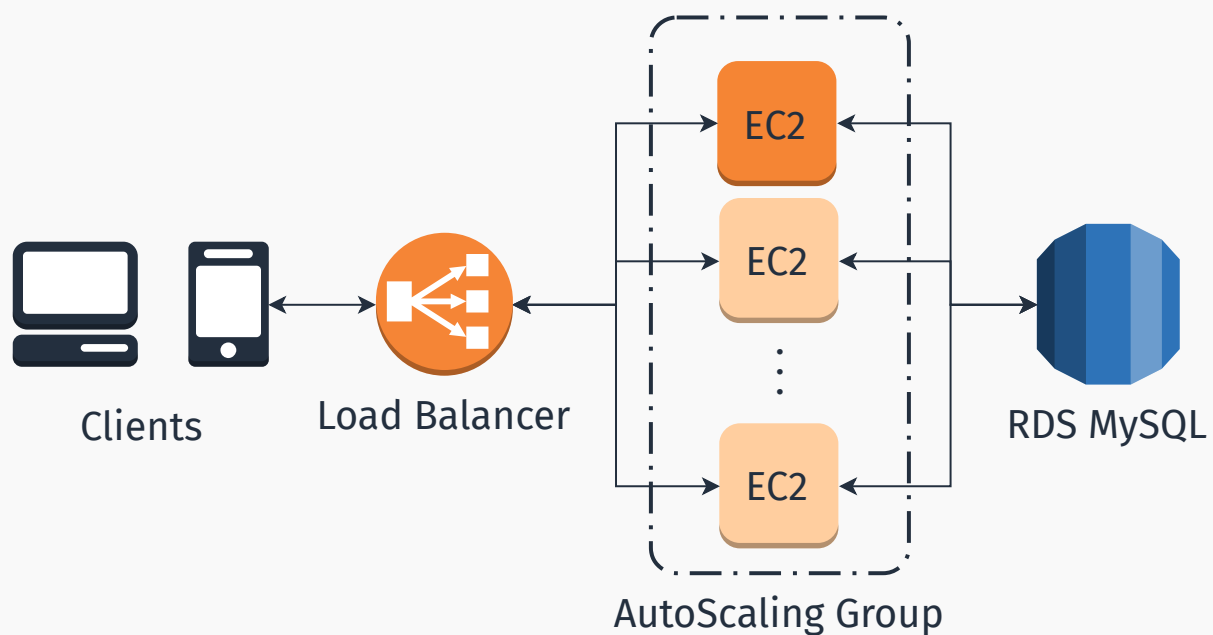
- a substantially static **homepage**
- a **comment** page allowing users to leave feedbacks.

Technologies involved:

- Symfony framework
- Doctrine ORM
- Webpack, Sass

37/187

ARCHITECTURE



- An AWS account (a free one will suffice)
- `git` version control (recommended)
- If you want to build and run the app locally:
 - An AMP (Apache, PHP \geq 7.1.3, MySQL \geq 5.7) stack
 - Composer package manager
 - Node.js

39/187

STEP 1: GET THE APP

Clone the git repository  Github

```
D:\Desktop> git clone https://github.com/luistar/
serverful-webapp.git serverful-webapp
```

40/187

STEP 2: INSTALL DEPENDENCIES

Install dependencies with composer

```
D:\Desktop> cd serverful-webapp
```

```
D:\Desktop\serverful-webapp> composer install
```

Then install Node.js dependencies

```
D:\Desktop\serverful-webapp> npm install
```

41/187

STEP 3: CONFIGURATION PARAMETERS

Start your database instance and create an user for the webapp. Once you are done, update the configuration file `config/packages/database-config.php` accordingly.

```
2 //get parameter from environment or fallback to defaults
3 $db_host = (
4     (isset($_SERVER['RDS_HOST'])) ?
5     ($_SERVER['RDS_HOST']) : ('localhost')
6 );
7 /* And following lines */
```

In `config/services.yaml` replace the dummy text with your Google Maps API Key.

```
1 parameters:
2     locale: 'en'
3     app.gmaps_api_key: '<YOUR GMAPS API KEY HERE>'
```

42/187

STEP 4: BUILD ASSETS AND CREATE DATABASE SCHEMA

Build assets with

```
D:\Desktop\serverful-webapp> npm run webpack-dev
```

Then create the database and the data schema by running

```
D:\Desktop\serverful-webapp> npm run drop-database  
D:\Desktop\serverful-webapp> npm run create-database  
D:\Desktop\serverful-webapp> npm run create-schema
```

43/187

STEP 5: RUN THE APP

Now you can start the dev server and check out the app.

```
D:\Desktop\serverful-webapp> npm run serve
```

Once the server started, visit the webapp at localhost:8000

44/187

STEP 6: CREATE A SOURCE BUNDLE

Elastic Beanstalk requires a single WAR or ZIP archive containing your app. To create a source bundle for our app, run

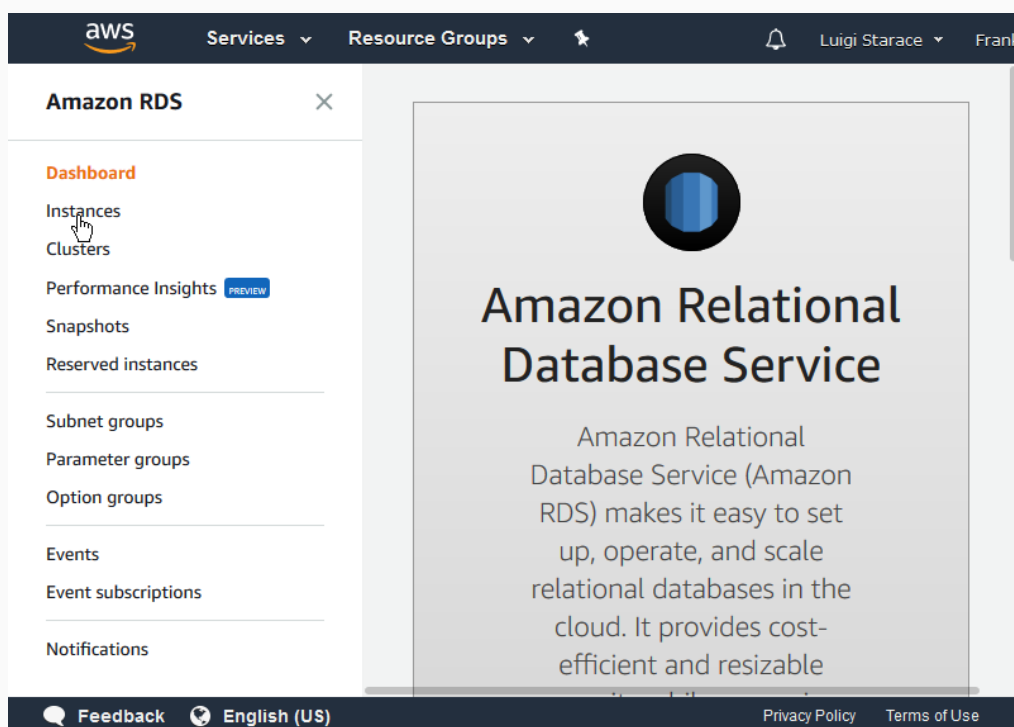
```
D:\Desktop\serverful-webapp> npm run create-source-bundle
```

A `serverful-app.zip` (our source bundle) archive will be created in the app root.

45/187

STEP 7: CREATE A DATABASE INSTANCE

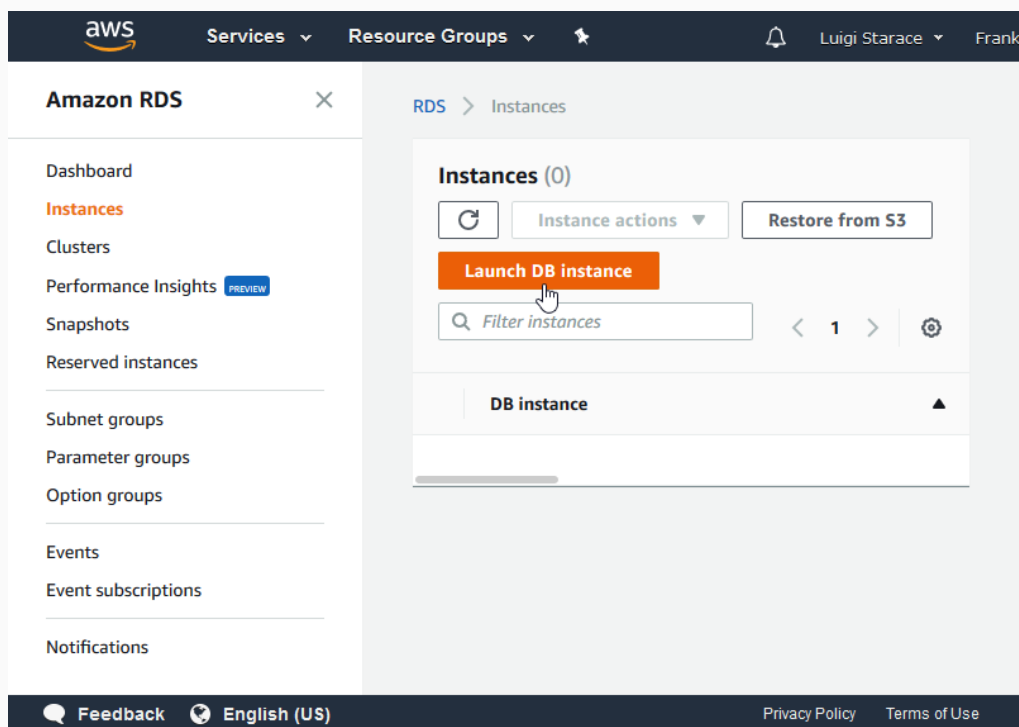
Go to the RDS Console and select “instances” .



46/187

STEP 7: CREATE A DATABASE INSTANCE

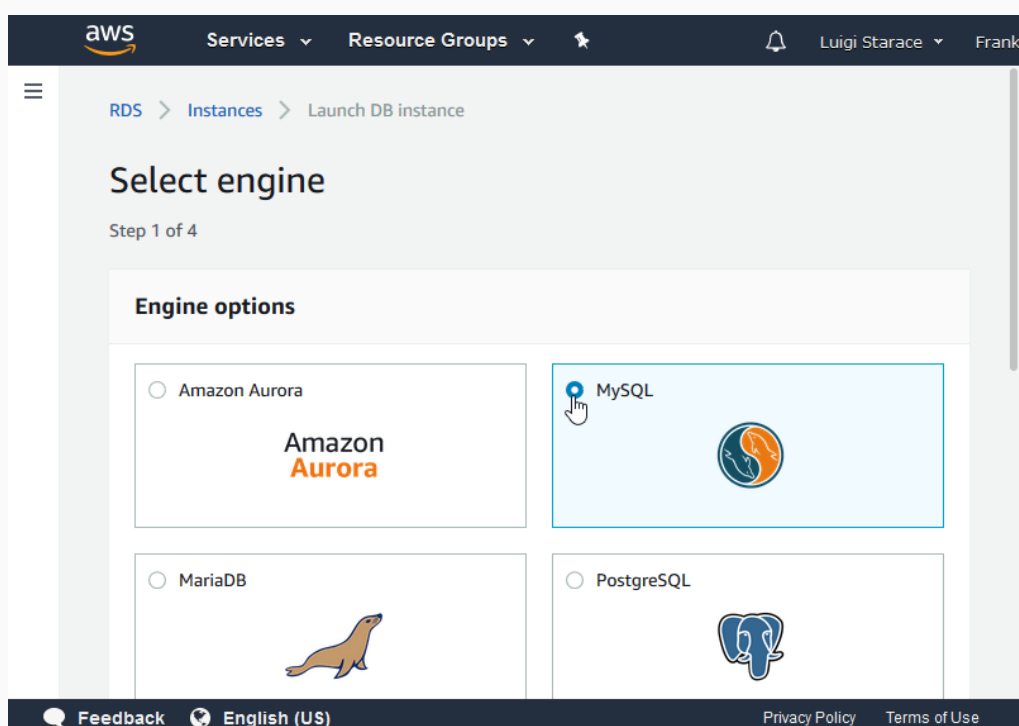
Select “Launch DB instance” .



47/187

STEP 7: CREATE A DATABASE INSTANCE

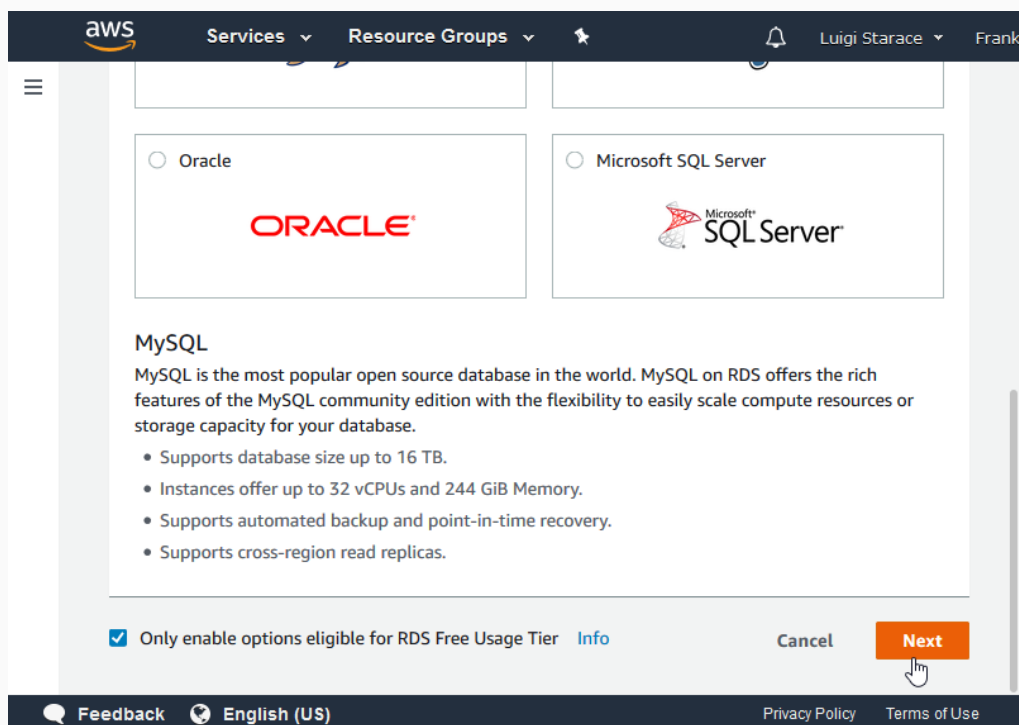
Select MySQL DBMS.



48/187

STEP 7: CREATE A DATABASE INSTANCE

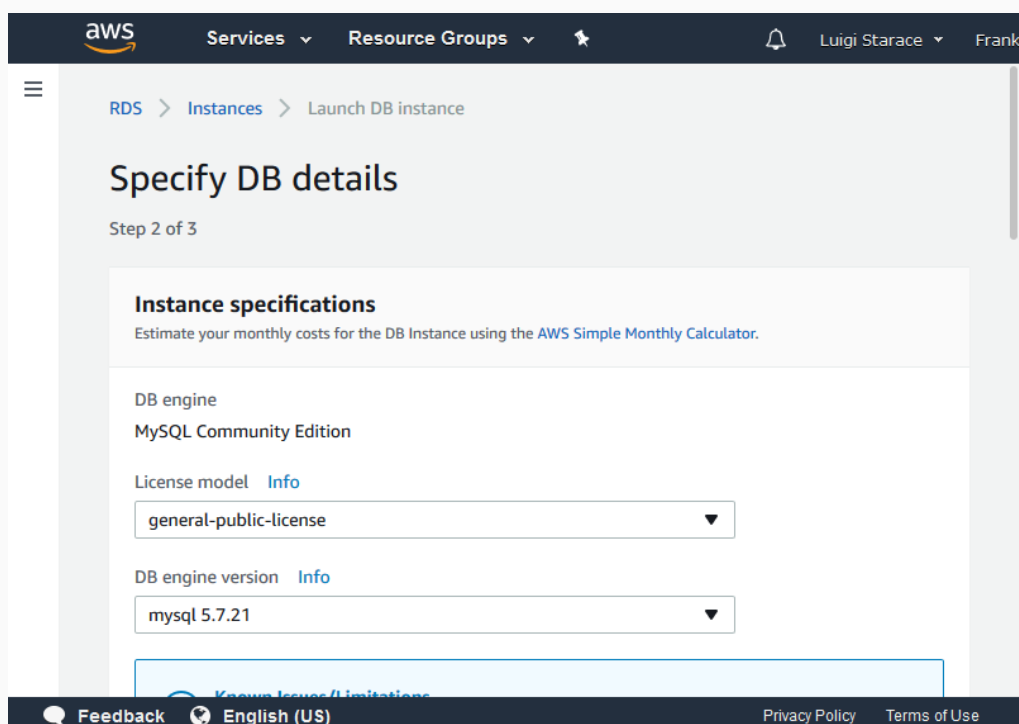
Enable only free-tier options and continue.



49/187

STEP 7: CREATE A DATABASE INSTANCE

Select MySQL version 5.7.21



50/187

STEP 7: CREATE A DATABASE INSTANCE

Select db.t2.micro instance.

The screenshot shows the AWS Management Console interface for creating a new RDS database instance. At the top, the 'Free tier' section is highlighted with a blue box and a circled '1'. It states: 'The Amazon RDS Free Tier provides a single db.t2.micro instance as well as up to 20 GB of storage, allowing new AWS customers to gain hands-on experience with Amazon RDS. Learn more about the RDS Free Tier and the instance restrictions [here](#).' Below this, there is a checkbox labeled 'Only enable options eligible for RDS Free Usage Tier' which is checked, and an 'Info' link. The 'DB instance class' dropdown is set to 'db.t2.micro — 1 vCPU, 1 GiB RAM'. The 'Multi-AZ deployment' section has two radio buttons: 'Create replica in different zone' (unselected) and 'No' (selected). The 'Storage type' dropdown is set to 'General Purpose (SSD)'. The 'Allocated storage' is set to '20 GB'. At the bottom, there is a note: '(Minimum: 20 GB, Maximum: 20 GB) Higher allocated storage [may improve](#) IOPS performance.'

51/187

STEP 7: CREATE A DATABASE INSTANCE

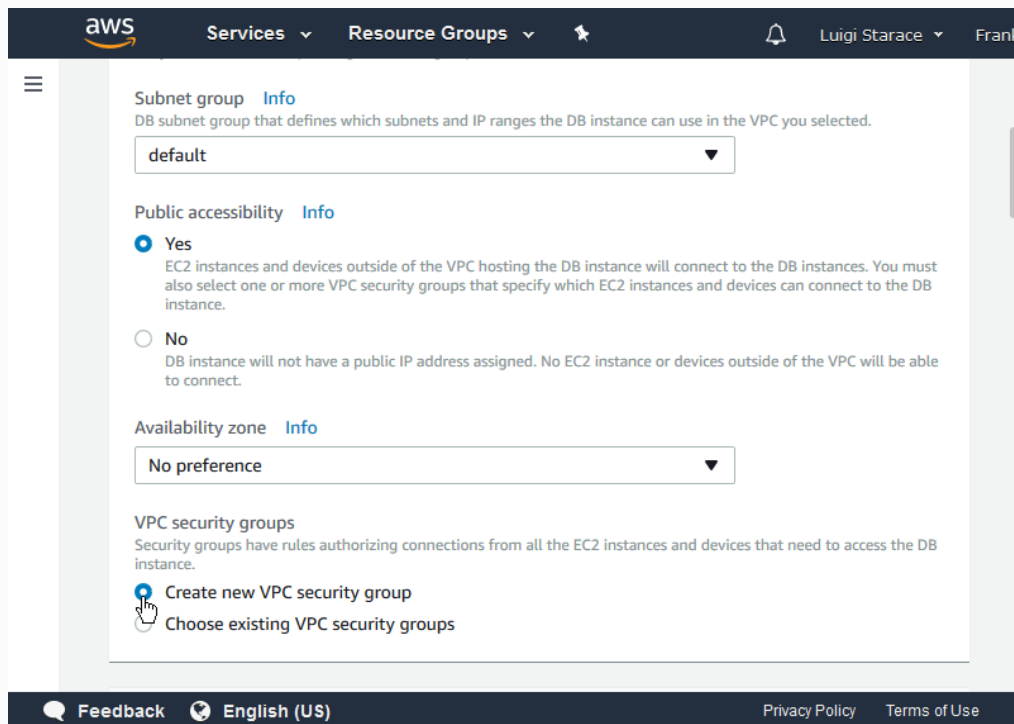
Enter your desired settings (remember the password! ⚠️).

The screenshot shows the 'Settings' section of the AWS RDS console. It contains three main input fields: 'DB instance identifier' with the value 'serverful-webapp-db', 'Master username' with the value 'serverfulwebapp', and 'Master password' with a masked password. Below the password field is a 'Confirm password' field, also with a masked password. Each field has an 'Info' link. The 'DB instance identifier' field has a detailed note: 'DB instance identifier is case insensitive, but stored as all lower-case, as in "mydbinstance". Must contain from 1 to 63 alphanumeric characters or hyphens (1 to 15 for SQL Server). First character must be a letter. Cannot end with a hyphen or contain two consecutive hyphens.' The 'Master username' field has a note: 'Specify an alphanumeric string that defines the login ID for the master user. Master Username must start with a letter. Must contain 1 to 16 alphanumeric characters.' The 'Master password' field has a note: 'Master Password must be at least eight characters long, as in "mypassword". Can be any printable ASCII character except "/", "", or "@'.' At the bottom right, there are three buttons: 'Cancel', 'Previous', and 'Next'. The 'Next' button is highlighted in orange and has a mouse cursor pointing at it.

52/187

STEP 7: CREATE A DATABASE INSTANCE


Be sure to select “create a new security group” .

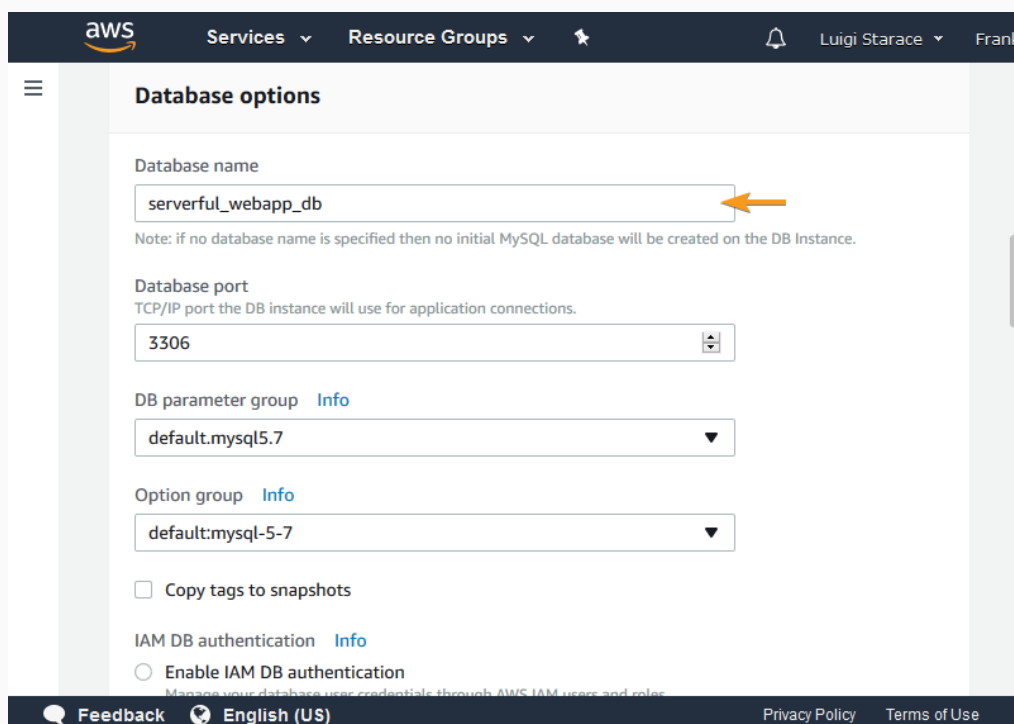


The screenshot shows the AWS console interface for configuring a Subnet group. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', and user information. The main content area is titled 'Subnet group' with an 'Info' link. Below the title is a description: 'DB subnet group that defines which subnets and IP ranges the DB instance can use in the VPC you selected.' A dropdown menu shows 'default'. The 'Public accessibility' section has two radio buttons: 'Yes' (selected) and 'No'. The 'Availability zone' section has a dropdown menu showing 'No preference'. The 'VPC security groups' section has a description and two radio buttons: 'Create new VPC security group' (selected) and 'Choose existing VPC security groups'. The bottom of the page has a footer with 'Feedback', 'English (US)', 'Privacy Policy', and 'Terms of Use'.

53/187

STEP 7: CREATE A DATABASE INSTANCE

Enter a database name for the instance (**important!** ) and leave the rest as is.

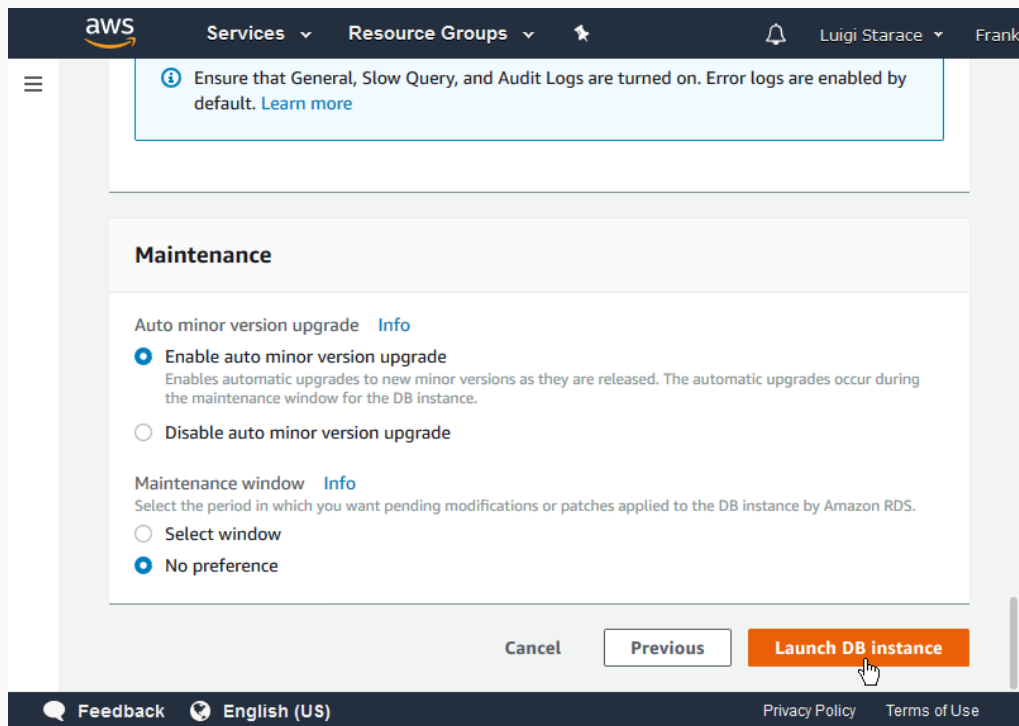


The screenshot shows the AWS console interface for configuring 'Database options'. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', and user information. The main content area is titled 'Database options'. The 'Database name' field contains 'serverful_webapp_db' and has an orange arrow pointing to it. Below the field is a note: 'Note: if no database name is specified then no initial MySQL database will be created on the DB Instance.' The 'Database port' section has a dropdown menu showing '3306'. The 'DB parameter group' section has a dropdown menu showing 'default.mysql5.7'. The 'Option group' section has a dropdown menu showing 'default.mysql-5-7'. The 'Copy tags to snapshots' checkbox is unchecked. The 'IAM DB authentication' section has a radio button for 'Enable IAM DB authentication'. The bottom of the page has a footer with 'Feedback', 'English (US)', 'Privacy Policy', and 'Terms of Use'.

54/187

STEP 7: CREATE A DATABASE INSTANCE

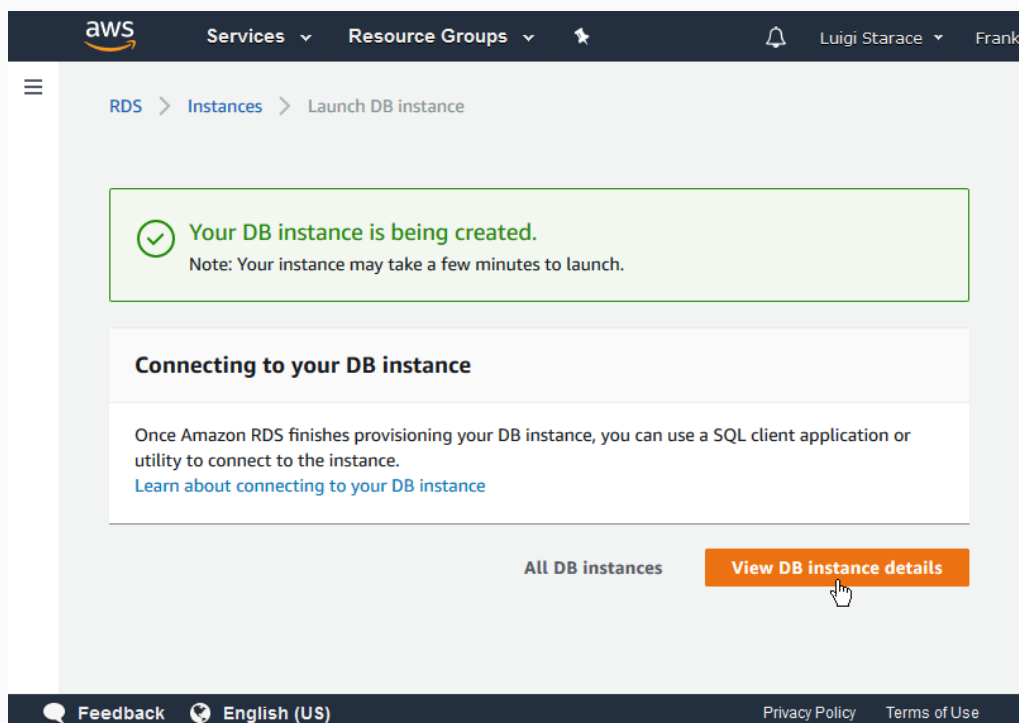
Click on “Launch DB Instance” .



55/187

STEP 7: CREATE A DATABASE INSTANCE

The creation process takes around 15 minutes. Click on “View DB Instance Details” to visit the detail page for the instance you just created.



56/187

STEP 7: CREATE A DATABASE INSTANCE

When done, the status in your instance detail page will change to “available”.

Resource Groups ▼ ★ Luigi Starace ▼ Frankfurt ▼ Support ▼

RDS > Instances > serverful-webapp-db

serverful-webapp-db

Instance actions ▼

Summary

Engine MySQL 5.7.21	DB instance class db.t2.micro	DB instance status available	Pending maintenance none
------------------------	----------------------------------	--	-----------------------------

CloudWatch (54)

🔄 Add instance to compare Monitoring ▼ Last Hour ▼

Legend: **serverful-webapp-db**

< 1 2 3 4 5 6 7 8 9 > ⚙️

© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

57/187

STEP 7: CREATE A DATABASE INSTANCE

Notice a few important elements in the details section. We're going to need these later.

Details Modify

Configurations

ARN
arn:aws:rds:eu-central-1:788880174327:db:serverful-webapp-db

Engine
MySQL 5.7.21

License Model
General Public License

Created Time
Sun Apr 15 08:40:55 GMT+200 2018

DB Name
serverful_webapp_db

Username
serverfulwebapp

Option Group
default:mysql-5-7

Parameter group
default:mysql5.7 (in-sync)

Security and network

Availability zone
eu-central-1c

VPC
vpc-12e77979

Subnet group
default

Subnets
subnet-e4184a8f
subnet-97274ada
subnet-f53fb788

Security groups
rds-launch-wizard-1 (sg-03a2d775170d52c34) (active)

Publicly accessible
Yes

Endpoint
serverful-webapp-db.civyafoewont.eu-central-1.rds.amazonaws.com

Instance and IOPS

Instance Class
db.t2.micro

Storage Type
General Purpose (SSD)

Storage
20 GB

Availability and durability

DB instance status
available

Multi AZ
No

Automated backups
Enabled (7 Days)

Latest restore time
April 15, 2018 at 8:45:00 AM UTC+2

Maintenance details

Auto minor version upgrade
Yes

Maintenance window
mon:01:28-mon:01:58 UTC (GMT)

Backup window
20:55-21:25 UTC (GMT)

Pending Modifications
None

Pending maintenance
none

Encryption details

Encryption enabled
No

58/187

STEP 7: CREATE A DATABASE INSTANCE

We'll need this instance to be accessible by our web application. To do so we're going to add a new rule to allow all instances in the same security group to access the database instance.

59/187

STEP 7: CREATE A DATABASE INSTANCE

Click on the security group in the section *Security Group Rules*.

Security group rules (2)

🔍 Filter security group rules

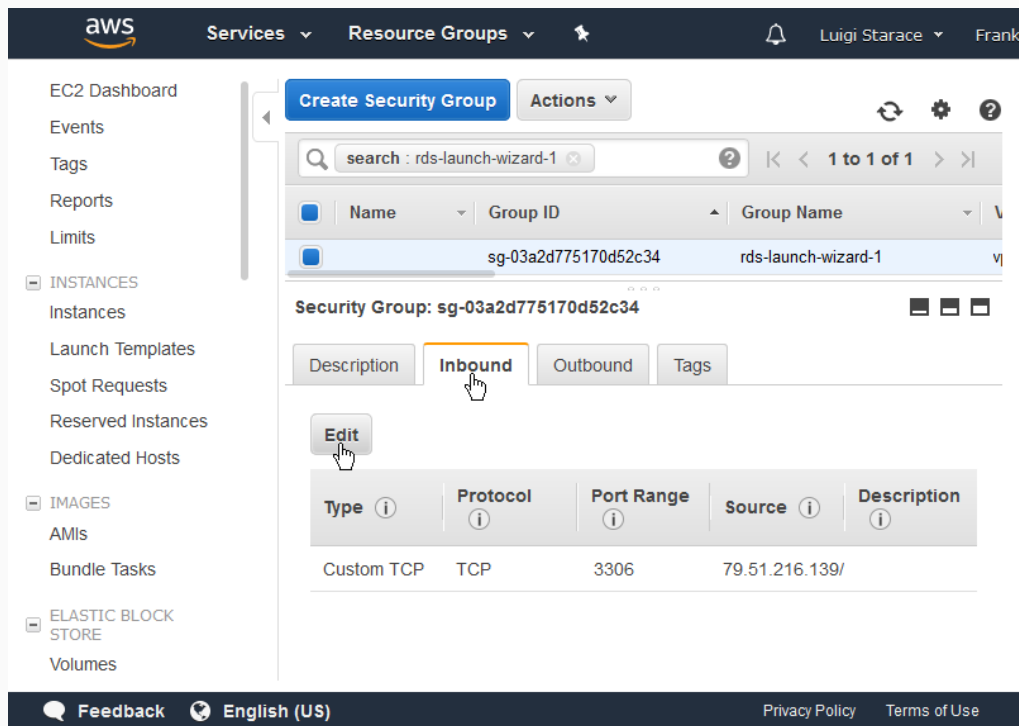
< 1 > ⚙️

Security group	Type	Rule
rds-launch-wizard-1 (sg-03a2d775170d52c34)	CIDR/IP - Inbound	79.51.216.139/32
rds-launch-wizard-1 (sg-03a2d775170d52c34)	CIDR/IP - Outbound	0.0.0.0/0

60/187

STEP 7: CREATE A DATABASE INSTANCE

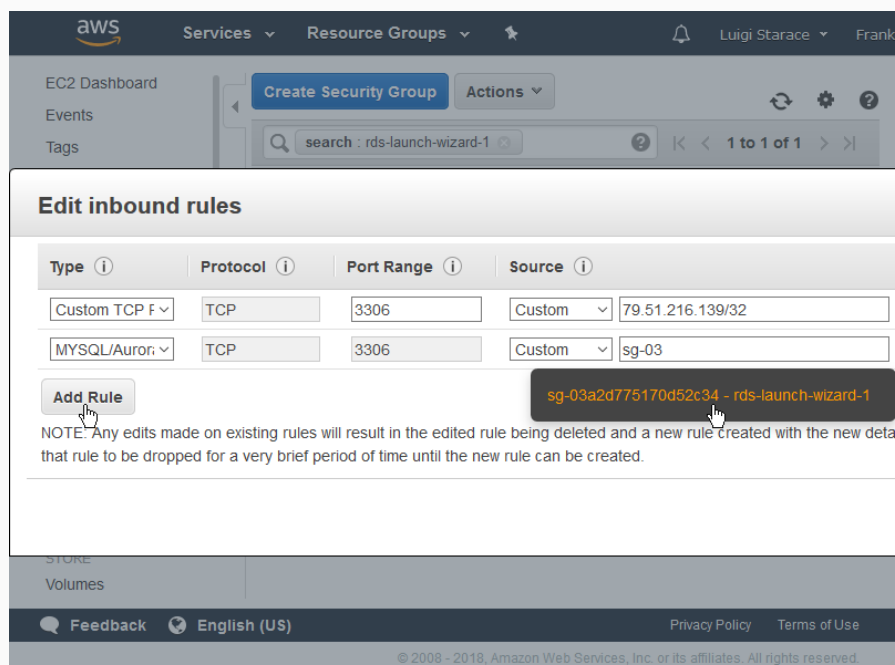
Select the *Inbound* tab then click on the Edit button.



61/187

STEP 7: CREATE A DATABASE INSTANCE

Add a new rule as shown in the picture. Be sure to select the same security group of the database instance. Then save and return to the RDS instance detail page.



62/187

STEP 7: CREATE A DATABASE INSTANCE


The rule you just added should be displayed among the other two.


Security group rules (3)		
<input type="text" value="Filter security group rules"/>		
< 1 > ⚙		
Security group	Type	Rule
rds-launch-wizard-1 (sg-03a2d775170d52c34)	CIDR/IP - Inbound	79.51.216.139/32
rds-launch-wizard-1 (sg-03a2d775170d52c34)	Security Group - Inbound	sg-03a2d775170d52c34
rds-launch-wizard-1 (sg-03a2d775170d52c34)	CIDR/IP - Outbound	0.0.0.0/0

63/187

STEP 7: CREATE A BEANSTALK APPLICATION

Go to the Beanstalk console and select *Create New Application*.

 Services ▾ Resource Groups ▾ ⚙

 Elastic Beanstalk

Create New Application

MobileBackend

Dashboard

Configuration

Logs

Monitoring

Alarms

Events

Overview

53.6
Average Latency in seconds

148K
Sum Requests

65%
CPU Utilization

354KB
Max network In

12KB
Maximum DiskReadBytes

Monitoring

Average Latency in seconds

Sum Requests by count

CPU Utilization in percent

Max Network In in bytes

Welcome to
AWS Elastic
Beanstalk

With Elastic Beanstalk, you can **deploy, monitor, and scale** an application quickly and easily. Let us do the heavy lifting so you can focus on your business.

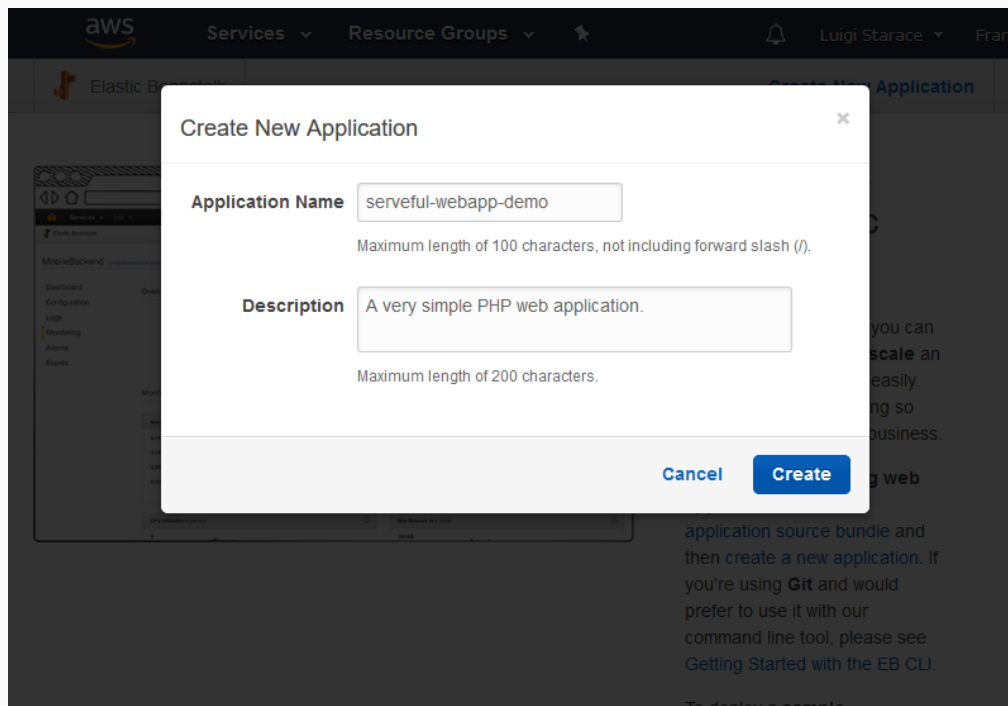
To deploy your **existing web application**, create an **application source bundle** and then **create a new application**. If you're using **Git** and would prefer to use it with our command line tool, please see [Getting Started with the EB CLI](#).

To deploy a **sample**

64/187

STEP 7: CREATE A BEANSTALK APPLICATION

Fill the form with your application information and continue.

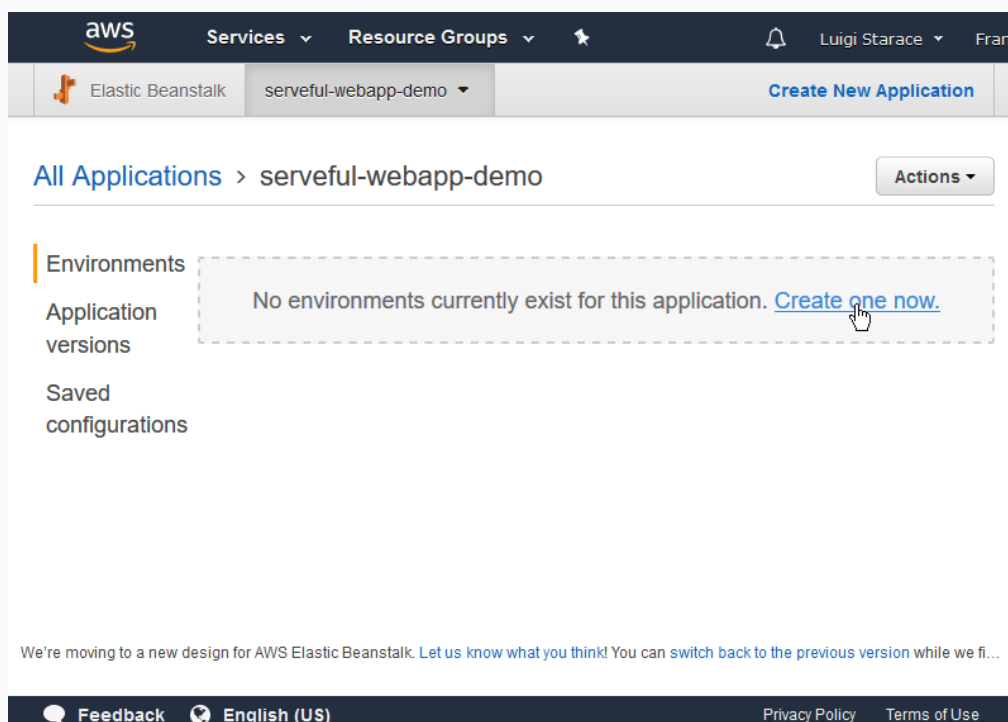


The screenshot shows the 'Create New Application' dialog box in the AWS console. The dialog has a title bar with a close button. It contains two input fields: 'Application Name' with the value 'serveful-webapp-demo' and a note 'Maximum length of 100 characters, not including forward slash (/).', and 'Description' with the value 'A very simple PHP web application.' and a note 'Maximum length of 200 characters.'. At the bottom right are 'Cancel' and 'Create' buttons.

65/187

STEP 7: CREATE A BEANSTALK APPLICATION

Then select *Create one now* to create a new environment for your application.

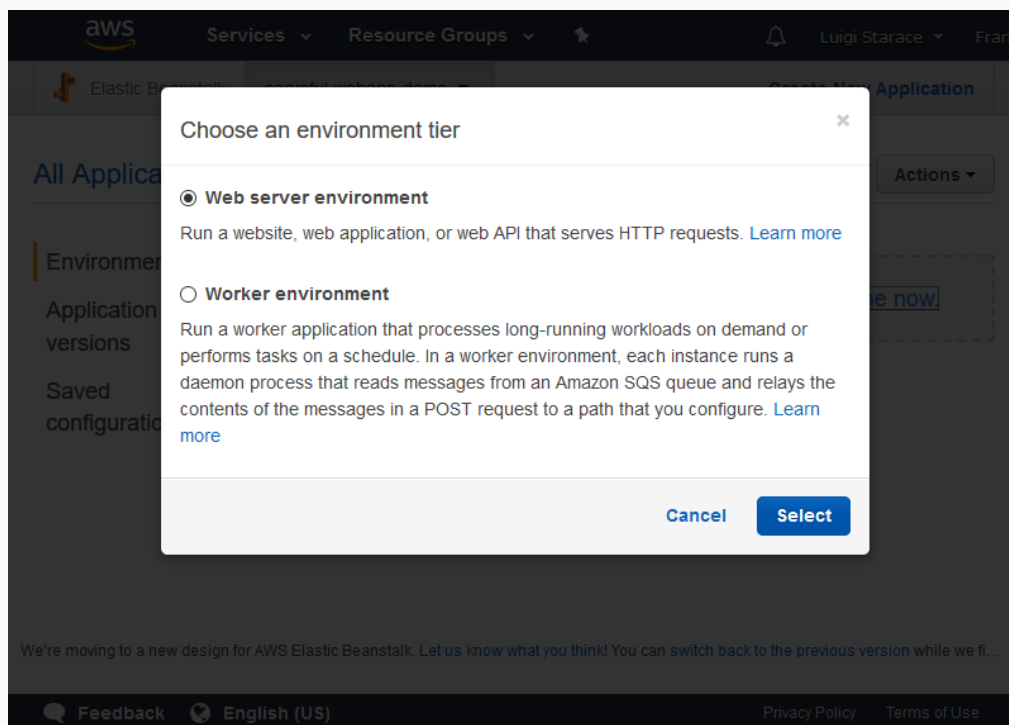


The screenshot shows the 'All Applications' page in the AWS console for the application 'serveful-webapp-demo'. The page has a header with the AWS logo, 'Services', 'Resource Groups', and user information. Below the header is a breadcrumb 'All Applications > serveful-webapp-demo' and an 'Actions' dropdown. The main content area has a sidebar with 'Environments', 'Application versions', and 'Saved configurations'. The 'Environments' section shows a message: 'No environments currently exist for this application. [Create one now.](#)' with a cursor pointing to the link. At the bottom, there is a footer with 'Feedback', 'English (US)', 'Privacy Policy', and 'Terms of Use'.

66/187

STEP 7: CREATE A BEANSTALK APPLICATION

Select *Web Server Environment*, as we are going to deploy a web application.



67/187

STEP 7: CREATE A BEANSTALK APPLICATION

Fill the form with information about your environment.

A screenshot of the "Create a new environment" form in the AWS Elastic Beanstalk console. The form has a title "Create a new environment" with a globe icon. Below the title is a paragraph explaining the purpose of creating an environment. The form is divided into sections: "Environment information" and "Domain". Under "Environment information", there is a text field for "Application name" with the value "serveful-webapp-demo" and a text field for "Environment name" with the value "ServefulWebappDemo-production". Under "Domain", there is a text field for "Domain" with the value "serveful-webapp" and a dropdown menu showing ".eu-central-1.elasticbeanstalk.com". Below these fields is a "Check availability" button. Below the button, a message states "serveful-webapp.eu-central-1.elasticbeanstalk.com is available." At the bottom, there is a text field for "Description" with the value "The production environment."

68/187

STEP 7: CREATE A BEANSTALK APPLICATION

Select PHP as preconfigured platform and upload the source bundle you previously prepared.

Tier Web Server ([Choose tier](#))

Platform ☒ Preconfigured platform
Platforms published and maintained by AWS Elastic Beanstalk.

→

☐ Custom platform ^{NEW}
Platforms created and owned by you. [Learn more](#)

Application code ☐ Sample application
Get started right away with sample code.

☐ Existing version
Application versions that you have uploaded for **serveful-webapp-demo**.

☒ Upload your code
Upload a source bundle from your computer or copy one from Amazon S3.

→ serveful-webapp-demo-source

69/187

STEP 7: CREATE A BEANSTALK APPLICATION

Select *Configure More Options* and continue.

Application code ☐ Sample application
Get started right away with sample code.

☐ Existing version
Application versions that you have uploaded for **serveful-webapp-demo**.

☒ Upload your code
Upload a source bundle from your computer or copy one from Amazon S3.

serveful-webapp-demo-source

[Cancel](#)

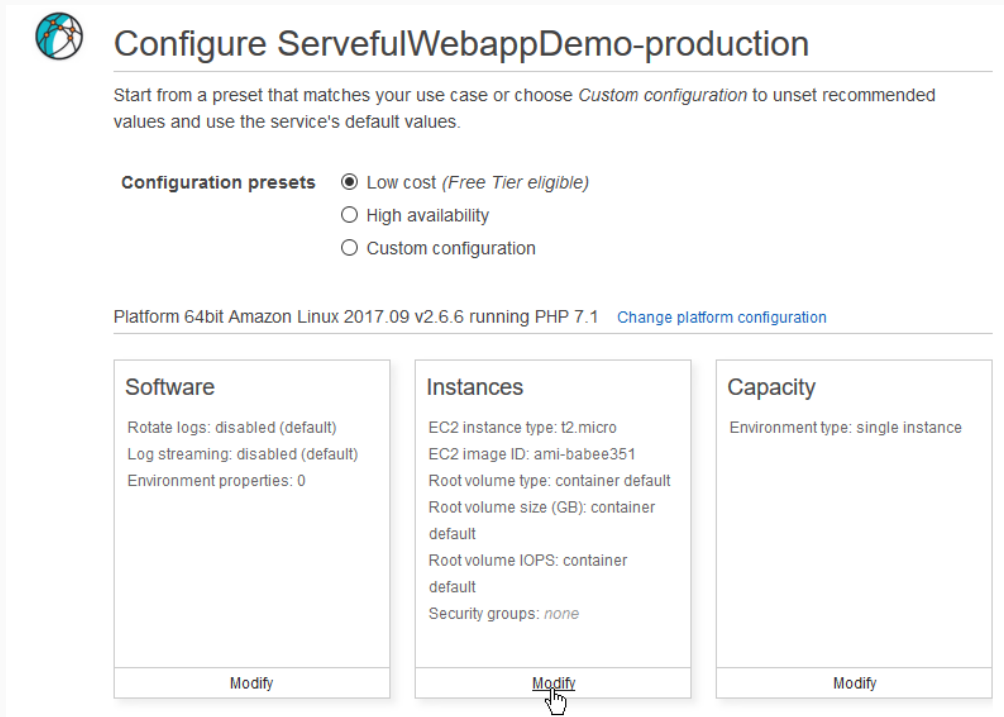
We're moving to a new design for AWS Elastic Beanstalk. [Let us know what you think!](#) You can [switch back to the previous version](#) while we fi...

[Feedback](#) [English \(US\)](#) [Privacy Policy](#) [Terms of Use](#)

70/187

STEP 7: CREATE A BEANSTALK APPLICATION

In the configuration page, select *modify* on the *Instances* card.



Configure ServefulWebappDemo-production

Start from a preset that matches your use case or choose *Custom configuration* to unset recommended values and use the service's default values.

Configuration presets

- ☒ Low cost (*Free Tier eligible*)
- ☐ High availability
- ☐ Custom configuration

Platform 64bit Amazon Linux 2017.09 v2.6.6 running PHP 7.1 [Change platform configuration](#)

Software
Rotate logs: disabled (default)
Log streaming: disabled (default)
Environment properties: 0
[Modify](#)

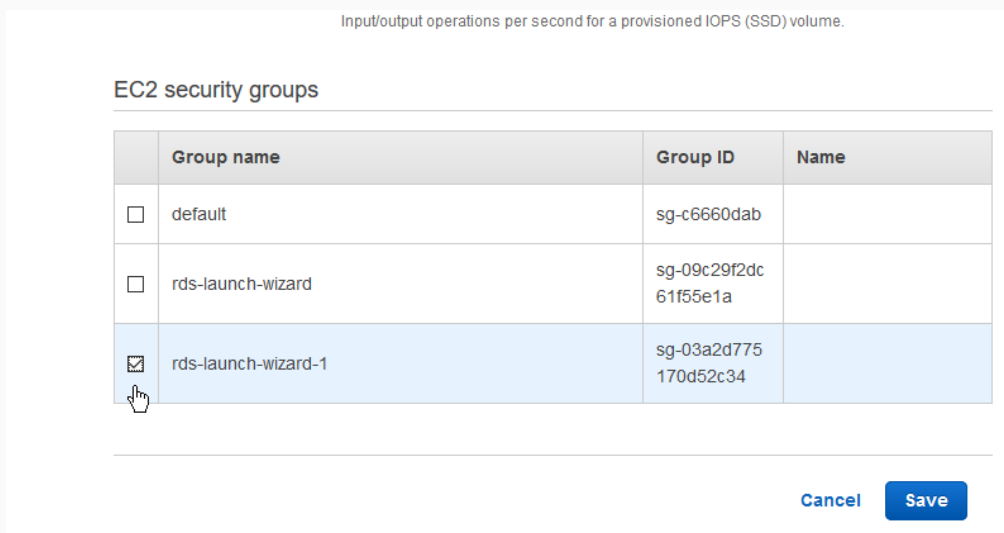
Instances
EC2 instance type: t2.micro
EC2 image ID: ami-babee351
Root volume type: container default
Root volume size (GB): container default
Root volume IOPS: container default
Security groups: none
[Modify](#)

Capacity
Environment type: single instance
[Modify](#)

71/187

STEP 7: CREATE A BEANSTALK APPLICATION

In the instances configuration page, add the t2 instance to the same security group as the DB instance. Then save and continue.



Input/output operations per second for a provisioned IOPS (SSD) volume.

EC2 security groups

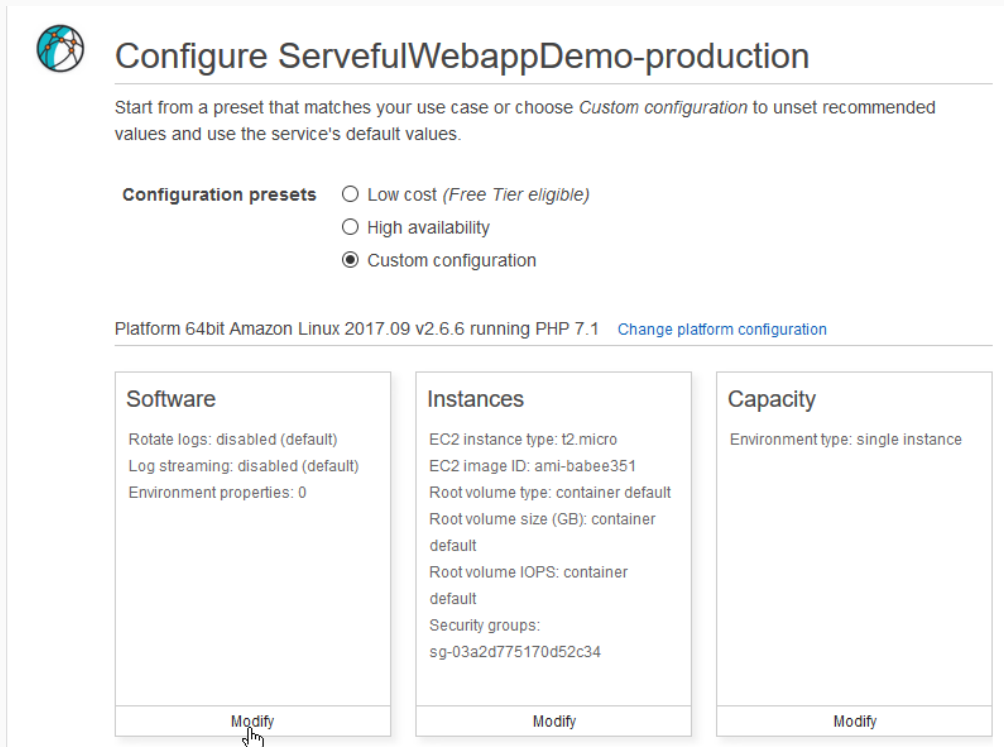
	Group name	Group ID	Name
<input type="checkbox"/>	default	sg-c6660dab	
<input type="checkbox"/>	rds-launch-wizard	sg-09c29f2dc61f55e1a	
<input checked="" type="checkbox"/>	rds-launch-wizard-1	sg-03a2d775170d52c34	

[Cancel](#) [Save](#)

72/187

STEP 7: CREATE A BEANSTALK APPLICATION

In the configuration page, select *modify* on the *Software* card.



Configure ServefulWebappDemo-production

Start from a preset that matches your use case or choose *Custom configuration* to unset recommended values and use the service's default values.

Configuration presets

- ☐ Low cost (Free Tier eligible)
- ☐ High availability
- ☒ Custom configuration

Platform 64bit Amazon Linux 2017.09 v2.6.6 running PHP 7.1 [Change platform configuration](#)

Software

Rotate logs: disabled (default)
Log streaming: disabled (default)
Environment properties: 0

Modify

Instances

EC2 instance type: t2.micro
EC2 image ID: ami-babee351
Root volume type: container default
Root volume size (GB): container default
Root volume IOPS: container default
Security groups:
sg-03a2d775170d52c34

Modify

Capacity

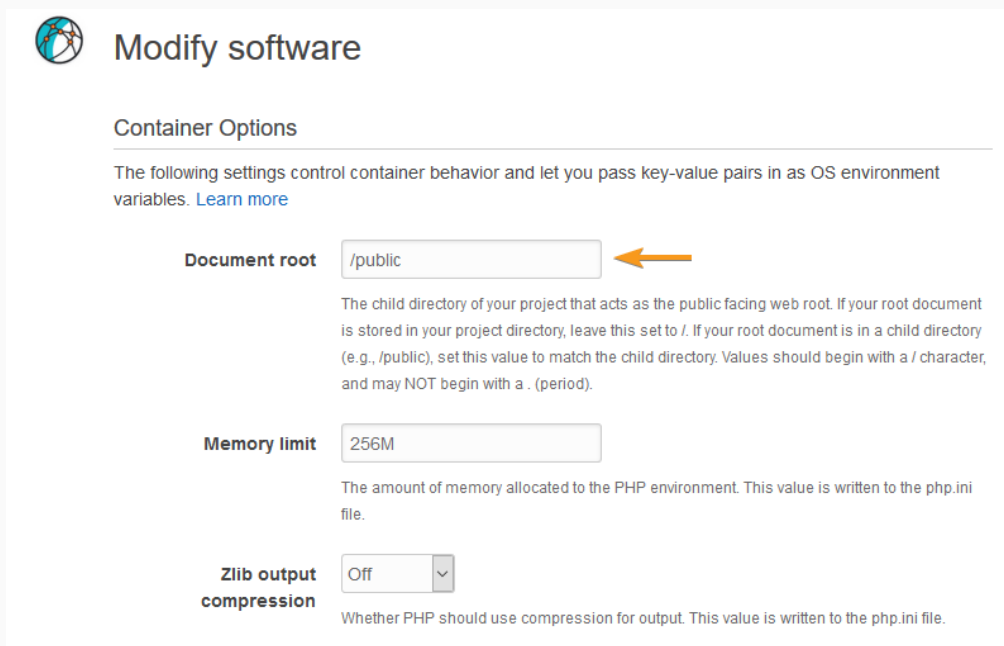
Environment type: single instance

Modify

73/187

STEP 7: CREATE A BEANSTALK APPLICATION


Enter `"/public"` as the document root and scroll down.



Modify software

Container Options

The following settings control container behavior and let you pass key-value pairs in as OS environment variables. [Learn more](#)

Document root 

The child directory of your project that acts as the public facing web root. If your root document is stored in your project directory, leave this set to /. If your root document is in a child directory (e.g., /public), set this value to match the child directory. Values should begin with a / character, and may NOT begin with a . (period).

Memory limit

The amount of memory allocated to the PHP environment. This value is written to the php.ini file.

Zlib output compression

Whether PHP should use compression for output. This value is written to the php.ini file.

74/187

STEP 7: CREATE A BEANSTALK APPLICATION

Enter the required environment parameters as shown in the picture. Be careful, deployment might fail if you mess up! ⚠

Environment properties

The following properties are passed in the application as environment properties. [Learn more](#)

Name	Value
<input type="text" value="APP_ENV"/>	<input type="text" value="prod"/> ✕
<input type="text" value="RDS_HOST"/>	<input type="text" value="-central-1.rds.amazonaws.com"/> ✕
<input type="text" value="RDS_NAME"/>	<input type="text" value="serverful_webapp_db"/> ✕
<input type="text" value="RDS_USER"/>	<input type="text" value="serverfulwebapp"/> ✕
<input type="text" value="RDS_PASSWORD"/>	<input type="password" value=""/> ✕
<input type="text" value="RDS_PORT"/>	<input type="text" value="3306"/> ✕
<input type="text"/>	<input type="text"/>

[Cancel](#) [Save](#)

75/187

STEP 7: CREATE A BEANSTALK APPLICATION

Click on *Create Environment* and continue.

Modify

Modify

Modify

Database
Engine: --
Instance class: --
Storage (GB): --
Multi-AZ: --
Modify

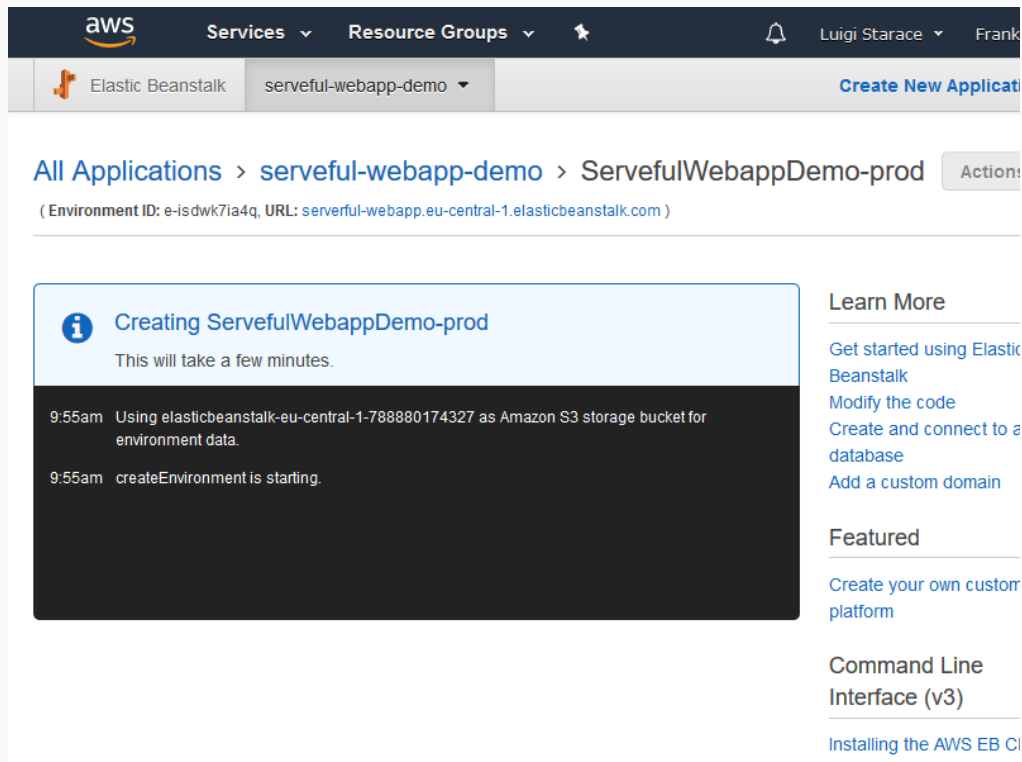
Tags
Tags: none
Modify

[Cancel](#) [Previous](#) [Create environment](#)

76/187

STEP 7: CREATE A BEANSTALK APPLICATION

Wait for the environment to be created. This takes about 10 minutes.



STEP 8: LOAD BALANCING

Right now we have our application running on a single (virtual) web server. That's not scaling at all. Let's take advantage of the cloud and make the web application load balanced.

STEP 8: LOAD BALANCING

Select the environment's configuration view, then select the Capacity card.

Configuration overview

Software Rotate logs: disabled (default) Log streaming: disabled (default) Environment properties: 6 Modify	Instances EC2 instance type: t2.micro EC2 image ID: ami-babee351 Monitoring interval: 5 minute Root volume type: container default Root volume size (GB): container default Root volume IOPS: container default Security groups: sg-03a2d775170d52c34, sg-03c173ae822a52c04 Modify	Capacity Environment type: single instance Modify
Load balancer <i>This configuration does not contain a load balancer.</i>	Rolling updates and deployments Deployment policy: All at once Rolling updates: disabled Health check: enabled Modify	Security Service role: aws-elasticbeanstalk-service-role Virtual machine key pair: -- Virtual machine instance profile: aws-elasticbeanstalk-ec2-role Modify

79/187

STEP 8: LOAD BALANCING

Select “Load balanced” as the environment type and customize the Auto Scaling Group.

Modify capacity

Auto Scaling Group

Configure the compute capacity of your environment and Auto Scaling settings to optimize the number of instances used.

Environment type

Instances Min Max

Availability Zones
Number of Availability Zones (AZs) to use.

Placement
eu-central-1b
eu-central-1c
Specify Availability Zones (AZs) to use.

Scaling cooldown seconds

80/187

STEP 8: LOAD BALANCING

Select some triggers (you can even setup time based ones), then save your changes.

Scaling triggers

Metric CPUUtilization Change the metric that is monitored to determine if the environment's capacity is too low or too high.

Statistic Average Choose how the metric is interpreted.

Unit Percent

Period 5 Min The period between metric evaluations.

Breach duration 5 Min The amount of time a metric can exceed a threshold before triggering a scaling operation.

Upper threshold 100 Percent

Scale up increment 1 EC2 instances

Lower threshold 90 Percent

Scale down increment -1 EC2 instances

81/187

STEP 9: ENJOY YOUR WEB APP

When it's done you should see something like this. Click on the URL to visit the load-balanced web application you just deployed on Beanstalk!

aws Services Resource Groups Luigi Starace Frankfurt Support


Elastic Beanstalk serveful-webapp-demo Create New Application

All Applications > serveful-webapp-demo > ServefulWebappDemo-prod (Environment ID: e-isdwk7ia4q, URL: serveful-webapp-eu-central-1.elasticbeanstalk.com) Actions


Dashboard Overview Refresh

Configuration

Logs

Health  **Health** **OK** Causes

Running Version serveful-webapp-demo-source Upload and Deploy

Configuration  PHP 7.1 running on 64bit Amazon Linux/2.5.6 Change

Alarms

Managed Updates

Events

Tags

Recent Events Show All

Time	Type	Details
2018-04-15 10:08:37 UTC+0200	INFO	createConfigurationTemplate completed successfully.
2018-04-15 10:08:36 UTC+0200	INFO	createConfigurationTemplate is starting.
2018-04-15 10:00:21 UTC+0200	INFO	Successfully launched environment: ServefulWebappDemo-prod
2018-04-15 09:59:55 UTC+0200	INFO	Environment health has transitioned from Pending to Ok. Initialization completed 29 seconds ago and took 3 minutes.
2018-04-15 09:57:55 UTC+0200	INFO	Added instance [i-0a2dacea5fe9f68e6] to your environment.

82/187

Sweet!



83/187

LET'S GET BACK TO COMPUTING SERVICES

- You provide the code and say when to run it.
- Execution is triggered by events
 - S3, Cognito, DynamoDB
 - CodeCommit, Scheduled Event
- Support for Java, Node.js, C# e Python (more to come).
- Pay only for **actual** execution time.
- Run your code without thinking about infrastructure
 - No need to worry about provisioning, load balancing, scaling...



84/187

- Named after anonymous (lambda) functions?
- Stress the *actual* execution time part.
- Give some examples of events triggering lambda execution
 - API Gateway call
 - File uploaded, record updated/added, cron events, ecc...

AWS Lambda imposes some limits

- Max 300 seconds execution time.
- Max 3008 MB memory allocation.
- Deployment package must be smaller than 50 MB (negotiable).
- No more than 10000 concurrent invocation of a Lambda function in a given region (negotiable).
- For a complete list: [📄 Lambda docs](#)



85/187

AMAZON LAMBDA: FAAS

FaaS (Functions as a Service)

- Functions are the unit of deployment
- Executed in ephemeral, stateless containers
- Event driven
- No provisioning, scales automatically
- Azure: Functions [▶ web](#)
- Google Cloud: Functions [▶ web](#)
- IBM: Cloud Functions [▶ web](#)
 - Based on Apache OpenWhisk [▶ web](#)

86/187

- Orchestrating Lambda functions
- Define a state machine



87/187

AMAZON STEP FUNCTIONS: SAMPLE I

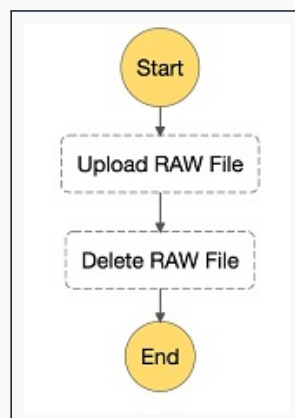


Figure 3: Sequential steps, from [AWS]

88/187

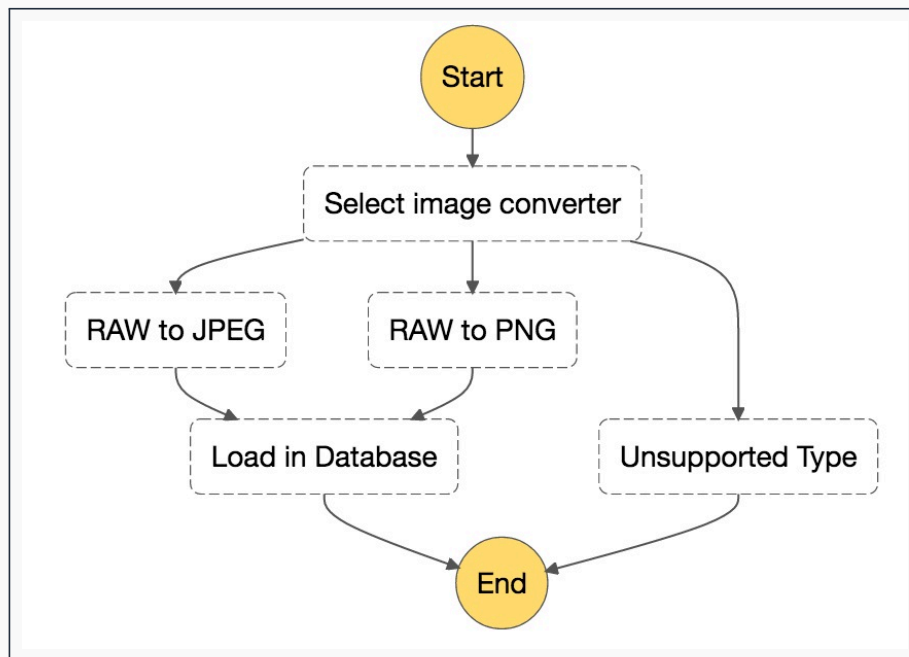


Figure 4: Branching, from [AWS]

89/187

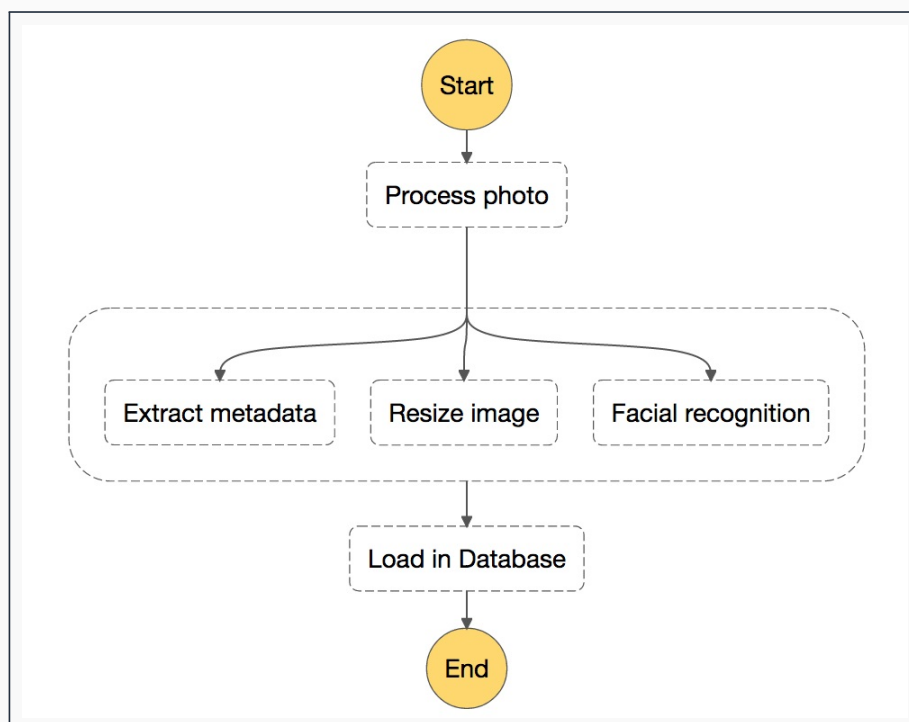


Figure 5: Parallel execution, from [AWS]

90/187

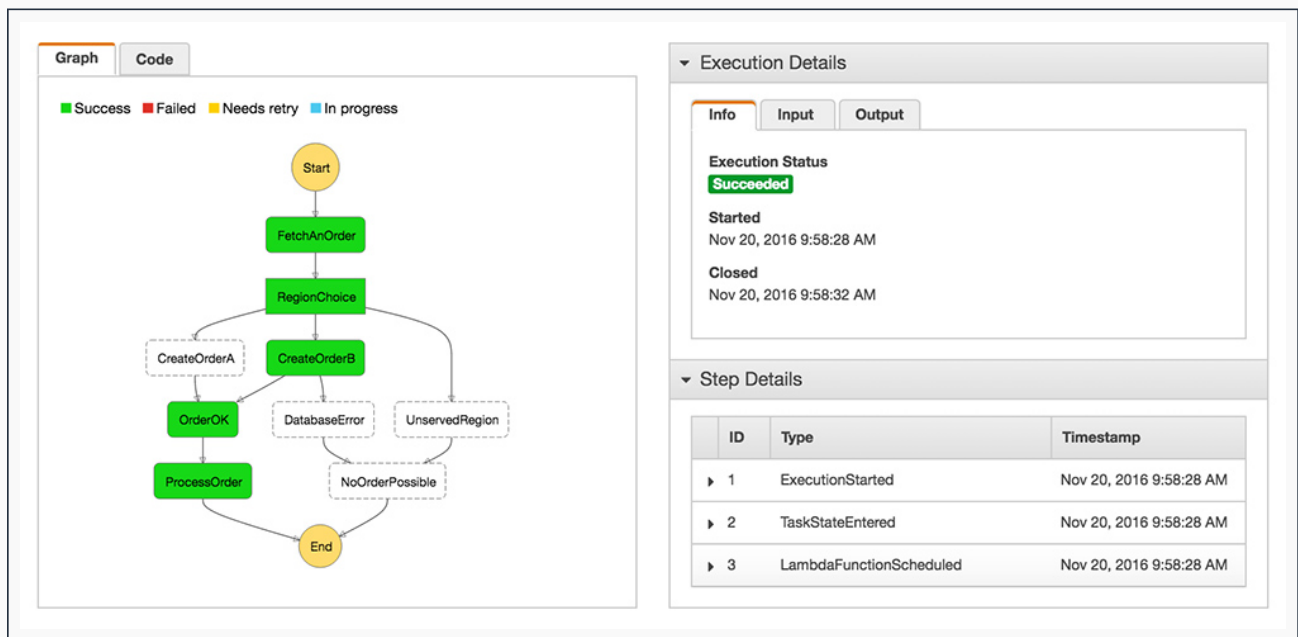


Figure 6: Monitoring executions, from [AWS]

91/187

AMAZON API GATEWAY

- Create, publish, and secure APIs at any scale
- Authorizers (Cognito)



92/187

Serverless?

What's all the FaaS about?

93/187

SERVERLESS TREND

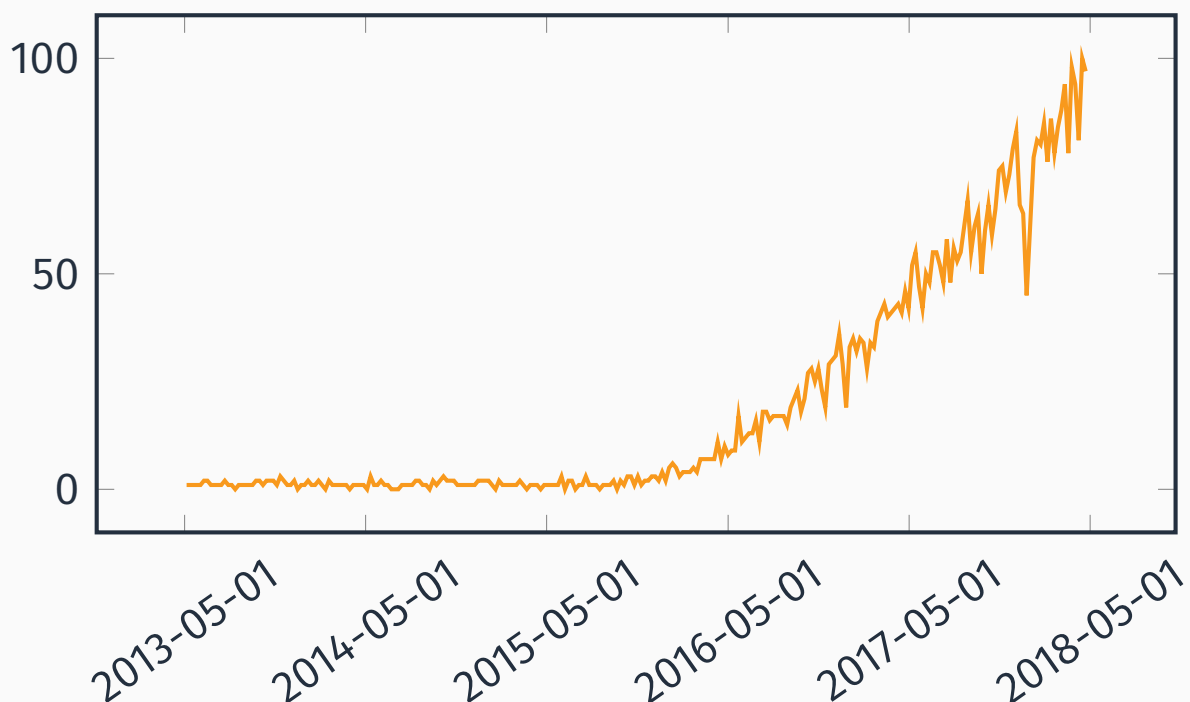


Figure 7: Last five years trend on Google for “serverless”

94/187

- No, they're not *actually* serverless...
- Rely on FaaS and third-party services so that traditional always-on servers are no longer needed
- No worries about provisioning and scaling
- “Smarter” clients

95/187

SERVERLESS USE CASES: SPORADIC REQUESTS

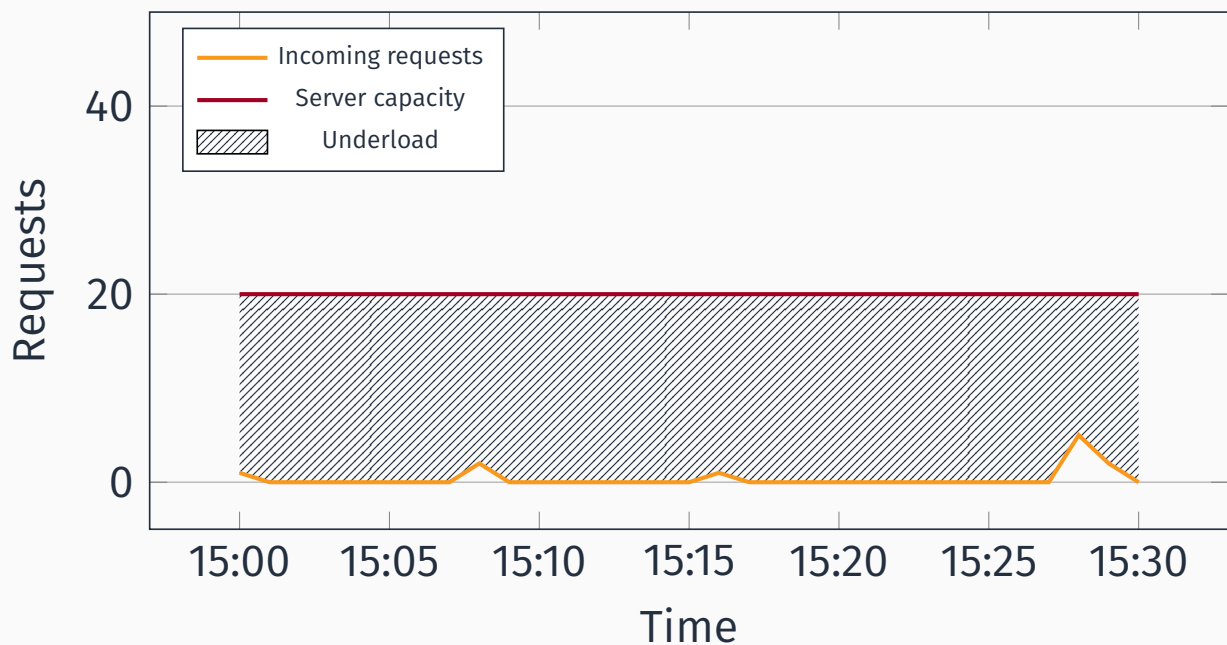


Figure 8: Sporadic requests example

96/187

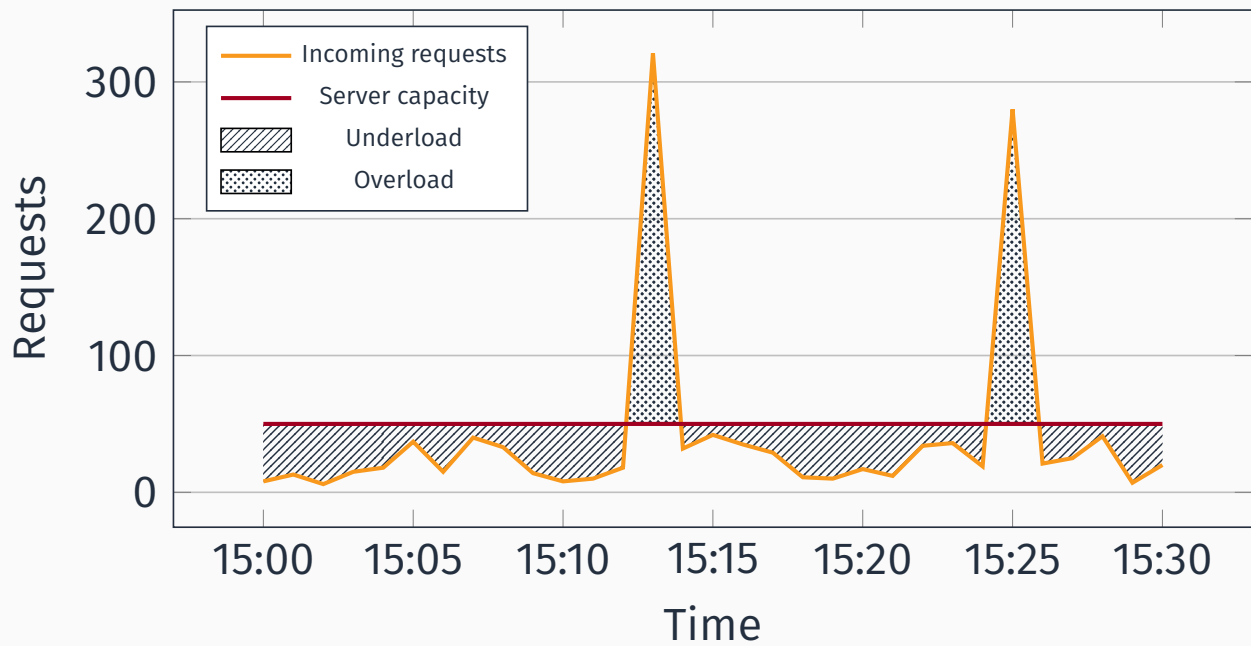



Figure 9: Inconsistent requests example

97/187

SERVERLESS ARCHITECTURES: TRADE-OFFS




Pros

- Reduce costs 
- No worries about provisioning, scaling
- Less time to market

Cons

- Limits
- Vendor lock-in
- Testing

98/187

- Serverless Framework 
 - “toolkit to deploy and operate serverless architecture” .
 - Works with AWS, Google, Microsoft, IBM.
- APEX 
- AWS SAM 
 - Serverless Application Model
 - “Define serverless applications with a simple and clean syntax”
 - SAM Local: CLI tool for local development and testing



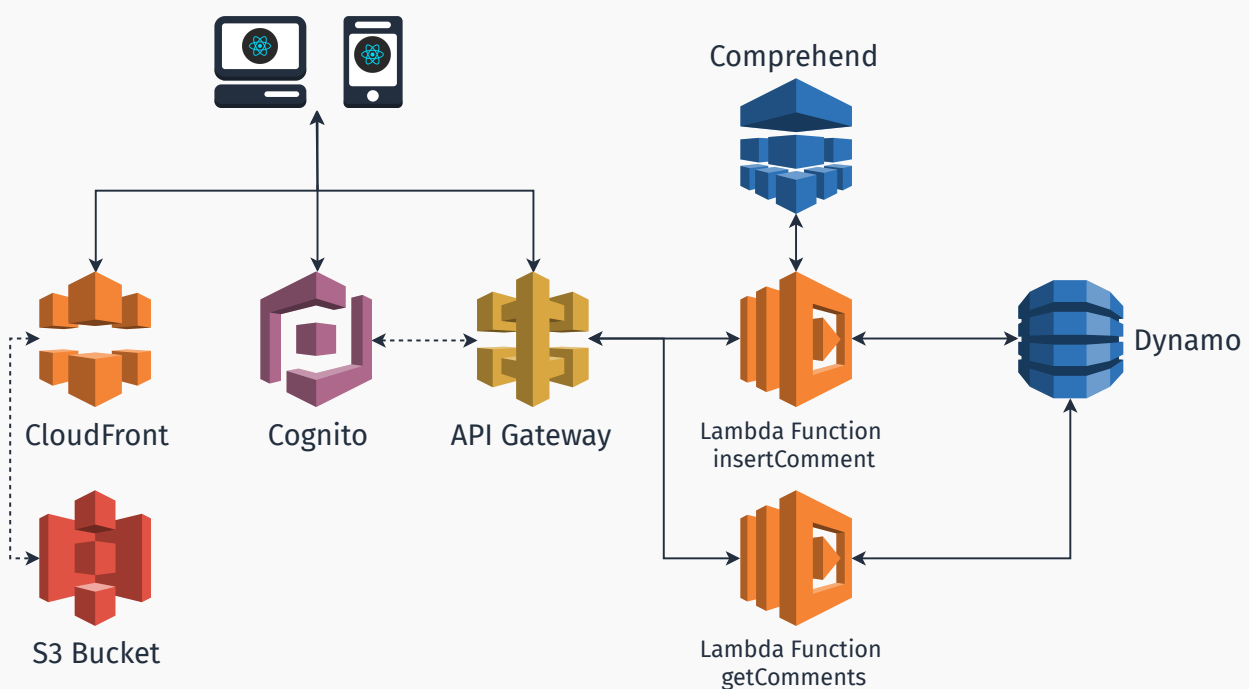
BUILDING A SERVERLESS WEB APPLICATION WITH AWS

It's demo time,
again!

- Remember the web application for this very talk we deployed earlier?
- Now we'll make it **serverless**, and add more features:
 - Sign-up and Authentication (Amazon Cognito)
 - Language detection and **sentiment analysis** on comments (Amazon Comprehend)
 - Deploy it on a global CDN to minimize latency (Amazon CloudFront)

101/187

ARCHITECTURE



102/187

WHAT YOU'LL NEED

- An AWS account (a free one will suffice)
- git version control (recommended)
- Node.js
- Python (recommended)

103/187

STEP 1: GET THE APP

Clone the git repository  Github

```
D:> git clone https://github.com/luistar/serverless-webapp.git serverless-webapp
```

104/187

STEP 2: INSTALL DEPENDENCIES

```
D:> cd serverless-webapp  
D:\serverless-webapp> npm install
```

105/187

STEP 3: INSTALL AWS-MOBILE CLI TOOL

```
D:\serverless-webapp> npm -g install awsmobile-cli
```

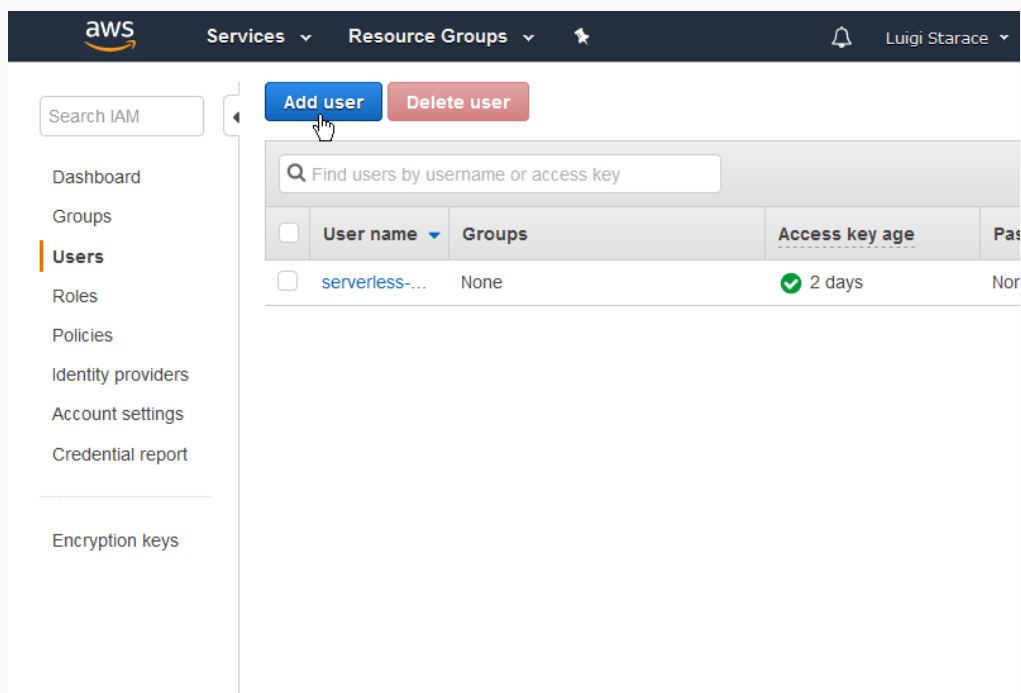
106/187

STEP 4: CREATE A NEW USER ON IAM

In order to use awsmobile-cli you're gonna need an access key id and a secret access key. If you don't already have one, go the IAM console and create a new user.

107/187

STEP 4: CREATE A NEW USER ON IAM



108/187

STEP 4: CREATE A NEW USER ON IAM

aws

Services

Resource Groups

Luigi Starace

Add user

12

Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name*serverless-webapp-administrator

+ Add another user

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Access type*☒ Programmatic access

Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

☐ AWS Management Console access

Enables a **password** that allows users to sign-in to the AWS Management Console.

STEP 4: CREATE A NEW USER ON IAM

aws

Services

Resource Groups

Luigi Starace

Add user

12

Set permissions for serverless-webapp-administrator

Add user to group

Copy permissions from existing user

Attach existing policies directly

Attach one or more existing policies directly to the users or create a new policy. [Learn more](#)

Create policyRefresh

Filter: Policy typeSearch

	Policy name	Type	Attachments	Description
<input checked="" type="checkbox"/>	AdministratorAccess	Job function	1	Provides full access to AWS services
<input type="checkbox"/>	AlexaForBusinessDe...	AWS managed	0	Provide device setup access to Alex...
<input type="checkbox"/>	AlexaForBusinessFu...	AWS managed	0	Grants full access to AlexaForBusine...
<input type="checkbox"/>	AlexaForBusinessCa...	AWS managed	0	Provide gateway execution access tr...

STEP 4: CREATE A NEW USER ON IAM

aws

Services

Resource Groups

Luigi Starace

Global

Support

Add user

1234

Review

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

User details

User name

serverless-webapp-administrator

AWS access type

Programmatic access - with an access key

Permissions summary

The following policies will be attached to the user shown above.

Type	Name
Managed policy	AdministratorAccess

Cancel

Previous

Create user

STEP 4: CREATE A NEW USER ON IAM

Be sure to write up your keys and to keep them safe!

 Read more about security

aws

Services

Resource Groups

Luigi Starace

Global

Support

Add user

1234

Success

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: [https://\[redacted\].signin.aws.amazon.com/console](https://[redacted].signin.aws.amazon.com/console)

Download .csv

	User	Access key ID	Secret access key
▶	serverless-webapp-administrator	[redacted]	[redacted] Hide

Close

STEP 5: CONFIGURE AWSMOBILE-CLI

Configure AWS Mobile CLI.

```
D:\serverless-webapp> awsmobile configure

configure aws
? accessKeyId:  <YOUR_ACCESS_KEY_ID>
? secretAccessKey:  <YOUR_SECRET_ACCESS_KEY>
? region:  eu-central-1
```

113/187

STEP 6: INITIALIZE A NEW AWS MOBILE PROJECT

```
D:\serverless-webapp> awsmobile init

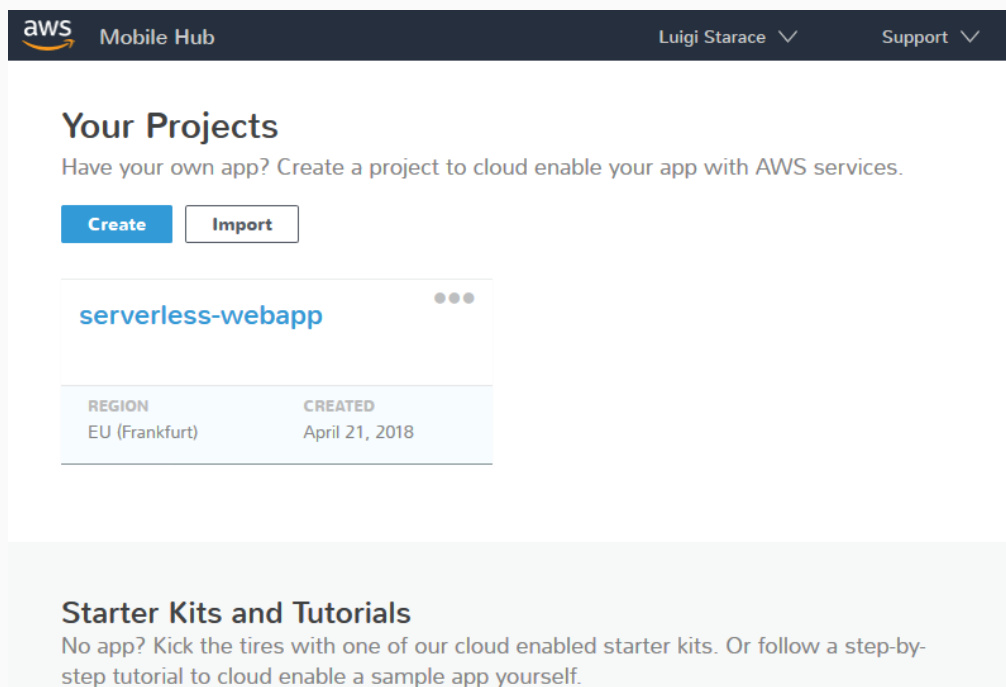
Please tell us about your project:
? Where is your project's source directory:  src
? Where is your project's distribution directory that
  stores build artifacts:  build
? What is your project's build command:  npm.cmd run-
  script build
? What is your project's start command for local test
  run:  npm.cmd run-script start

? What awsmobile project name would you like to use:
  serverless-webapp
```

114/187

STEP 6: INITIALIZE A NEW AWS MOBILE PROJECT

Visit the AWS Mobile Console. Your newly created project should be waiting for you there.



115/187

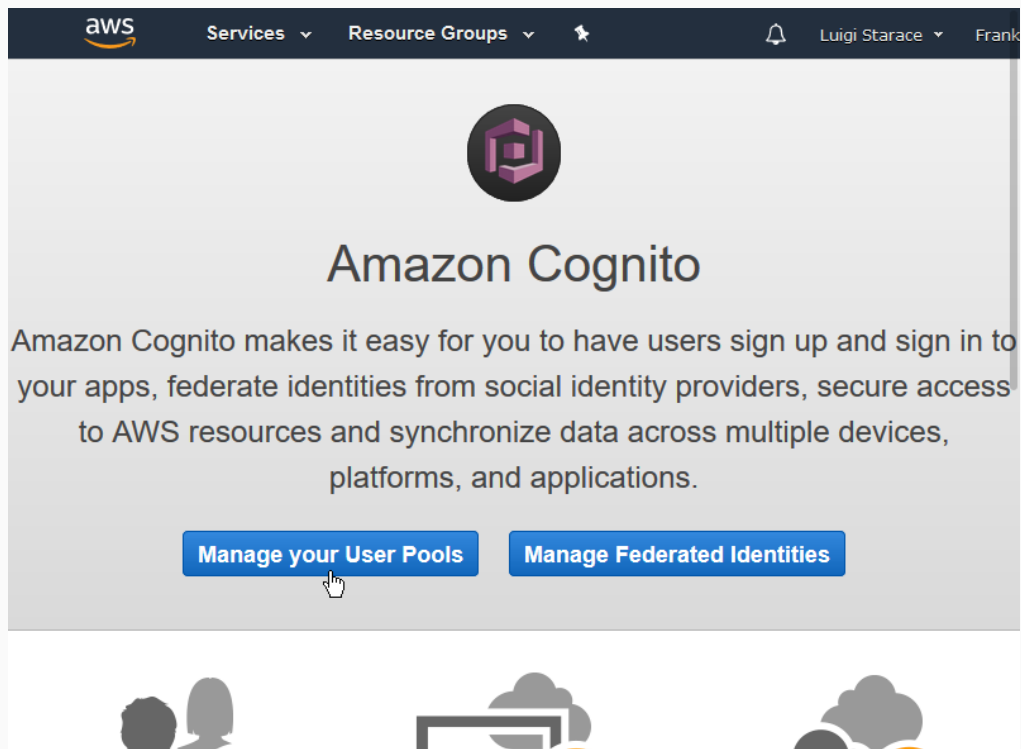
STEP 7: CREATE A COGNITO USER POOL

First we're gonna need a Cognito User Pool to authenticate our users. Let's create one.

116/187

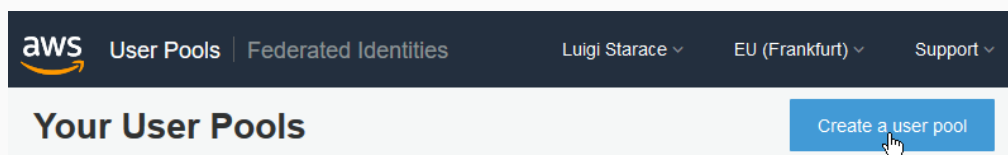
STEP 7: CREATE A COGNITO USER POOL

Visit the Cognito Console.



117/187

STEP 7: CREATE A COGNITO USER POOL



118/187

STEP 7: CREATE A COGNITO USER POOL

Insert a name for your user pool.

What do you want to name your user pool?

Give your user pool a descriptive name so you can easily identify it in the future.

Pool name

serverless-webapp-user-pool

How do you want to create your user pool?

Review defaults

Start by reviewing the defaults and then customize as desired

Step through settings

Step through each setting to make your choices

STEP 7: CREATE A COGNITO USER POOL

Make sure only an email is required.

Which standard attributes do you want to require?

All of the standard attributes can be used for user profiles, but the attributes you select will be required for sign up. You will not be able to change these requirements after the pool is created. If you select an attribute to be an alias, users will be able to sign-in using that value or their username. [Learn more about attributes.](#)

Required	Attribute	Required	Attribute
<input type="checkbox"/>	address	<input type="checkbox"/>	nickname
<input type="checkbox"/>	birthdate	<input type="checkbox"/>	phone number
<input checked="" type="checkbox"/>	email	<input type="checkbox"/>	picture
<input type="checkbox"/>	family name	<input type="checkbox"/>	preferred username
<input type="checkbox"/>	gender	<input type="checkbox"/>	profile
<input type="checkbox"/>	given name	<input type="checkbox"/>	zoneinfo
<input type="checkbox"/>	locale	<input type="checkbox"/>	updated at
<input type="checkbox"/>	middle name	<input type="checkbox"/>	website
<input type="checkbox"/>	name		

STEP 7: CREATE A COGNITO USER POOL

Review your configuration and create the pool.

Create a user pool

Name

Attributes

Policies

MFA and verifications

Message customizations

Tags

Devices

App clients

Triggers

Review

Pool name serverless-webapp-user-pool

Required attributes email

Alias attributes [Choose alias attributes...](#)

Username attributes [Choose username attributes...](#)

Custom attributes [Choose custom attributes...](#)

Minimum password length 8

Password policy no requirements

User sign ups allowed? Users can sign themselves up

MFA [Enable MFA...](#)

Verifications Email

Tags [Choose tags for your user pool](#)

App clients [Add app client...](#)

Triggers [Add triggers...](#)

Create pool

121/187

STEP 8: CONFIGURE USER SIGN-IN IN THE MOBILE APP CONSOLE

Return to the AWS Mobile Console and open your project.

Your Projects

Have your own app? Create a project to cloud enable your app with AWS services.

Create **Import**

serverless-webapp

REGION	CREATED
EU (Frankfurt)	April 21, 2018

Starter Kits and Tutorials


No app? Kick the tires with one of our cloud enabled starter kits. Or follow a step-by-step tutorial to cloud enable a sample app yourself.

122/187

STEP 8: CONFIGURE USER SIGN-IN IN THE MOBILE APP CONSOLE

Add user sign in to the project.

Add More Backend Features




User Sign-in

Let your users sign in with public identity providers or your own identity system.

Powered by Amazon Cognito

+




NoSQL Database

Store data in a fully managed cloud database.

Powered by Amazon DynamoDB

+




User File Storage

Store files in the cloud.

Powered by Amazon S3

+



Conversational Bots

Add voice and chat bots to your mobile app.

Powered by Amazon Lex


+

123/187


STEP 8: CONFIGURE USER SIGN-IN IN THE MOBILE APP CONSOLE

Import your newly created user pool.


Add sign-in Providers




Email and Password



Facebook Login



Google Sign-In



SAML Federation

Create new or import

☐ Create a new user pool
Create a basic user pool powered by Cognito

☒ Import an existing user pool
Use one of your existing Cognito user pools

Select user pool

1

Name	Id
serverless-webapp-user-pool	eu-central-1_kznXIIIdC

124/187

STEP 8: CONFIGURE USER SIGN-IN IN THE MOBILE APP CONSOLE

Pull your new project configuration with

```
D:\serverless-webapp> awsmobile pull
```

If you were to start the application locally with

```
D:\serverless-webapp> npm start
```

The authentication will now work.

125/187

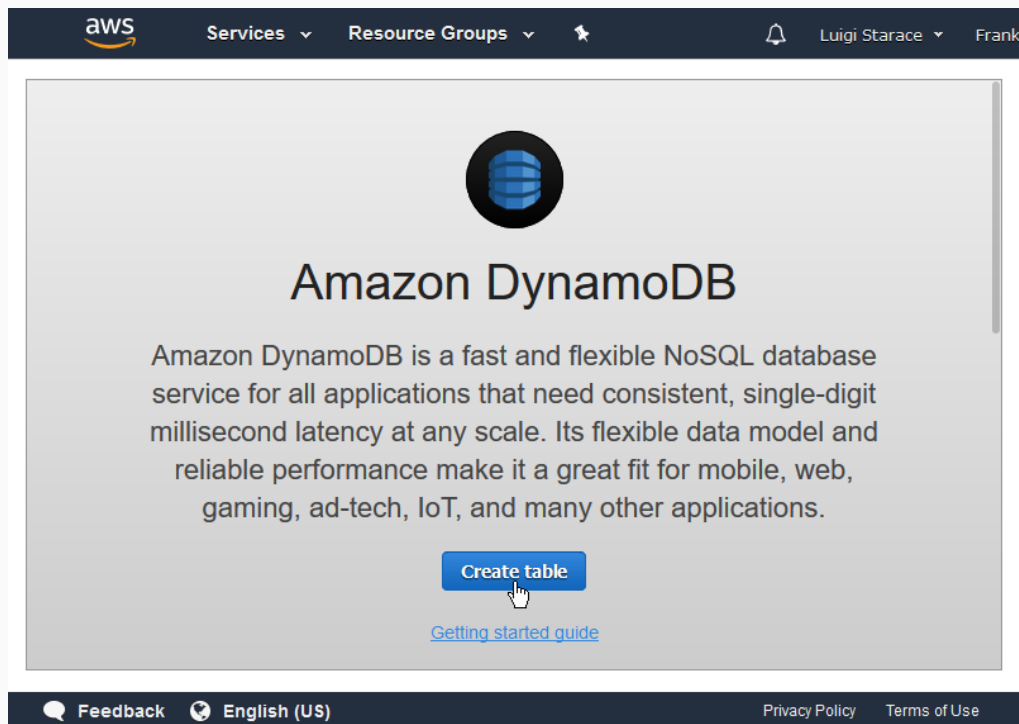
STEP 9: CONFIGURE AMAZON DYNAMO DATABASE

Next thing we're gonna need is a database to store the comments. In this tutorial we'll use the NoSQL database Amazon Dynamo.

126/187

STEP 9: CONFIGURE AMAZON DYNAMO DATABASE

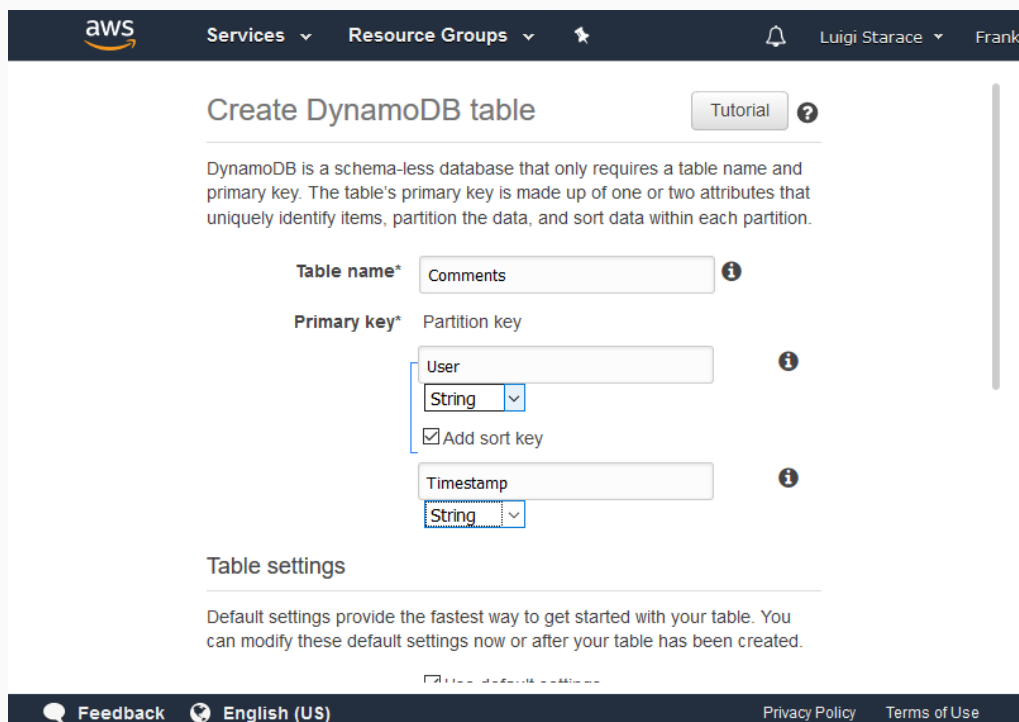
Visit the Dynamo Dashboard.



127/187

STEP 9: CONFIGURE AMAZON DYNAMO DATABASE

Create a new Comments table as shown. Leave other fields with their default values.



128/187

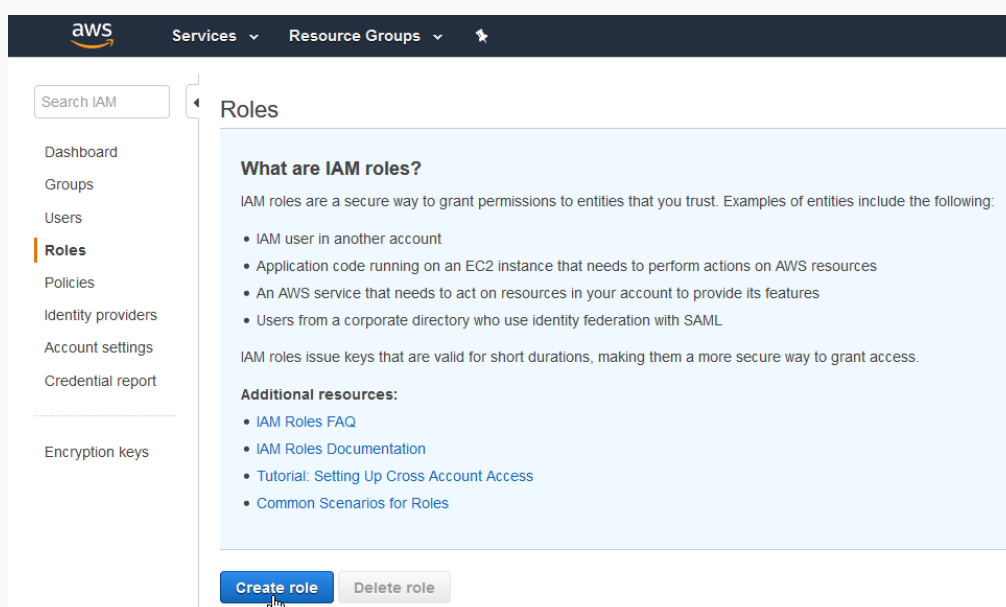
STEP 10: CREATE A ROLE FOR THE LAMBDA FUNCTIONS

Before we create our Lambda functions, let's create a new role defining the authorizations we want them to have.

129/187

STEP 10: CREATE A ROLE FOR THE LAMBDA FUNCTIONS

Return to the IAM console and select the “role” tab, then the “create role” button.



130/187

STEP 10: CREATE A ROLE FOR THE LAMBDA FUNCTIONS

Select “AWS Service” and “Lambda” in the wizard, as shown in the picture.

The screenshot shows the AWS IAM console 'Create role' wizard. The 'Select type of trusted entity' step has three options: 'AWS service' (selected), 'Another AWS account', and 'Web identity'. Below this, the 'Choose the service that will use this role' step shows a list of services. 'Lambda' is selected, and a list of services is displayed below it.

Service	Description
EC2	Allows EC2 instances to call AWS services on your behalf.
Lambda	Allows Lambda functions to call AWS services on your behalf.
API Gateway	
AppSync	
Application Auto Scaling	
Config	
DMS	
Data Pipeline	
Elastic Container Service	
Elastic Transcoder	
ElasticLoadBalancing	
Lex	
Machine Learning	
MediaConvert	

131/187

STEP 10: CREATE A ROLE FOR THE LAMBDA FUNCTIONS

Add the AWSLambdaBasicExecutionRole, as shown.

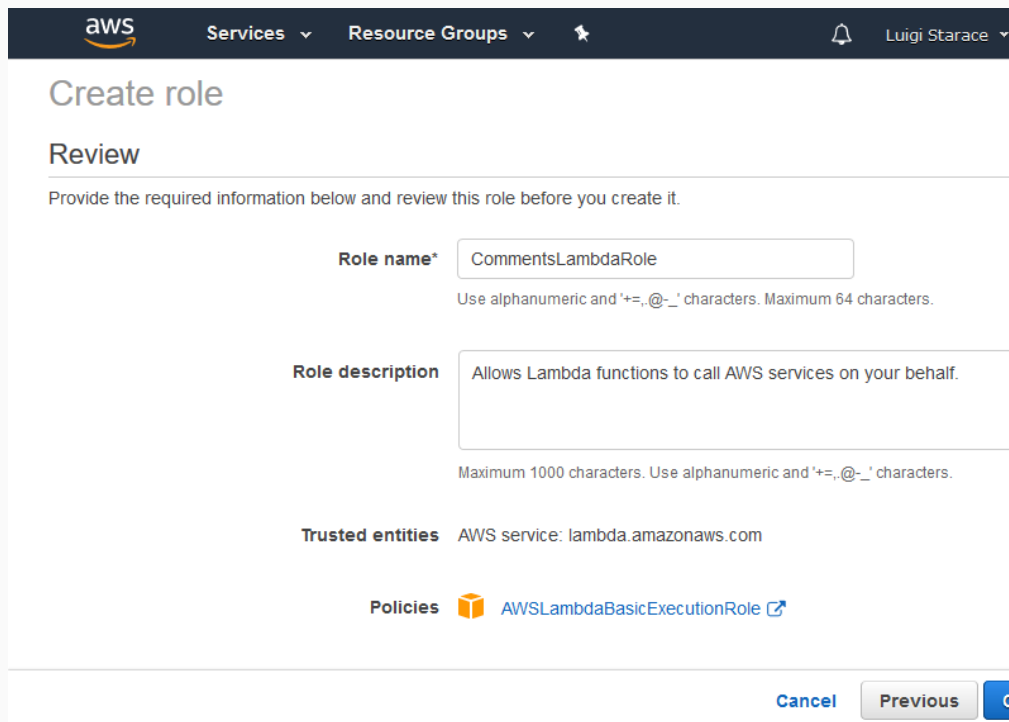
The screenshot shows the AWS IAM console 'Create role' wizard, step 2: 'Attach permissions policies'. The 'AWSLambdaBasicExecutionRole' policy is selected. The table below shows the details of the selected policy.

Policy name	Attachments	Description
AWSLambdaBasicExecutionRole	0	Provides write permissions to CloudWatch Logs.

132/187

STEP 10: CREATE A ROLE FOR THE LAMBDA FUNCTIONS

Insert a name and a description and create the role.

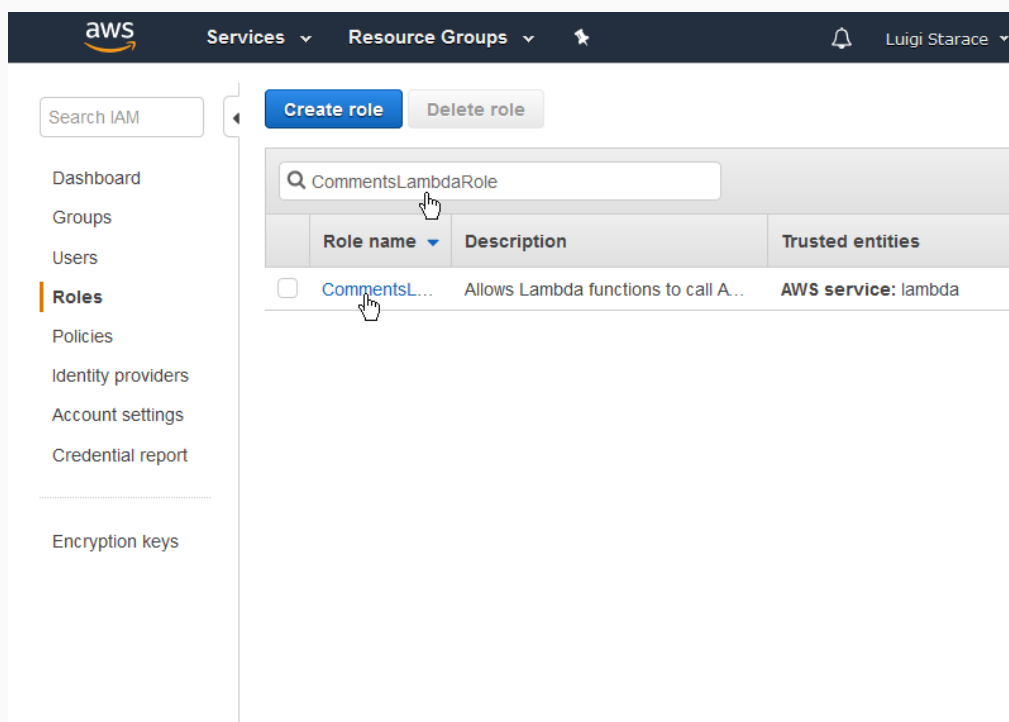


The screenshot shows the 'Create role' review page in the AWS IAM console. The page has a dark header with the AWS logo, 'Services', 'Resource Groups', and a user profile 'Luigi Starace'. The main heading is 'Create role' followed by a sub-heading 'Review'. Below this, a message states: 'Provide the required information below and review this role before you create it.' The form contains three sections: 'Role name*' with the value 'CommentsLambdaRole' and a note 'Use alphanumeric and '+=, @-_' characters. Maximum 64 characters.'; 'Role description' with the value 'Allows Lambda functions to call AWS services on your behalf.' and a note 'Maximum 1000 characters. Use alphanumeric and '+=, @-_' characters.'; and 'Trusted entities' with the value 'AWS service: lambda.amazonaws.com'. At the bottom, there is a 'Policies' section showing 'AWSLambdaBasicExecutionRole' with a link icon. At the very bottom right, there are three buttons: 'Cancel', 'Previous', and 'Create' (partially visible).

133/187

STEP 10: CREATE A ROLE FOR THE LAMBDA FUNCTIONS

Go back to the roles tab in the IAM Dashboard and select your newly created role.

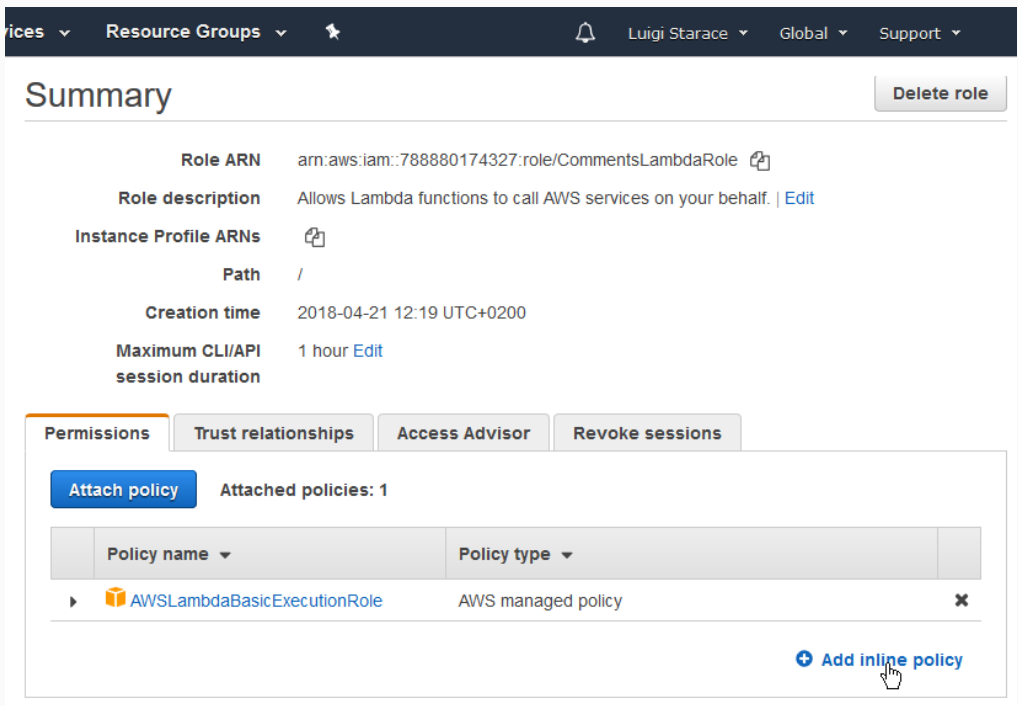


The screenshot shows the 'Roles' tab in the AWS IAM console. The left sidebar contains a search bar 'Search IAM' and a list of navigation items: 'Dashboard', 'Groups', 'Users', 'Roles' (highlighted), 'Policies', 'Identity providers', 'Account settings', 'Credential report', and 'Encryption keys'. The main area has a 'Create role' button and a 'Delete role' button. Below these is a search bar containing 'CommentsLambdaRole'. A table lists the roles, with one role selected: 'CommentsL...' (truncated). The table has columns for 'Role name', 'Description', and 'Trusted entities'. The selected role's description is 'Allows Lambda functions to call A...' and its trusted entity is 'AWS service: lambda'. A mouse cursor is pointing at the role name 'CommentsL...'.

134/187

STEP 10: CREATE A ROLE FOR THE LAMBDA FUNCTIONS

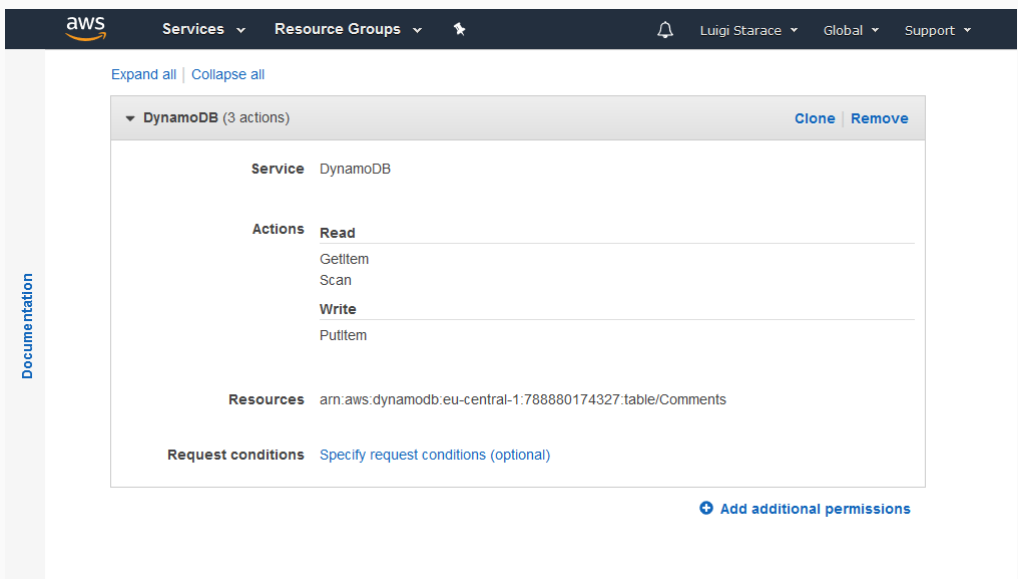
Add inline policies to allow the role to access Dynamo DB and Comprehend.



135/187

STEP 10: CREATE A ROLE FOR THE LAMBDA FUNCTIONS

Add an inline policy to allow this role to access Dynamo DB tables.



136/187

STEP 10: CREATE A ROLE FOR THE LAMBDA FUNCTIONS

Save the inline policy.

aws

Services ▾Resource Groups ▾★

Luigi Starace ▾Global ▾Support ▾

Create policy

12

Review policy

Before you create this policy, provide the required information and review this policy.

Name*

DynamoCommentsPolicy

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

Summary

Filter

Service ▾	Access level	Resource
Allow (1 of 136 services) <a>Show remaining 135		
DynamoDB	Limited: Read, Write	TableName string like Com

STEP 10: CREATE A ROLE FOR THE LAMBDA FUNCTIONS

Now add another inline policy to allow this role to access Comprehend’s detect language and detect sentiment features.

aws

Services ▾Resource Groups ▾★

Luigi Starace ▾Global ▾Sup

Create policy

12

A policy defines the AWS permissions that you can assign to a user, group, or role. You can create and edit a policy in the visual editor and using JSON. Learn more

Visual editorJSON

Import managed policy

Expand all | Collapse all

Comprehend (2 actions)

CloneRemove

Service

Comprehend

Actions

Read

DetectDominantLanguage

DetectSentiment

Resources

All resources have been selected for you because this service does not allow you to choose specific resources.

Request conditions

Specify request conditions (optional)

Add additional permissions

STEP 10: CREATE A ROLE FOR THE LAMBDA FUNCTIONS

Your role should look like this.

The screenshot shows the AWS IAM console for the role 'CommentsLambdaRole'. The 'Summary' tab is active, displaying the following details:

- Role ARN:** `arn:aws:iam::788880174327:role/CommentsLambdaRole`
- Role description:** Allows Lambda functions to call AWS services on your behalf. [Edit](#)
- Instance Profile ARNs:** [Copy](#)
- Path:** /
- Creation time:** 2018-04-21 12:19 UTC+0200
- Maximum CLI/API session duration:** 1 hour [Edit](#)

The 'Permissions' tab is also visible, showing 'Attached policies: 3':

Policy name	Policy type	
AWSLambdaBasicExecutionRole	AWS managed policy	✕
ComprehendCommentsPolicy	Inline policy	✕
DynamoCommentsPolicy	Inline policy	✕

[Add inline policy](#)

139/187

STEP 11: CREATE THE LAMBDA FUNCTIONS

Go to the Lambda Dashboard and click on the “Create function” button.

The screenshot shows the AWS Lambda console. The 'Functions' tab is active, displaying a list of functions. The 'Create function' button is highlighted with a mouse cursor.

Functions (1)

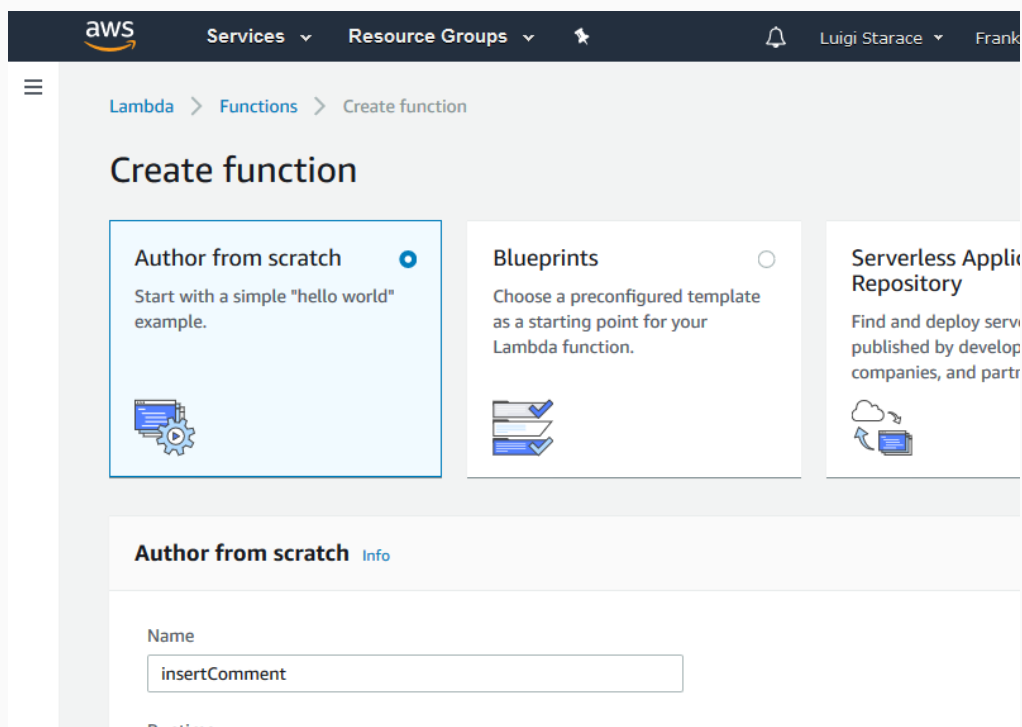
Filter by tags and attributes or search by keyword

Function name	Description	Runtime	Code size	Last Modified
autoAuthenticate		Node.js 6.10	422 bytes	3 days ago

140/187

STEP 11: CREATE THE LAMBDA FUNCTIONS

Select “Author from scratch”



The screenshot shows the AWS Lambda 'Create function' page. The 'Author from scratch' option is selected with a blue radio button. Below the options, the 'Name' field contains 'insertComment'. The 'Runtime' field is partially visible at the bottom.


aws Services Resource Groups Luigi Starace Frank

Lambda > Functions > Create function

Create function


Author from scratch ☒

Start with a simple "hello world" example.




Blueprints ☐

Choose a preconfigured template as a starting point for your Lambda function.



Serverless Application Repository ☐

Find and deploy serverless applications published by development companies, and partners.



Author from scratch [Info](#)

Name

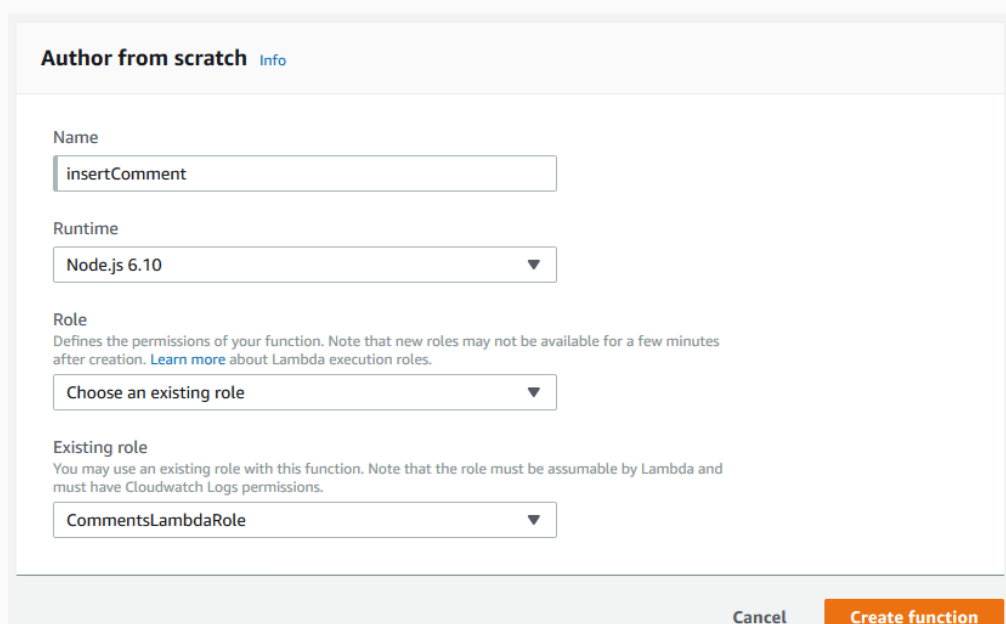
insertComment

Runtime

141/187

STEP 11: CREATE THE LAMBDA FUNCTIONS

Name the function `insertComment` and select Node.js 6.10 as the runtime and the role we created earlier as the role.



The screenshot shows the 'Author from scratch' section of the AWS Lambda console. The 'Name' field is 'insertComment', the 'Runtime' is 'Node.js 6.10', and the 'Role' is 'CommentsLambdaRole'.

Author from scratch [Info](#)

Name

insertComment

Runtime

Node.js 6.10

Role

Defines the permissions of your function. Note that new roles may not be available for a few minutes after creation. [Learn more](#) about Lambda execution roles.

Choose an existing role

Existing role

You may use an existing role with this function. Note that the role must be assumable by Lambda and must have Cloudwatch Logs permissions.

CommentsLambdaRole

Cancel Create function

142/187

STEP 11: CREATE THE LAMBDA FUNCTIONS

Insert the code provided in the `lambda/insertComment.js` file in the next screen, then save the lambda function.

143/187

STEP 11: CREATE THE LAMBDA FUNCTIONS

Proceed similarly and create the `getComments` Lambda function.

144/187

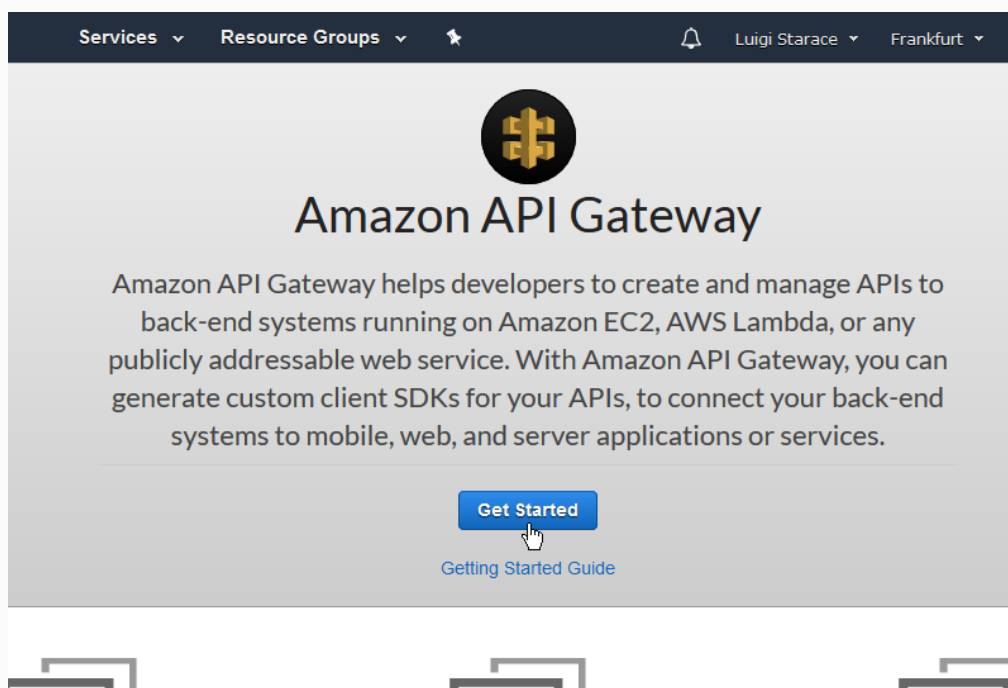
STEP 12: CREATE THE APIS TO EXPOSE THE LAMBDA FUNCTIONS

Once we have our Lambda functions, let's hook 'em up with an API our web app can rely upon.

145/187

STEP 12: CREATE THE APIS TO EXPOSE THE LAMBDA FUNCTIONS

Visit the API Gateway Dashboard



146/187

STEP 12: CREATE THE APIS TO EXPOSE THE LAMBDA FUNCTIONS

Create a new API and select a name and a description.

The screenshot shows the 'Create new API' page in the AWS Management Console. The breadcrumb trail is 'Amazon API Gateway' > 'APIs' > 'Create'. The page title is 'Create new API'. Below the title, a note states: 'In Amazon API Gateway, an API refers to a collection of resources and methods that can be invoked through HTTPS endpoints.' There are three radio buttons: 'New API' (selected), 'Import from Swagger', and 'Example API'. Under the 'Settings' section, it says 'Choose a friendly name and description for your API.' The form fields are: 'API name*' with the value 'serverless-webapp-api', 'Description' with the value 'REST API to store and retrieve comments.', and 'Endpoint Type' with a dropdown menu showing 'Edge optimized'. A '* Required' note is at the bottom left. The footer includes 'Feedback', 'English (US)', 'Privacy Policy', and 'Terms of Use'.

147/187

STEP 12: CREATE THE APIS TO EXPOSE THE LAMBDA FUNCTIONS

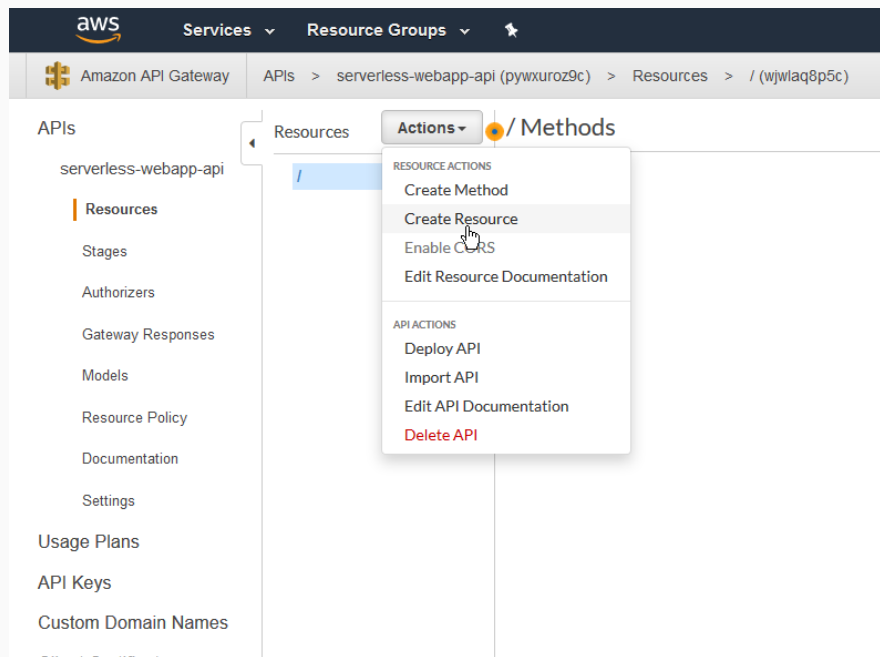
Select the Authorizers tab and create a new Authorizer for your API. Give it a name, select the user pool we created earlier, and enter “Authorization” in the “Token Source” field.

The screenshot shows the 'Create Authorizer' page in the AWS Management Console. The breadcrumb trail is 'Amazon API Gateway' > 'APIs' > 'serverless-webapp-api (pywxuroz9c)' > 'Authorizers'. The left sidebar shows the 'Authorizers' tab selected. The main content area has a title 'Authorizers' and a subtitle 'Authorizers enable you to control access to your APIs using Amazon Cognito User Pools or a Lambda function.' There is a '+ Create New Authorizer' button. Below it is the 'Create Authorizer' form. The form fields are: 'Name*' with the value 'serverless-webapp-api-authorizer', 'Type*' with radio buttons for 'Lambda' and 'Cognito' (selected), 'Cognito User Pool*' with a dropdown showing 'eu-central-1' and 'serverless-webapp-user-pool', 'Token Source*' with the value 'Authorization', and 'Token Validation*' which is empty. At the bottom are 'Create' and 'Cancel' buttons.

148/187

STEP 12: CREATE THE APIS TO EXPOSE THE LAMBDA FUNCTIONS

Select the resources tab and create a new Resource



149/187

STEP 12: CREATE THE APIS TO EXPOSE THE LAMBDA FUNCTIONS

Name the resource comments, enable CORS and continue.

New Child Resource

Use this page to create a new child resource for your resource. ⓘ

Configure as [proxy resource](#) ☐ ⓘ

Resource Name*

Resource Path*

You can add path parameters using brackets. For example, the resource path **{username}** represents a path parameter called 'username'. Configuring **/ {proxy+}** as a proxy resource catches all requests to its sub-resources. For example, it works for a GET request to /foo. To handle requests to /, add a new ANY method on the / resource.

Enable API Gateway CORS ☒ ⓘ

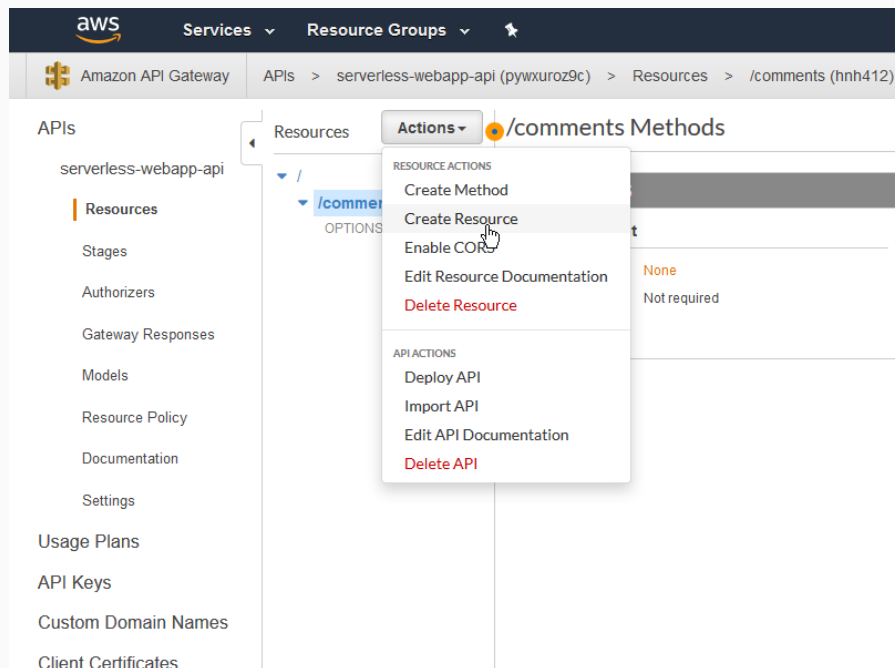
* Required

[Cancel](#) [Create Resource](#)

150/187

STEP 12: CREATE THE APIs TO EXPOSE THE LAMBDA FUNCTIONS

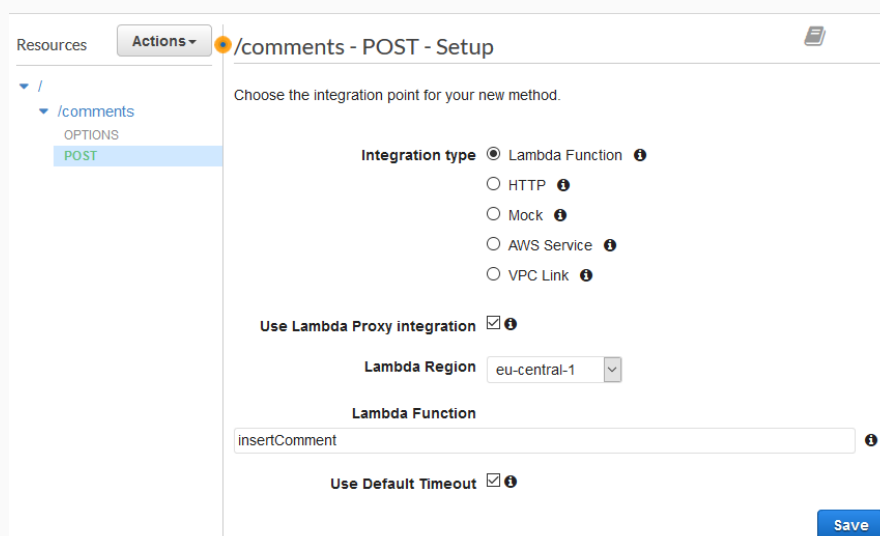
Select the comments resource and create a new POST method.



151/187

STEP 12: CREATE THE APIs TO EXPOSE THE LAMBDA FUNCTIONS

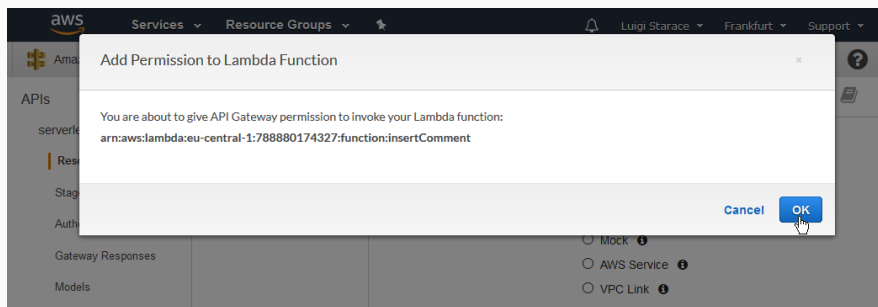
As shown, select the insertComment Lambda function you created earlier as the integration point.



152/187

STEP 12: CREATE THE APIS TO EXPOSE THE LAMBDA FUNCTIONS

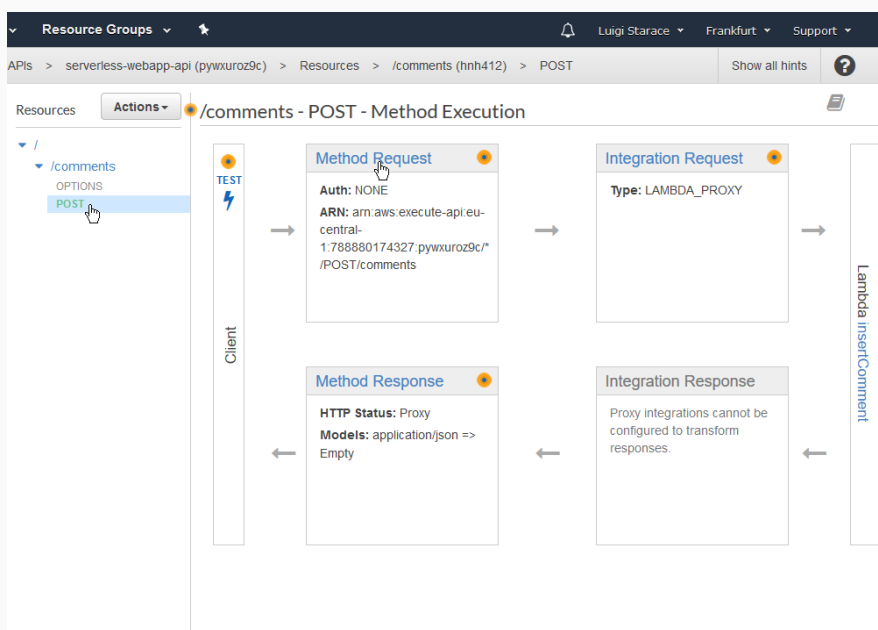
Give API Gateway the permission to invoke the Lambda function



153/187

STEP 12: CREATE THE APIS TO EXPOSE THE LAMBDA FUNCTIONS

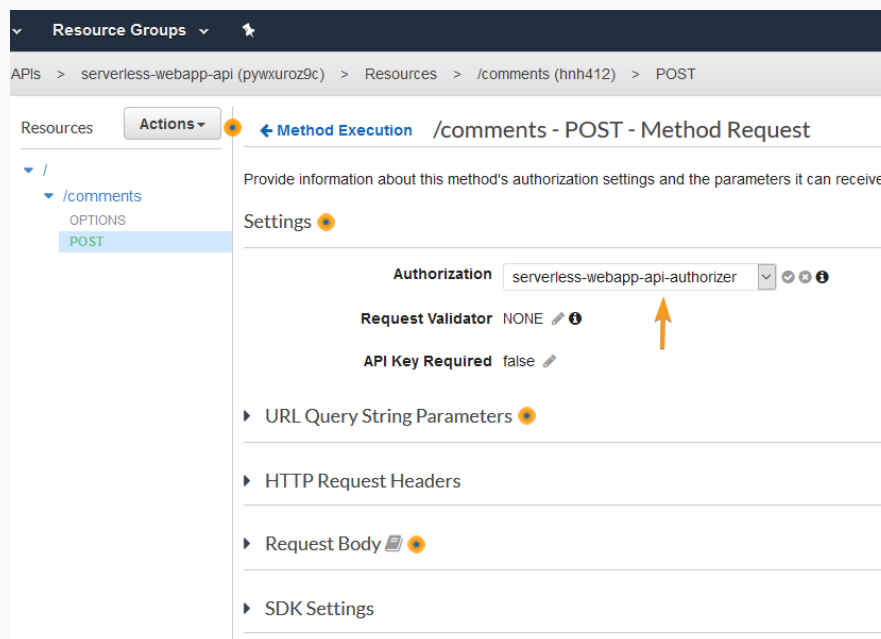
Select the POST method on the comments resource, then select the Method Request card.



154/187

STEP 12: CREATE THE APIs TO EXPOSE THE LAMBDA FUNCTIONS

Select the authorizer you created earlier for the Authorization field in the Settings section.



155/187

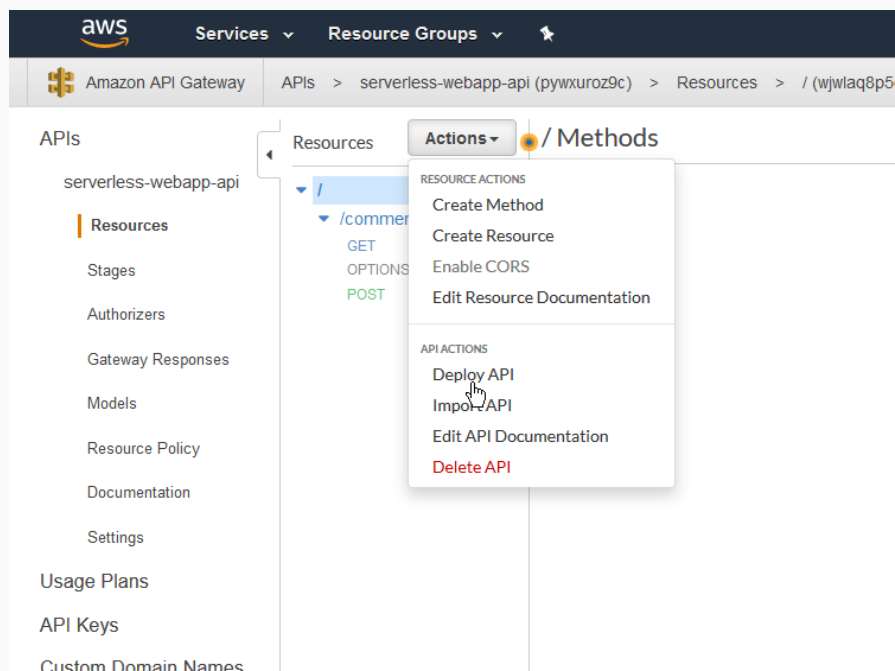
STEP 12: CREATE THE APIs TO EXPOSE THE LAMBDA FUNCTIONS

Proceed similarly to hook up the GET method with the getComments Lambda function. This time authorization is not needed. We want non-authenticated users to be able to fetch the comments.

156/187

STEP 12: CREATE THE APIS TO EXPOSE THE LAMBDA FUNCTIONS

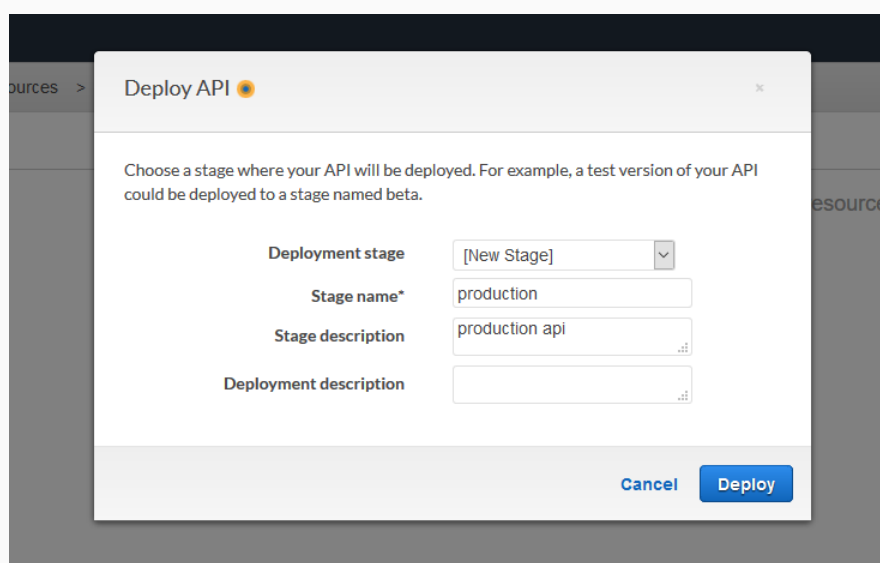
Once you are done setting up the GET method, select the root resource, then Deploy API.



157/187

STEP 12: CREATE THE APIS TO EXPOSE THE LAMBDA FUNCTIONS

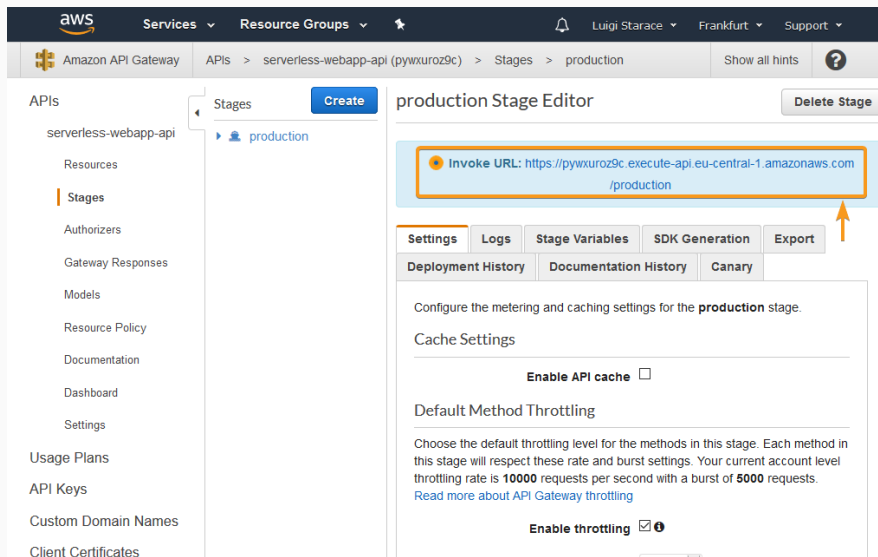
Insert deployment stage informations and deploy.



158/187

STEP 12: CREATE THE APIS TO EXPOSE THE LAMBDA FUNCTIONS

Select the stages tab and note the invoke url.



159/187

STEP 12: CREATE THE APIS TO EXPOSE THE LAMBDA FUNCTIONS

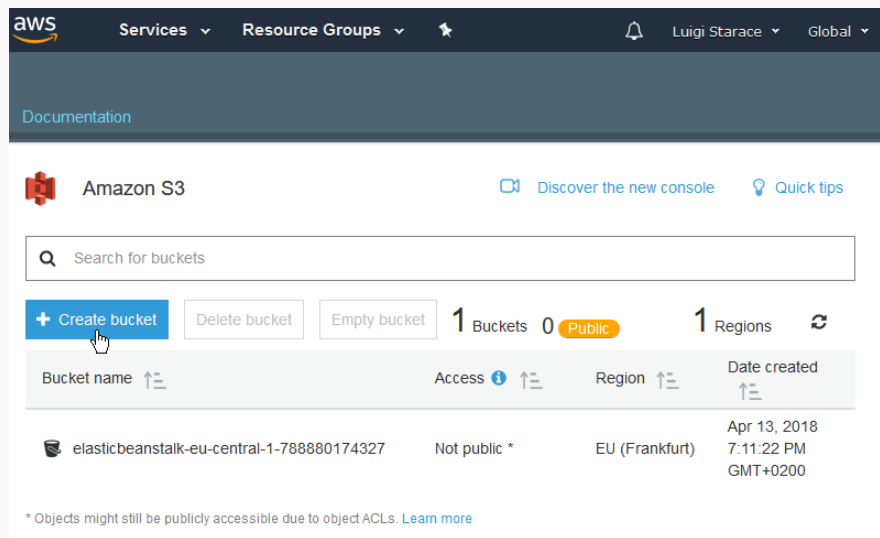
Change the CommentsAPI class accordingly in `src/API/CommentsAPI.js`.

```
class CommentsAPI {  
  
    constructor(){  
        this.endpoint = '<YOUR_INVOKE_URL_HERE>';  
    }  
  
    // ...  
  
}
```

160/187

STEP 13: HOST THE STATIC FILES WITH S3

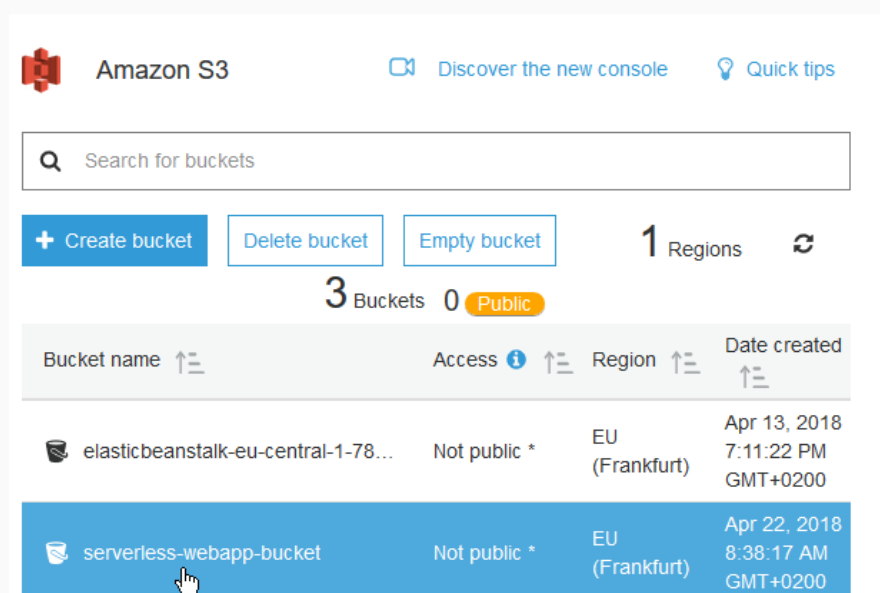
Visit the S3 Dashboard and create a new bucket.



161/187

STEP 13: HOST THE STATIC FILES WITH S3

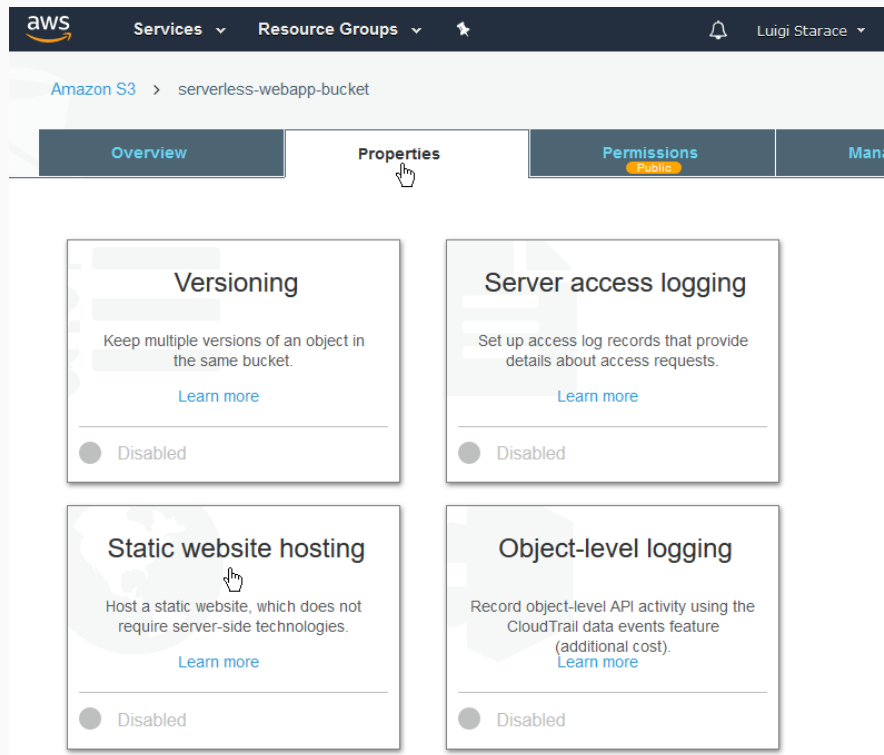
Select your newly created S3 bucket.



162/187

STEP 13: HOST THE STATIC FILES WITH S3

Under the properties tab, select the Static website hosting card.



163/187

STEP 13: HOST THE STATIC FILES WITH S3

Fill the form as shown in the picture. Note the Endpoint, as it will be the URL of the website!

The screenshot shows the 'Static website hosting' configuration form. At the top, there's a blue header with the title 'Static website hosting' and a close button. Below the header, the 'Endpoint' is displayed as 'http://serverless-webapp-bucket.s3-website.eu-central-1.amazonaws.com'. There are two radio button options: 'Use this bucket to host a website' (which is selected) and 'Redirect requests'. Below the first option, there are two text input fields: 'Index document' (containing 'index.html') and 'Error document' (containing 'index.html'). There is also a text area for 'Redirection rules (optional)'. At the bottom, there are two radio button options: 'Redirect requests' and 'Disable website hosting'. At the very bottom, there are 'Cancel' and 'Save' buttons.

164/187

STEP 13: HOST THE STATIC FILES WITH S3

Now we'll show how to upload the static website via AWS CLI. This operation can also be performed via the web interface of the bucket.

165/187

STEP 13: HOST THE STATIC FILES WITH S3

Install AWS CLI

```
D:\serverless-webapp> pip install awscli --upgrade
```

Then configure it

```
D:\serverless-webapp> aws configure
AWS Access Key ID [None]: <YOUR_ACCESS_KEY_ID>
AWS Secret Access Key [None]: <YOUR_SECRET_ACCESS_KEY>
Default region name [None]: eu-central-1
Default output format [None]: json
```

166/187

STEP 13: HOST THE STATIC FILES WITH S3

Build the website

```
D:\serverless-webapp> npm run build-css  
D:\serverless-webapp> npm run build
```

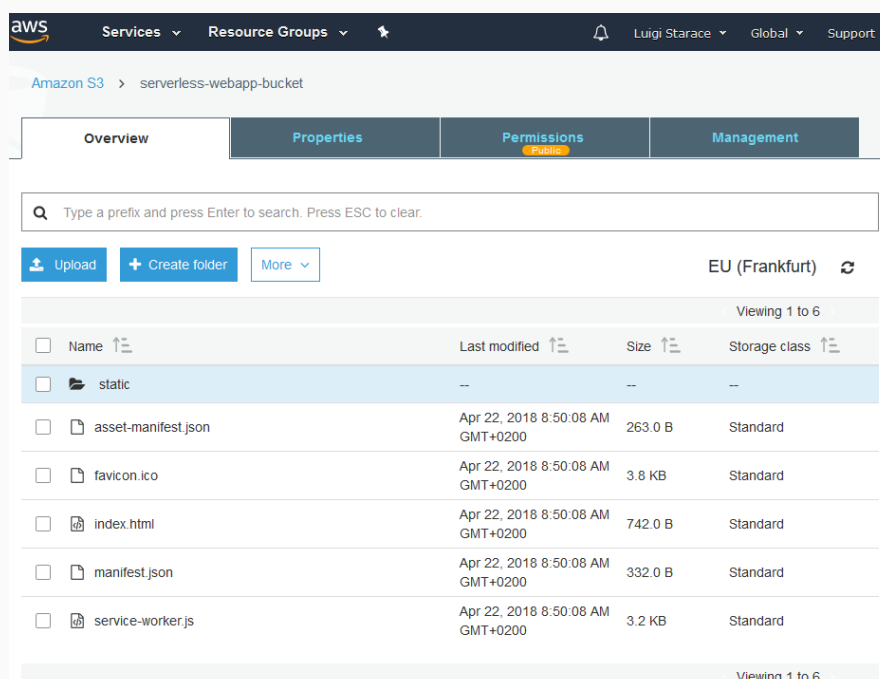
Then upload the files with

```
D:\serverless-webapp> aws s3 sync ./build s3://  
serverless-webapp-bucket --acl public-read
```

167/187

STEP 13: HOST THE STATIC FILES WITH S3

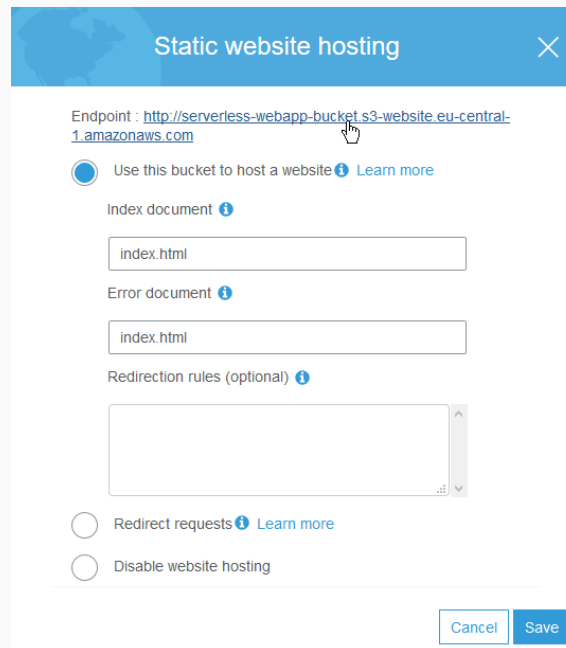
After the upload is done, your bucket should look like this.



168/187

STEP 13: HOST THE STATIC FILES WITH S3

Visit the Static website hosting card again under the properties tab, then click on the endpoint URL.



Static website hosting

Endpoint : <http://serverless-webapp-bucket-s3-website.eu-central-1.amazonaws.com>

☒ Use this bucket to host a website [Learn more](#)

Index document ⓘ

Error document ⓘ

Redirection rules (optional) ⓘ

☐ Redirect requests [Learn more](#)

☐ Disable website hosting

[Cancel](#) [Save](#)

169/187

STEP 13: HOST THE STATIC FILES WITH S3

You should see a very nice single-page serverless web application!



170/187

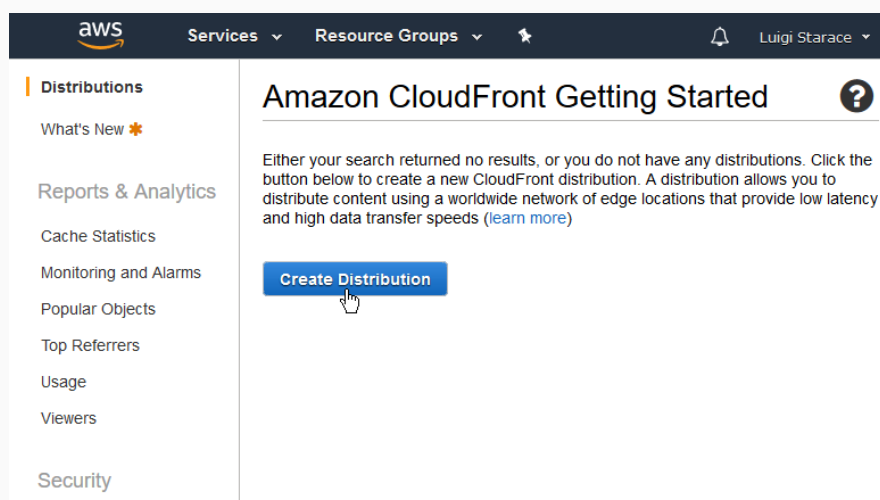
But wait, there's more!

Such a nice web application wouldn't be complete without a global CDN to speed up load times. So we'll now set up Amazon CloudFront to distribute the static files all over the globe.

171/187

STEP 14: OPTIMIZE LATENCY WITH CLOUDFRONT

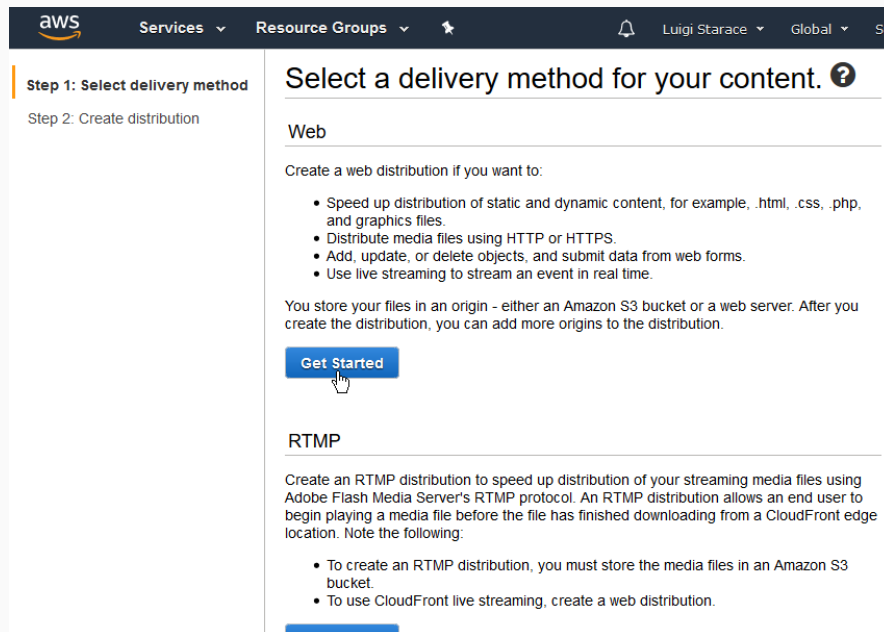
Visit the CloudFront Dashboard and create a new distribution.



172/187

STEP 14: OPTIMIZE LATENCY WITH CLOUDFRONT

Select web delivery method.



aws Services Resource Groups Luigi Starace Global

Step 1: Select delivery method
Step 2: Create distribution

Select a delivery method for your content. ?

Web

Create a web distribution if you want to:

- Speed up distribution of static and dynamic content, for example, .html, .css, .php, and graphics files.
- Distribute media files using HTTP or HTTPS.
- Add, update, or delete objects, and submit data from web forms.
- Use live streaming to stream an event in real time.

You store your files in an origin - either an Amazon S3 bucket or a web server. After you create the distribution, you can add more origins to the distribution.

Get Started

RTMP

Create an RTMP distribution to speed up distribution of your streaming media files using Adobe Flash Media Server's RTMP protocol. An RTMP distribution allows an end user to begin playing a media file before the file has finished downloading from a CloudFront edge location. Note the following:

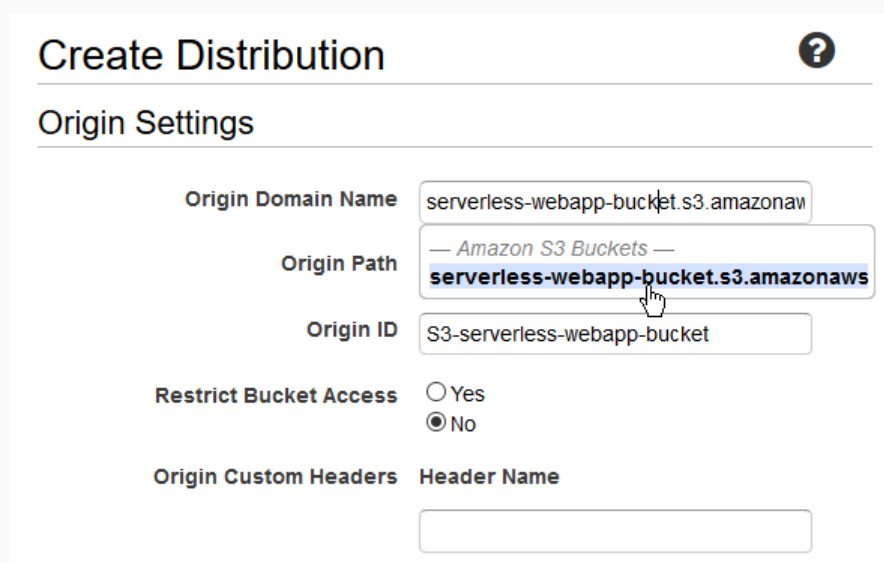
- To create an RTMP distribution, you must store the media files in an Amazon S3 bucket.
- To use CloudFront live streaming, create a web distribution.

Get Started

173/187

STEP 14: OPTIMIZE LATENCY WITH CLOUDFRONT

Select your website bucket as the origin.



Create Distribution ?

Origin Settings

Origin Domain Name: serverless-webapp-bucket.s3.amazonaws.com

Origin Path: — Amazon S3 Buckets —
serverless-webapp-bucket.s3.amazonaws.com

Origin ID: S3-serverless-webapp-bucket

Restrict Bucket Access: ☐ Yes ☒ No

Origin Custom Headers: Header Name

174/187

STEP 14: OPTIMIZE LATENCY WITH CLOUDFRONT

Select index.html as the default root object.

Supported HTTP Versions ☒ HTTP/2, HTTP/1.1, HTTP/1.0 ☐ HTTP/1.1, HTTP/1.0

Default Root Object

Logging ☐ On ☒ Off

Bucket for Logs

Log Prefix

Cookie Logging ☐ On ☒ Off

Enable IPv6 ☒ [Learn more](#)

Comment

175/187

STEP 14: OPTIMIZE LATENCY WITH CLOUDFRONT

It takes a few minutes to setup the distribution. When it's done the status will change to Deployed.

CloudFront Distributions

Create Distribution Distribution Settings Delete Enable Disable

Viewing: Any Delivery Method Any State

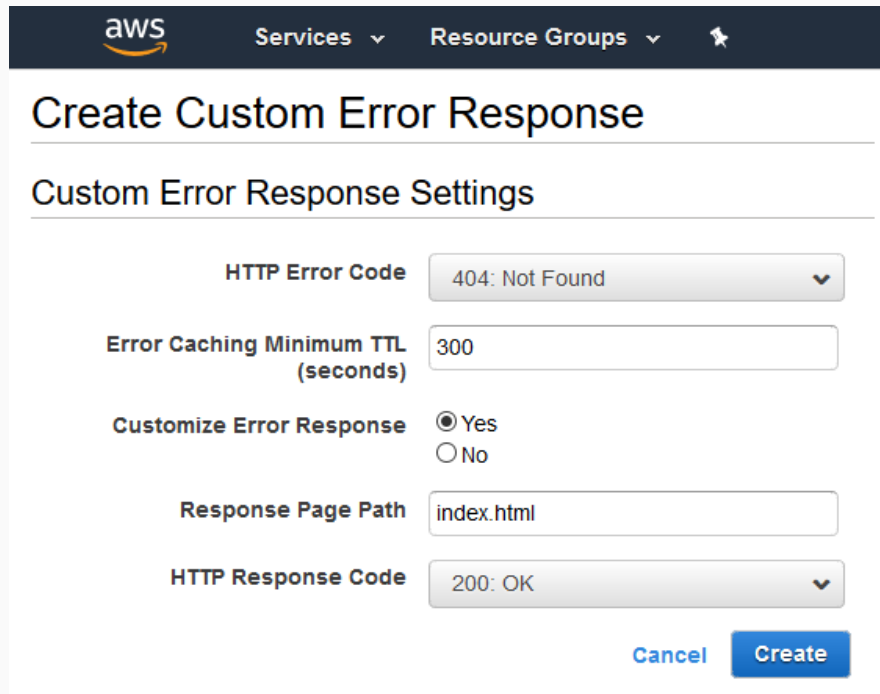
	Delivery Method	Id	C	Origin	CNAMEs	Status	State	Last Modified
<input checked="" type="checkbox"/>	Web	E	d	serverless-webapp-bucket.s3.amazonaws.com	-	In Progress	Enabled	2018-04-22 12:21 UTC+2

Viewing 1 to 1

176/187

STEP 14: OPTIMIZE LATENCY WITH CLOUDFRONT

In the distribution detail page add a custom error response as shown in the picture below to make sure 404 errors are handled by the application.



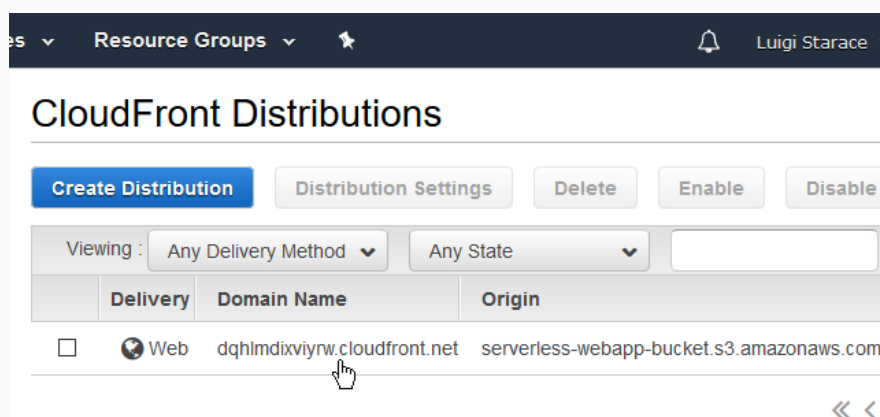
The screenshot shows the 'Create Custom Error Response' page in the AWS console. The page has a dark header with the AWS logo and navigation links for 'Services' and 'Resource Groups'. The main heading is 'Create Custom Error Response'. Below it is a section titled 'Custom Error Response Settings'. The settings include: 'HTTP Error Code' set to '404: Not Found', 'Error Caching Minimum TTL (seconds)' set to '300', 'Customize Error Response' with 'Yes' selected, 'Response Page Path' set to 'index.html', and 'HTTP Response Code' set to '200: OK'. At the bottom right are 'Cancel' and 'Create' buttons.

HTTP Error Code	404: Not Found
Error Caching Minimum TTL (seconds)	300
Customize Error Response	<input checked="" type="radio"/> Yes <input type="radio"/> No
Response Page Path	index.html
HTTP Response Code	200: OK

177/187

STEP 14: OPTIMIZE LATENCY WITH CLOUDFRONT

Now you can visit the application from the cloudfront URL



The screenshot shows the 'CloudFront Distributions' page in the AWS console. The page has a dark header with navigation links for 'es' and 'Resource Groups', and a user profile for 'Luigi Starace'. The main heading is 'CloudFront Distributions'. Below the heading are buttons for 'Create Distribution', 'Distribution Settings', 'Delete', 'Enable', and 'Disable'. There is a 'Viewing' section with filters for 'Any Delivery Method' and 'Any State'. Below this is a table with columns 'Delivery', 'Domain Name', and 'Origin'. The table contains one row with a checkbox, a globe icon, the domain 'dqhlmdixviyrw.cloudfront.net', and the origin 'serverless-webapp-bucket.s3.amazonaws.com'. A hand cursor is pointing at the domain name. At the bottom right are navigation arrows.

	Delivery	Domain Name	Origin
<input type="checkbox"/>	Web	dqhlmdixviyrw.cloudfront.net	serverless-webapp-bucket.s3.amazonaws.com

178/187

TAKE HOME MESSAGES

TAKE HOME MESSAGES

- Cloud computing and service models
- AWS
- Deploy a “classic” web application on AWS
- FaaS and serverless computing
- Build and deploy a serverless one-page web application on AWS

SECURITY RECOMMENDATIONS

- Be **very** careful not to expose your IAM credentials;
- Enforce the least privilege principle: each user should only be able to access the minimum resources necessary to fulfill its purpose.

SECURITY RECOMMENDATIONS

- The very second you expose your credentials to the public, some bot may use them to spin up large numbers of EC2 instances. If that happens, the billing might be a scary surprise!
- Tools like the AWS-developed `git-secrets` help avoiding the exposure of IAM credentials

[⏪ Back to the tutorial](#)

IAM best practices [🔗 web](#)

Git Secrets - Github repository [🔗 web](#)

NICE READINGS I

- [Jan16] Badri Janakiraman. *Serverless*. June 20, 2016. URL: <https://martinfowler.com/bliki/Serverless.html> (visited on 05/21/2018).
- [LF14] James Lewis and Martin Fowler. *Microservices: a definition of this new architectural term*. Mar. 25, 2014. URL: <https://martinfowler.com/articles/microservices.html> (visited on 05/21/2018).
- [Rob16] Mike Roberts. *Serverless Architectures*. Apr. 6, 2016. URL: <https://martinfowler.com/articles/serverless.html> (visited on 05/21/2018).

NICE READINGS II

- [Rus16] Mark Russinovich. *Microservices: An application revolution powered by the cloud*. Mar. 17, 2016. URL: <https://azure.microsoft.com/it-it/blog/microservices-an-application-revolution-powered-by-the-cloud/> (visited on 05/21/2018).
- [Ser] Serverless inc. *Serverless guide*. URL: <https://github.com/serverless/guide> (visited on 05/21/2018).

REFERENCES I

- [Amaa] Amazon Web Services. *What is cloud computing?*. URL: <https://aws.amazon.com/what-is-cloud-computing/> (visited on 03/30/2018).
- [Amab] Inc. Amazon Web Services. *Set up a Continuous Deployment Pipeline using AWS CodePipeline*. URL: <https://aws.amazon.com/it/getting-started/tutorials/continuous-deployment-pipeline/> (visited on 06/10/2018).

REFERENCES II

- [Amac] Inc. Amazon Web Services. *Tutorial: Create a Four-Stage Pipeline*. URL: <https://docs.aws.amazon.com/codepipeline/latest/userguide/tutorials-four-stage-pipeline.html> (visited on 06/10/2018).
- [Ama17] Inc. Amazon Web Services. *Practicing Continuous Integration and Continuous Delivery on AWS*. Tech. rep. June 2017. URL: <https://d1.awsstatic.com/whitepapers/DevOps/practicing-continuous-integration-continuous-delivery-on-AWS.pdf> (visited on 06/01/2018).

REFERENCES III

- [AWS] AWS. *AWS Step Functions*. URL: https://aws.amazon.com/step-functions/?nc1=f_ls (visited on 05/01/2018).
- [Gar17] Gartner. *Gartner Forecasts Worldwide Public Cloud Services Revenue to Reach \$260 Billion in 2017*. Oct. 12, 2017. URL: <https://www.gartner.com/newsroom/id/3815165> (visited on 03/30/2018).

REFERENCES IV

- [Syn18] Synergy Research Group. *Cloud Growth Rate Increases; Amazon, Microsoft & Google all Gain Market Share*. Feb. 2, 2018. URL: <https://www.srgresearch.com/articles/cloud-growth-rate-increases-amazon-microsoft-google-all-gain-market-share> (visited on 03/30/2018).