Karam MANSOUR, Sofian YAHYAOUI, Ayyoub ZEBDA

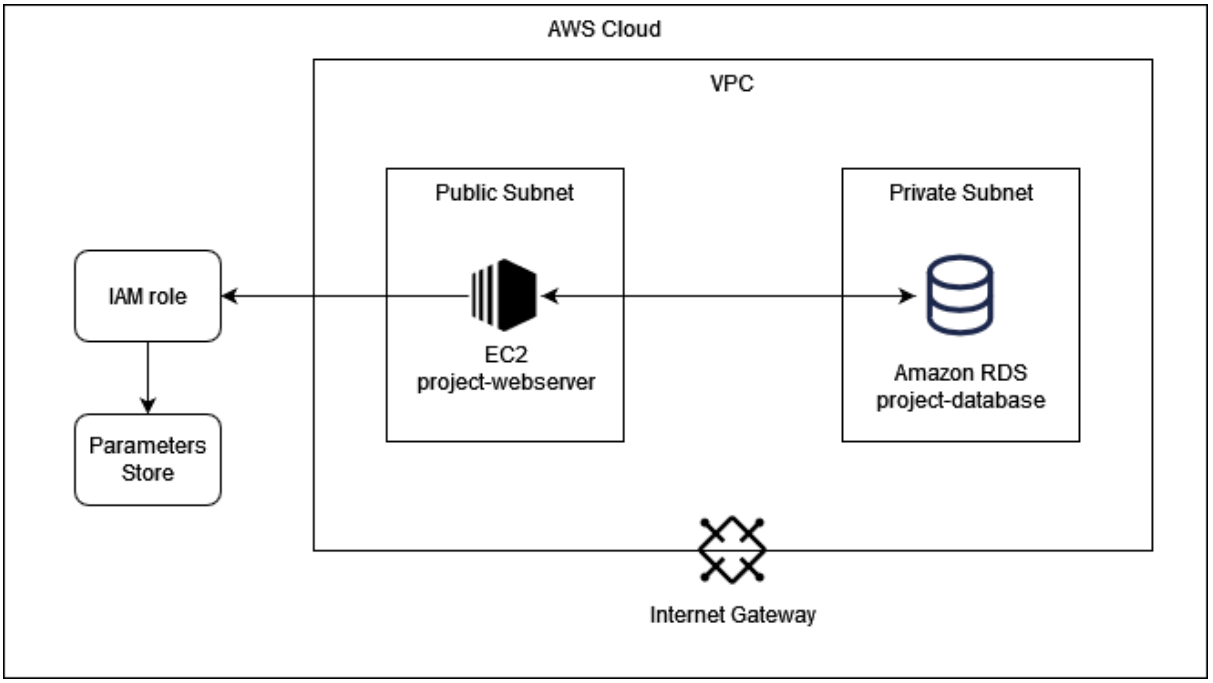
Capstone Project

As a reminder, the project required us to set up a PHP web application through an EC2 instance, which could communicate with a MySQL database managed by the Amazon RDS service.

App Deployment

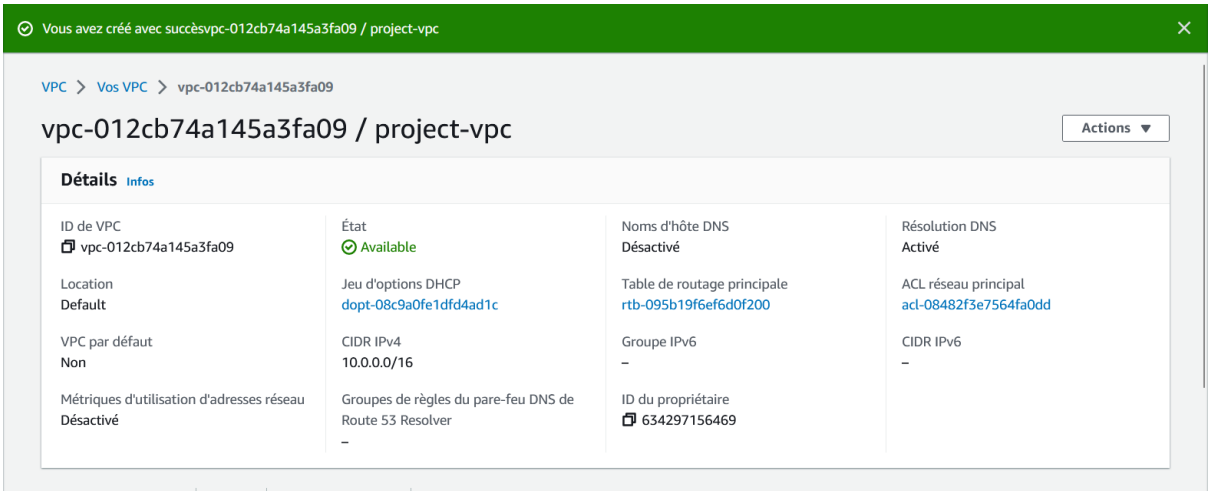
Architecture

Here's a very simplified version of our architecture. We'll go into more detail in the rest of the report.

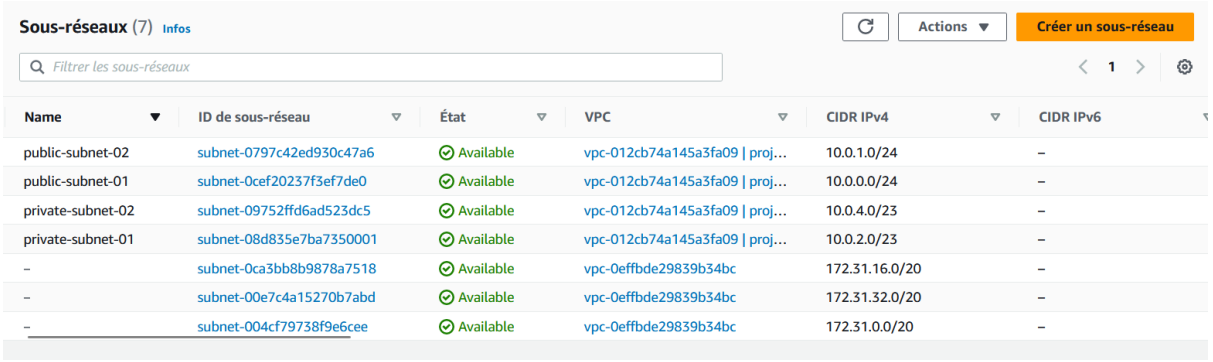


Step 1: Setup the cloud environment

So we started by creating a VPC to provide a private, secure network for our EC2 instance. This will allow us to control access to our future instance, and protect it from outside attacks.

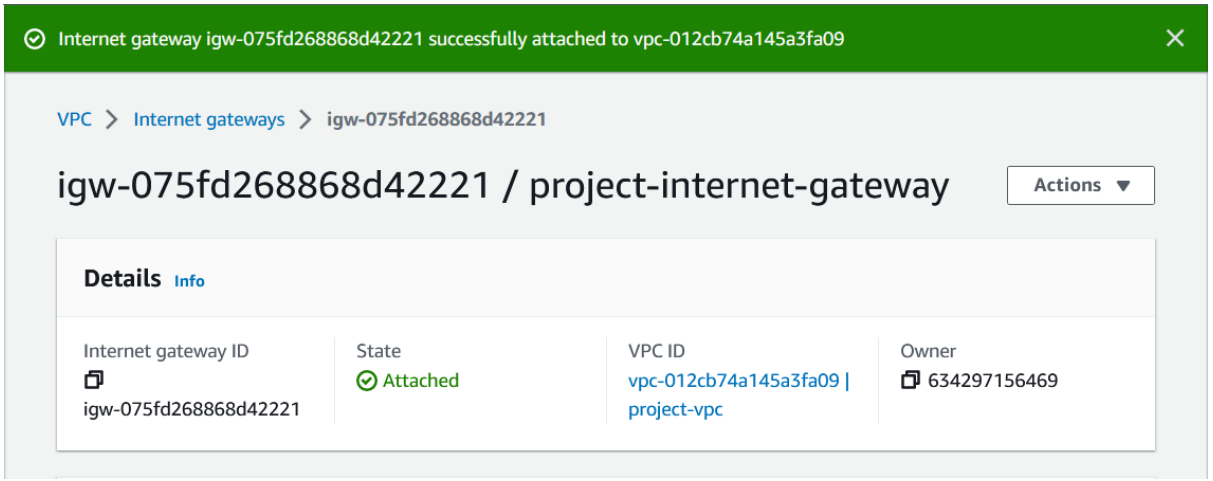


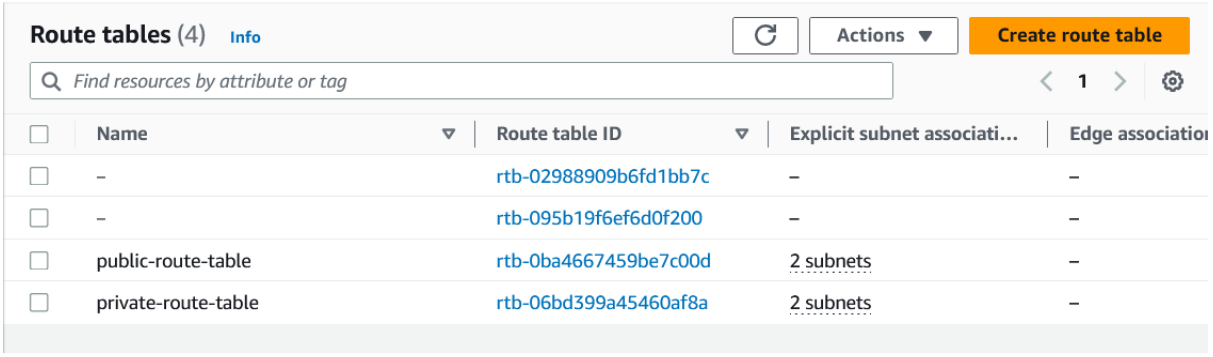
We then created two subnets, one public and one private, linked to our initial VPC (the ones we're using here for our project are the "public-subnet-1" and "private-subnet-1" subnets).



Name	ID de sous-réseau	État	VPC	CIDR IPv4	CIDR IPv6
public-subnet-02	subnet-0797c42ed930c47a6	Available	vpc-012cb74a145a3fa09 | proj...	10.0.1.0/24	–
public-subnet-01	subnet-0cef20237f3ef7de0	Available	vpc-012cb74a145a3fa09 | proj...	10.0.0.0/24	–
private-subnet-02	subnet-09752ffd6ad523dc5	Available	vpc-012cb74a145a3fa09 | proj...	10.0.4.0/23	–
private-subnet-01	subnet-08d835e7ba7350001	Available	vpc-012cb74a145a3fa09 | proj...	10.0.2.0/23	–
–	subnet-0ca3bb8b9878a7518	Available	vpc-0effbde29839b34bc	172.31.16.0/20	–
–	subnet-00e7c4a15270b7abd	Available	vpc-0effbde29839b34bc	172.31.32.0/20	–
–	subnet-004cf79738f9e6cee	Available	vpc-0effbde29839b34bc	172.31.0.0/20	–

Of course, we also built an internet gateway to allow external access to our application (especially useful when using Cloud9), and set up routing tables to go with it all.





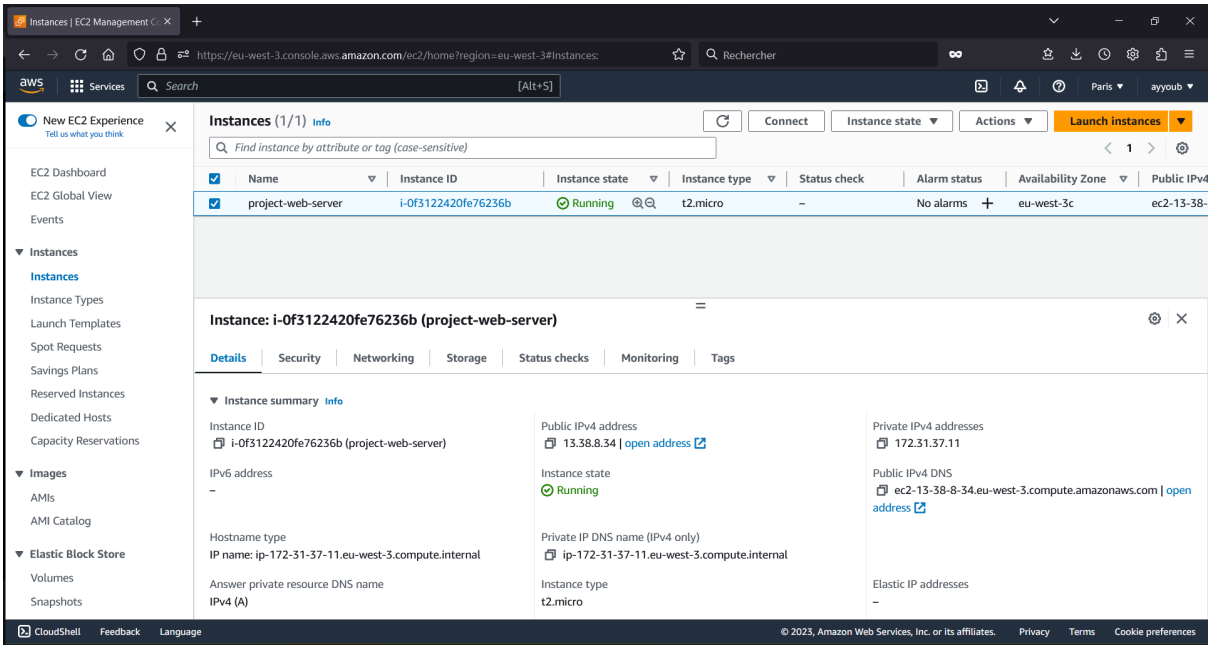
Here, the routing tables each contain the two public/private subnets (left over from previous projects and tests), but once again we ended up using only one of each, as we had defined in the architecture.

Step 2: Setup the EC2 instance that will be used to deploy our application

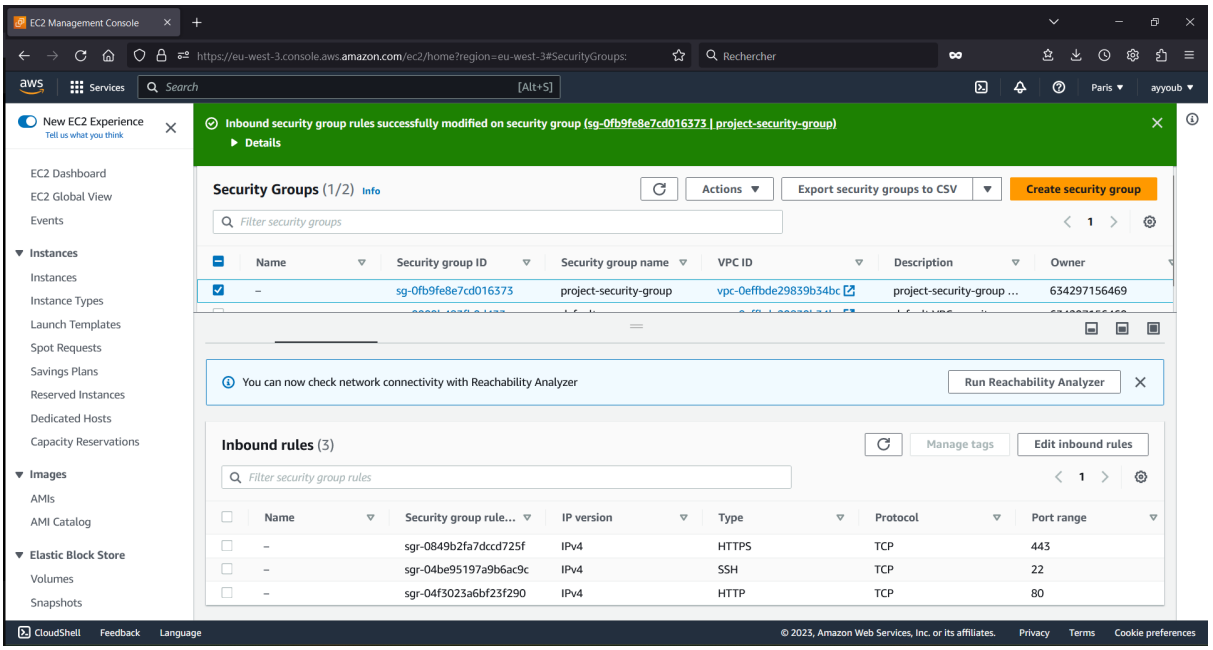
The EC2 instance is the core element of our project. It is what will allow us to host our application and allow any user to access it once it is deployed.

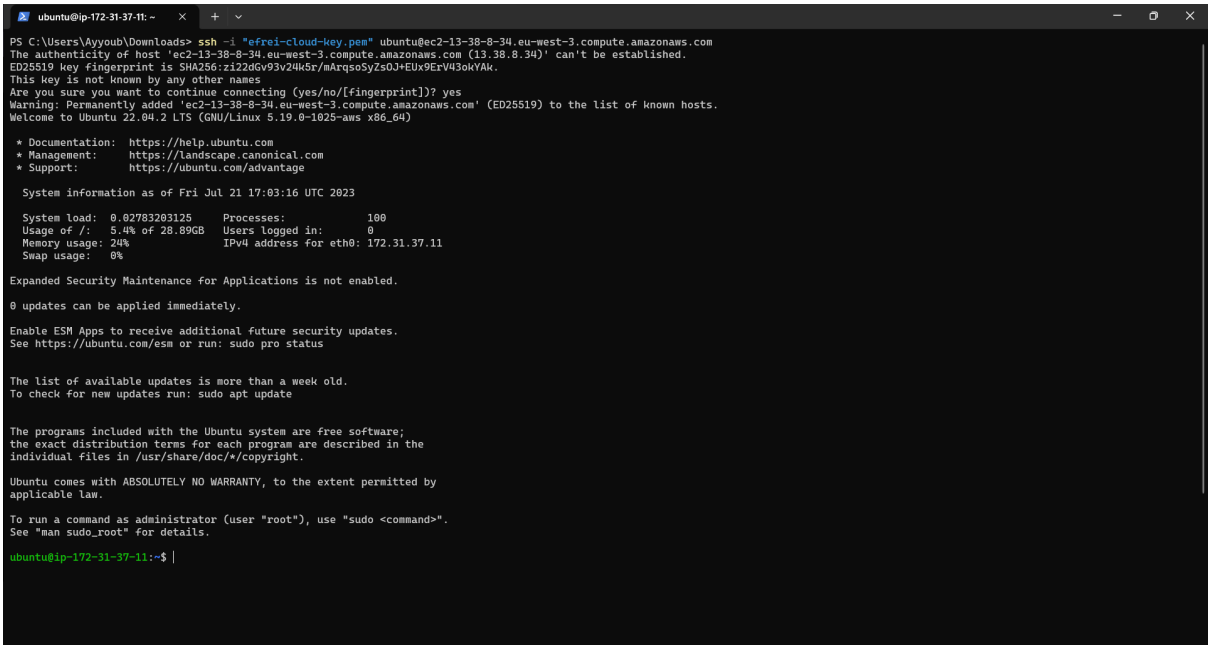
The original plan was to create an EC2 instance through Cloud9. However, we were unable to get the instance to work, despite carefully checking our environment. We therefore decided to create the EC2 instance ourselves and access it through our personal terminal via SSH.

We also encountered a problem when creating an instance under Amazon Linux. The instance did not recognize certain commands or did not have access to certain modules, even after updating the default module library. This is why we created our instance based on Ubuntu.



We also took the occasion to create a security group authorizing access to our instance from any IP, via SSH, HTTP and HTTPS.

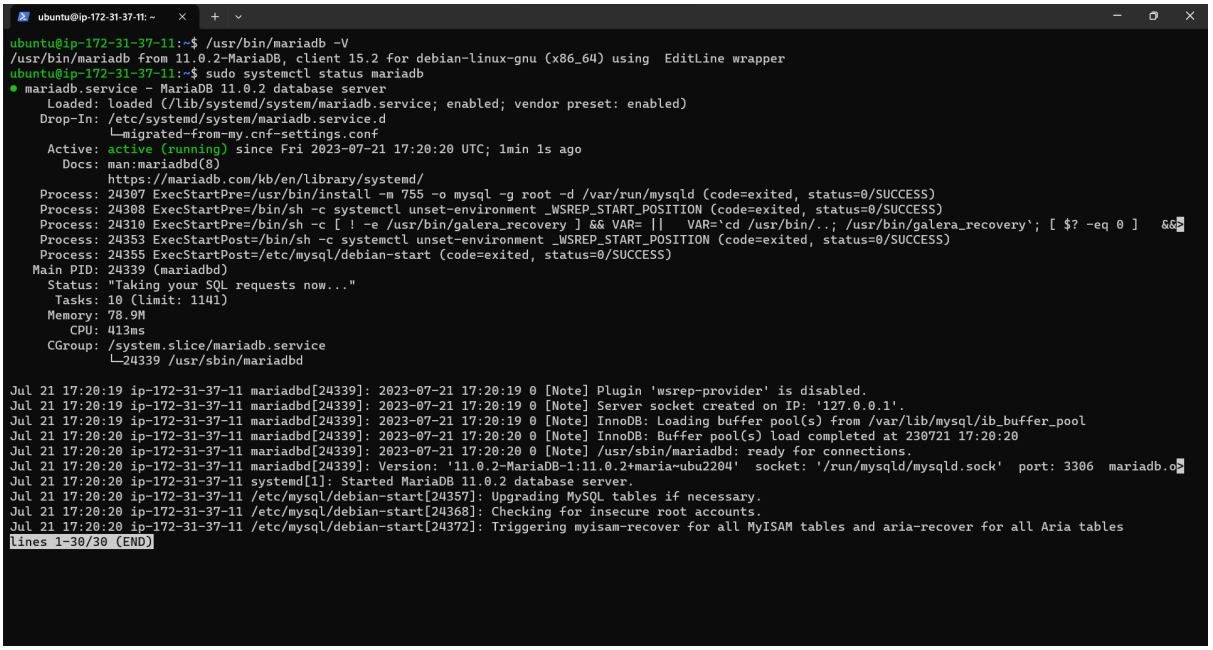




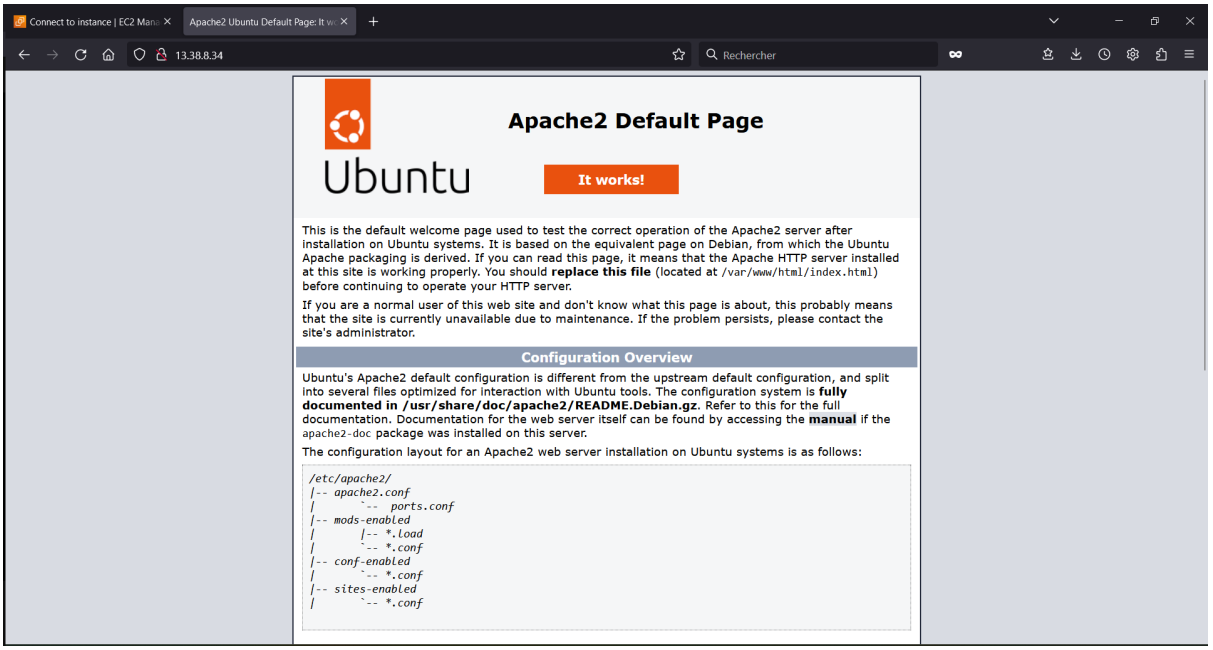
Once connected to our instance using the command `ssh -i {path/to/key.pem} ubuntu@{ec2-public-dns}`, we executed the following commands to initialize our linux environment with the correct modules (the commands are not exactly the same as in the statement, as they were run on Ubuntu and not Amazon Linux):

```
sudo apt-get update -y
sudo apt-get upgrade -y
sudo apt-get install apache2 -y
sudo apt-get install php -y
sudo apt-get install mariadb-server -y
```

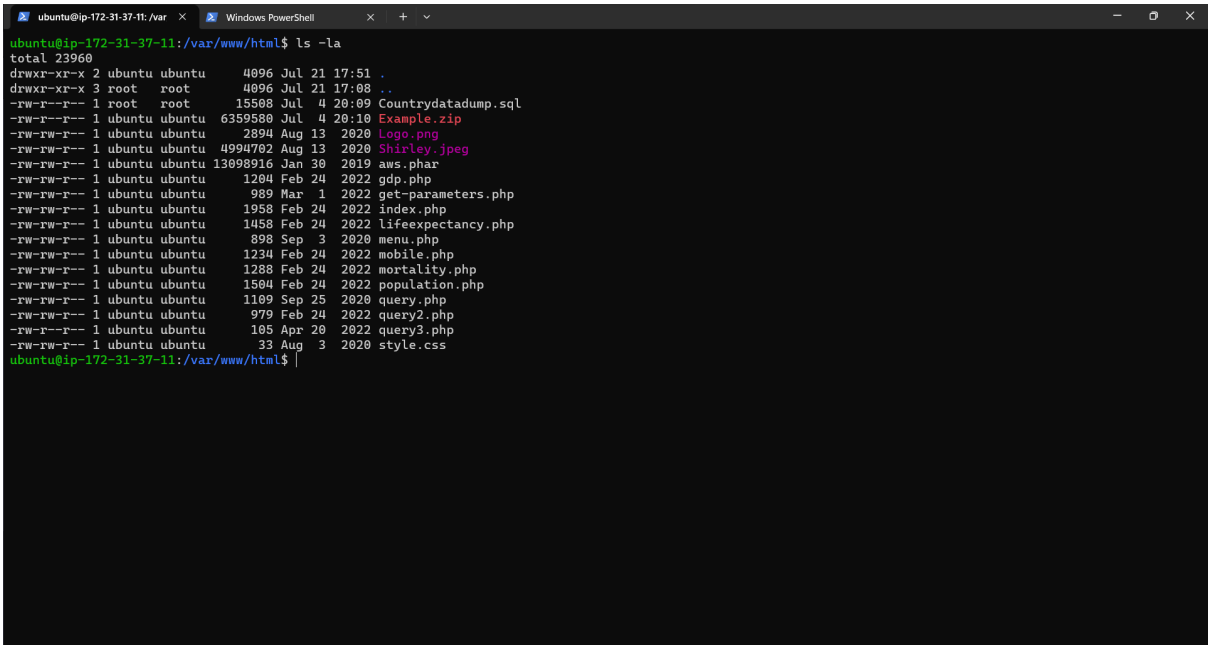
Example: checking MariaDB installation

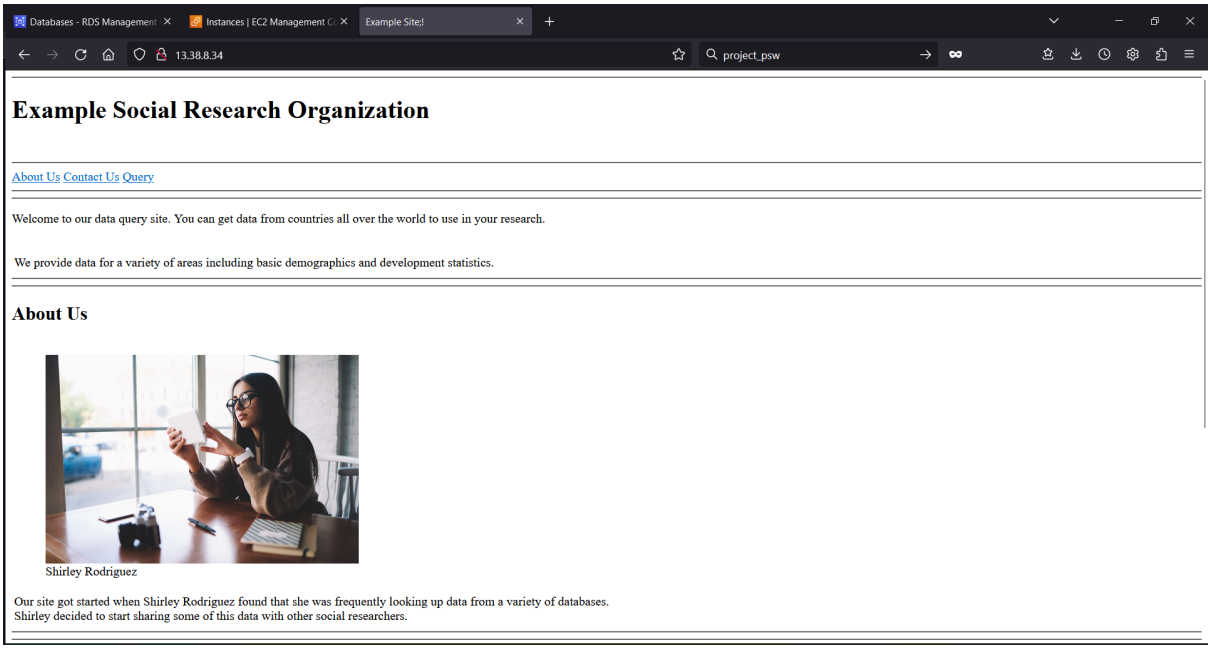


Apache default home page, our server is working!



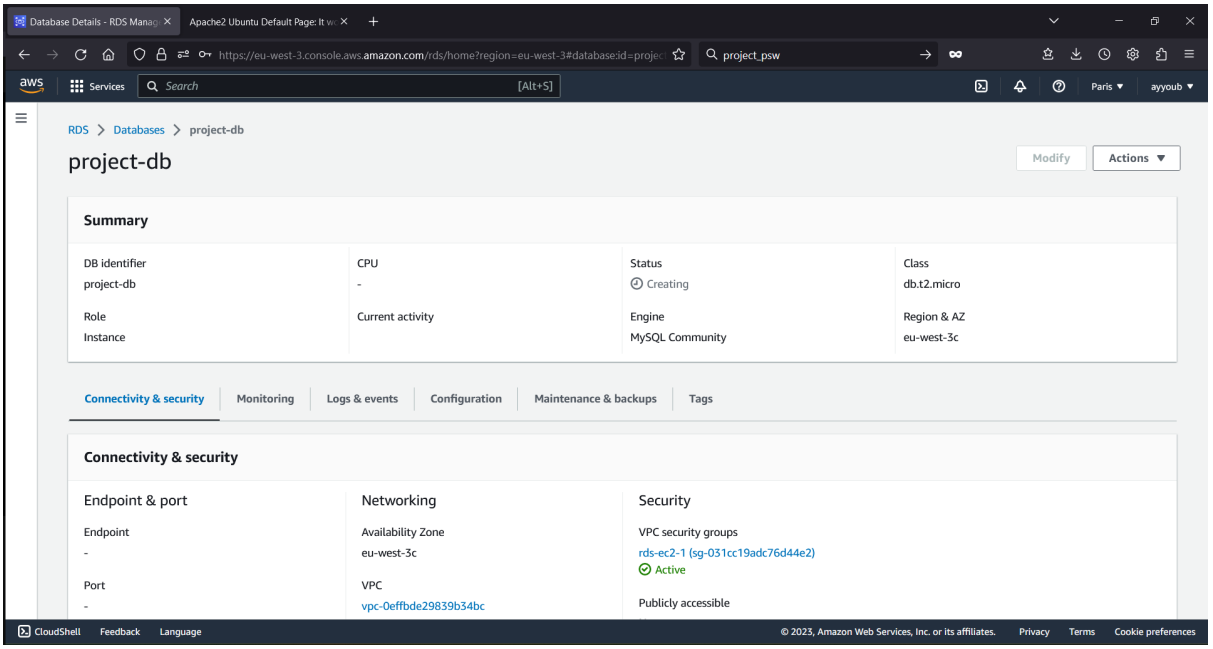
Next, we upload our files to the server using the `wget` command, then extract all the files into the `/var/www/html/` folder, taking care to delete the `index.html` file created by default when the Apache server was installed.





Step 3: Creating the Amazon RDS database

To store the data we wanted our web application to access, the next step was to create a database using the Amazon RDS service. We directly checked the option to connect it to our "project-web-server" EC2 instance.

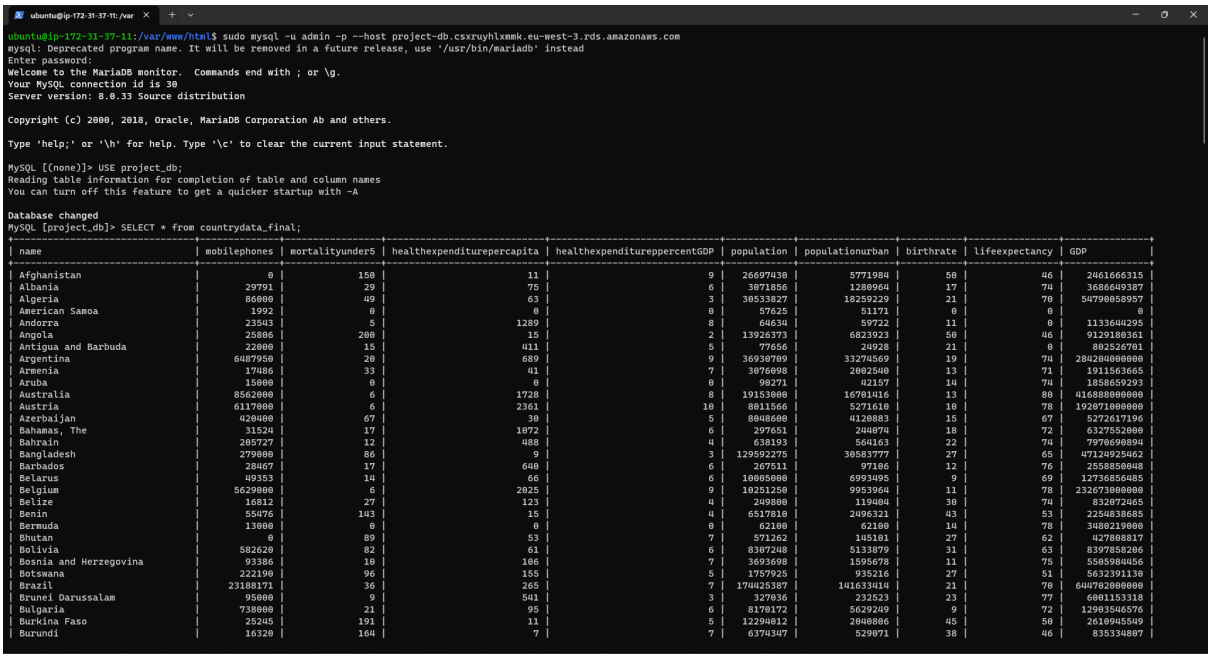


(This screenshot was taken before AWS could finish instantiating the DB, which is why you can't see any endpoints, ports, etc. yet.)

Via the terminal of our EC2 instance, we could then connect to our DB with the following command:

```
sudo mysql -u {username} -p --host {endpoint}
```

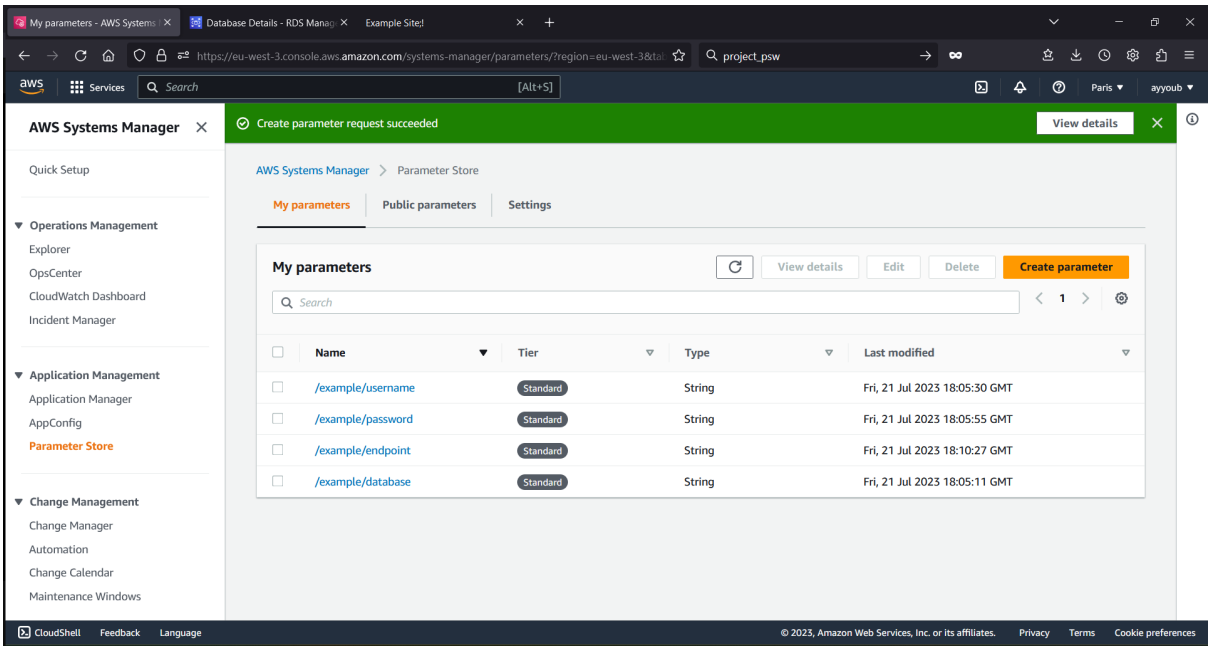
And execute the database initialisation script to create our table:

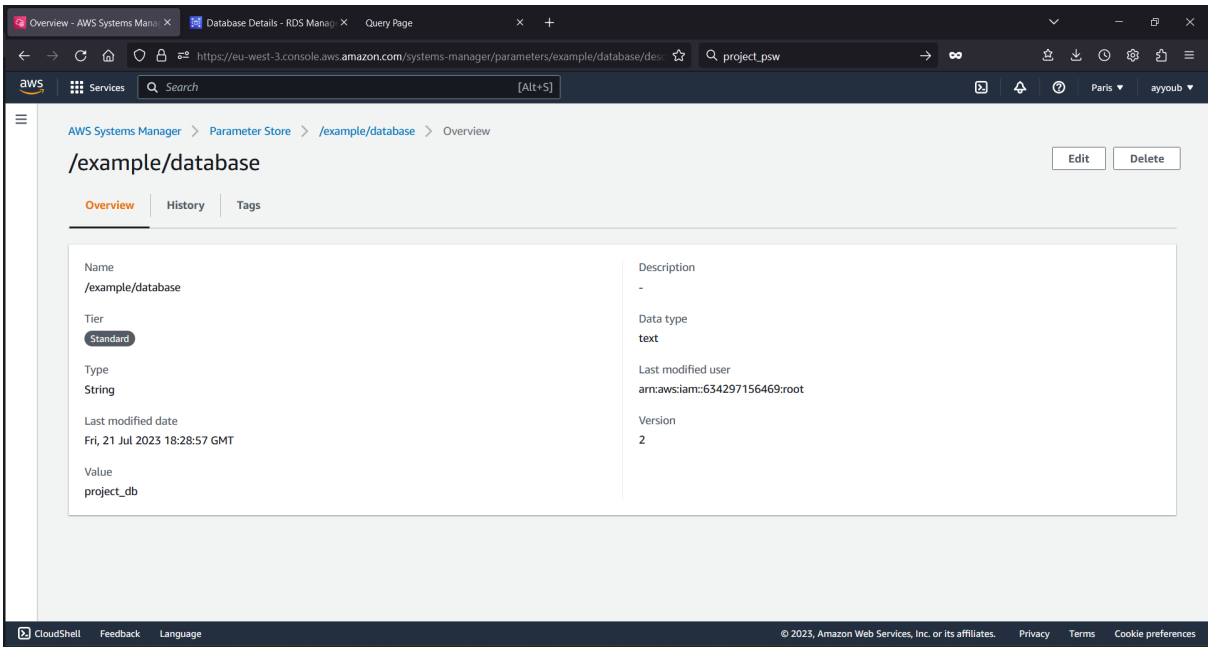


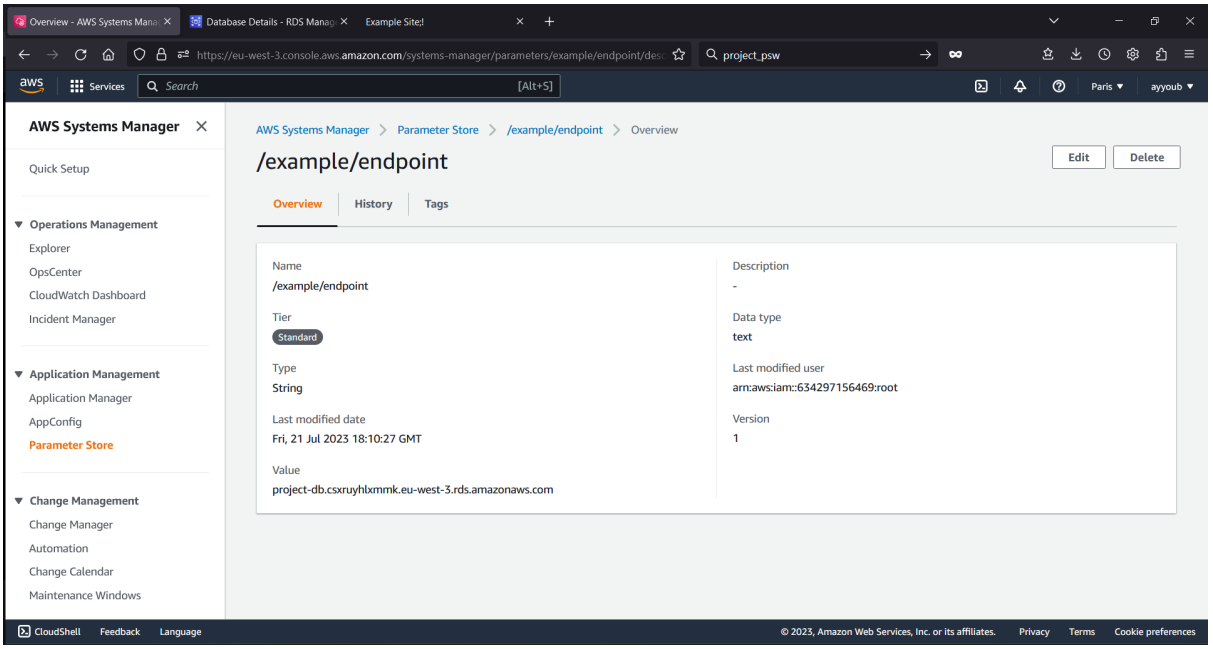
```
ubuntu@ip-172-31-37-11:/var/www/html$ sudo mysql -u admin -p --host project-db.csxruyhlxmmk.eu-west-3.rds.amazonaws.com
mysql: Deprecated program name. It will be removed in a future release, use '/usr/bin/mariadb' instead
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 30
Server version: 8.0.33 Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> USE project_db;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MySQL [project_db]> SELECT * from countrydata_final;
+--------------------------------+--------------+-----------------+-------------------------+-----------------------------+------------+---------------+-----------+----------------+---------------+
| name                           | mobilephones | mortalityunder5 | healthexpenditurepercapita | healthexpenditureppercentGDP | population | populationurban | birthrate | lifeexpectancy | GDP           |
+--------------------------------+--------------+-----------------+-------------------------+-----------------------------+------------+---------------+-----------+----------------+---------------+
| Afghanistan                    |            0 |             150 |                      11 |                           9 |   26697430 |       5771984 |        50 |             46 |    2461666315 |
| Albania                        |        29791 |              29 |                      75 |                           6 |    3071856 |       1280964 |        17 |             74 |    3686649387 |
| Algeria                        |        86000 |              49 |                      63 |                           3 |   30533827 |      18259229 |        21 |             70 |   54790058957 |
| American Samoa                 |         1992 |               0 |                       0 |                           0 |      57625 |         51171 |         0 |              0 |             0 |
| Andorra                        |        23543 |               5 |                    1289 |                           8 |      64634 |         59722 |        11 |              0 |    1133644295 |
| Angola                         |        25806 |             200 |                      15 |                           2 |   13926373 |       6823923 |        50 |             46 |    9129180361 |
| Antigua and Barbuda            |        22000 |              15 |                     411 |                           5 |      77656 |         24928 |        21 |              0 |     802526701 |
| Argentina                      |      6487950 |              20 |                     689 |                           9 |   36930709 |      33274569 |        19 |             74 |  28420400000  |
| Armenia                        |        17486 |              33 |                      41 |                           7 |    3076098 |       2002540 |        13 |             71 |    1911563665 |
| Aruba                          |        15000 |               0 |                       0 |                           0 |      90271 |         42157 |        14 |             74 |    1858659293 |
| Australia                      |      8562000 |               6 |                    1728 |                           8 |   19153000 |      16701416 |        13 |             80 |  416888000000 |
| Austria                        |      6117000 |               6 |                    2361 |                          10 |    8011566 |       5271610 |        10 |             78 |  192071000000 |
| Azerbaijan                     |       420400 |              67 |                      30 |                           5 |    8048600 |       4120883 |        15 |             67 |    5272617196 |
| Bahamas, The                   |        31524 |              17 |                    1072 |                           6 |     297651 |        244074 |        18 |             72 |    6327552000 |
| Bahrain                        |       205727 |              12 |                     488 |                           4 |     638193 |        564163 |        22 |             74 |    7970690894 |
| Bangladesh                     |       279000 |              86 |                       9 |                           3 |  129592275 |      30583777 |        27 |             65 |   47124925462 |
| Barbados                       |        28467 |              17 |                     640 |                           6 |     267511 |         97106 |        12 |             76 |    2558850048 |
| Belarus                        |        49353 |              14 |                      66 |                           6 |   10065000 |       6993495 |         9 |             69 |   12736856485 |
| Belgium                        |      5629000 |               6 |                    2025 |                           9 |   10251250 |       9953964 |        11 |             78 |  232673000000 |
| Belize                         |        16812 |              27 |                     123 |                           4 |     249800 |        119404 |        30 |             74 |     832072465 |
| Benin                          |        55476 |             143 |                      15 |                           4 |    6517810 |       2496321 |        43 |             53 |    2254838685 |
| Bermuda                        |        13000 |               0 |                       0 |                           0 |      62100 |         62100 |        14 |             78 |    3480219000 |
| Bhutan                         |            0 |              89 |                      53 |                           7 |     571262 |        145101 |        27 |             62 |     427808817 |
| Bolivia                        |       582620 |              82 |                      61 |                           6 |    8307248 |       5131879 |        31 |             63 |    8397858286 |
| Bosnia and Herzegovina         |        93386 |              10 |                     106 |                           7 |    3693698 |       1595678 |        11 |             75 |    5505984456 |
| Botswana                       |       222190 |              96 |                     155 |                           5 |    1757925 |        935216 |        27 |             51 |    5632391130 |
| Brazil                         |     23188171 |              36 |                     265 |                           7 |  174425387 |     141633414 |        21 |             70 |  644702000000 |
| Brunei Darussalam              |        95000 |               9 |                     541 |                           3 |     327036 |        232523 |        23 |             77 |    6001153318 |
| Bulgaria                       |       738000 |              21 |                      95 |                           6 |    8170172 |       5629249 |         9 |             72 |   12903546576 |
| Burkina Faso                   |        25245 |             191 |                      11 |                           5 |   12294012 |       2040806 |        45 |             50 |    2610945549 |
| Burundi                        |        16320 |             164 |                       7 |                           7 |    6374347 |        529071 |        38 |             46 |     835334887 |
```

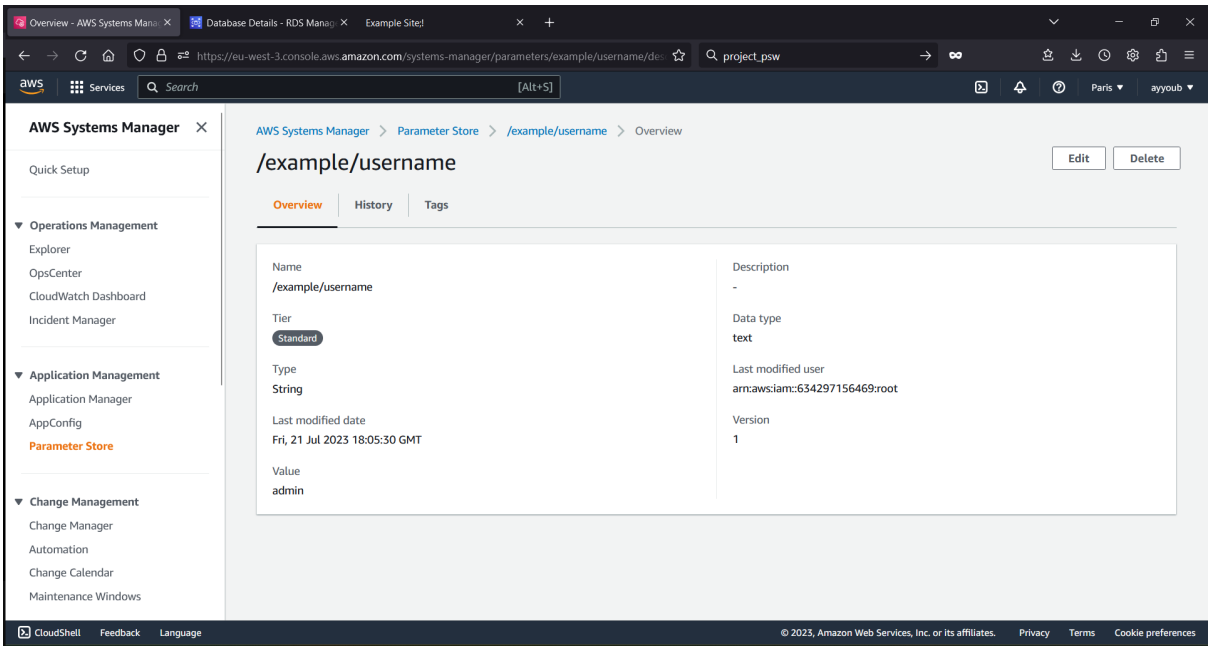
Step 4: Creating the Parameter Manager

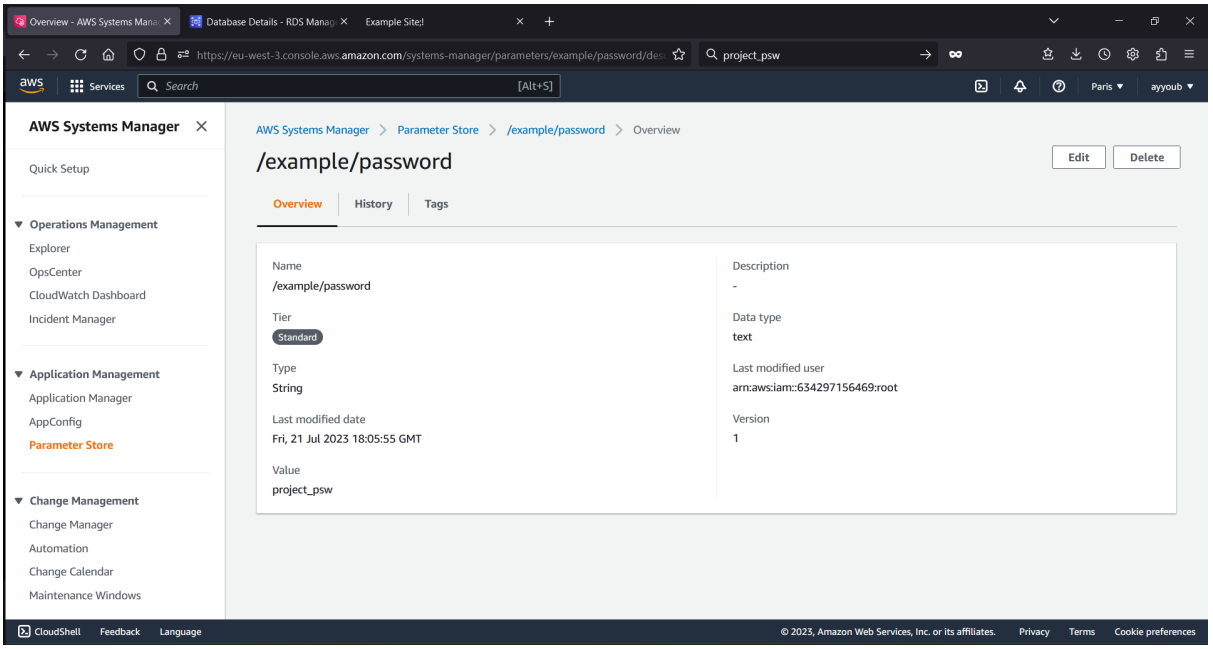
For our web app to access the database, we had to give it access to all its connection parameters, so we used the AWS System Manager to create a parameter registry.






Overview - AWS Systems Mana
Database Details - RDS Manag
Example Site!
https://eu-west-3.console.aws.amazon.com/systems-manager/parameters/example/endpoint/des
project_psw
aws
Services
Search
[Alt+S]
Paris
ayyoub
AWS Systems Manager
Quick Setup
Operations Management
Explorer
OpsCenter
CloudWatch Dashboard
Incident Manager
Application Management
Application Manager
AppConfig
Parameter Store
Change Management
Change Manager
Automation
Change Calendar
Maintenance Windows
AWS Systems Manager > Parameter Store > /example/endpoint > Overview
/example/endpoint
Edit
Delete
Overview
History
Tags
Name
/example/endpoint
Description
-
Tier
Standard
Data type
text
Type
String
Last modified user
arn:aws:iam::634297156469:root
Last modified date
Fri, 21 Jul 2023 18:10:27 GMT
Version
1
Value
project-db.csxruyhlxmmk.eu-west-3.rds.amazonaws.com
CloudShell
Feedback
Language
© 2023, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Overview - AWS Systems Mana
Database Details - RDS Manag
Example Site!
https://eu-west-3.console.aws.amazon.com/systems-manager/parameters/example/username/des
project_psw
aws
Services
Search
[Alt+S]
Paris
ayyoub
AWS Systems Manager
Quick Setup
Operations Management
Explorer
OpsCenter
CloudWatch Dashboard
Incident Manager
Application Management
Application Manager
AppConfig
Parameter Store
Change Management
Change Manager
Automation
Change Calendar
Maintenance Windows
AWS Systems Manager > Parameter Store > /example/username > Overview
/example/username
Edit
Delete
Overview
History
Tags
Name
/example/username
Description
-
Tier
Standard
Data type
text
Type
String
Last modified user
arn:aws:iam::634297156469:root
Last modified date
Fri, 21 Jul 2023 18:05:30 GMT
Version
1
Value
admin
CloudShell
Feedback
Language
© 2023, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences



The final step in making everything work was to define an IAM role and assign it the necessary permissions to allow our EC2 instance to access the defined parameters.

Here are the permissions we implemented:

```
{
      "Version": "2012-10-17",
      "Statement": [
      {
            "Effect": "Allow",
            "Action": ["ssm:GetParameter*"],
            "Resource":
["arn:aws:ssm:${data.aws_region.current.name}:${data.aws_caller_id
entity.current.account_id}:parameter/example/*"]
      }
      ]
}
```

There you go! We have our fully deployed application, which communicates with a MySQL database!

Quiz

IAM Quiz

Q1: Option 3
Q2: Option 1
Q3: Options 3 and 4
Q4: Option 4
Q5: Option 2
Q6: Option 3
Q7: Option 1

Network quiz

Q1: Option 3
Q2: Option 3
Q3: Options 1, 3 and 6
Q4: Option 4

IAM

Question 1:

The specific resources included in the policy are:

- All EC2 instances in the “us-east-1” region with the account ID “123456789012”.
- All objects in the “example-bucket” S3 bucket.

Question 2:

The policy allows access to VPC-related information only in the AWS region “us-west-2”.

Question 3:

The policy allows the following actions on the example-bucket and its objects:

- `s3:GetObject`
- `s3:PutObject`
- `s3:ListBucket`

The condition “StringLike” with the key “s3:prefix” and the values “documents/*” and “images/*” specifies that the user can only access objects that start with the prefixes documents or images. For example, the user could access the objects “./documents/exam.pdf” and “./images/cat.png”, but they could not access the object “./example.txt”.

Question 4:

The policy allows the following actions for IAM users:

- `iam:CreateUser`
- `iam:DeleteUser`

The resource ARNs are constructed using the following format:
`arn:aws:iam::ACCOUNT_ID:user/${aws:username}`

This policy allows the IAM user to create and delete IAM users in the account. It does not allow the user to perform any other actions on IAM users, such as listing users, updating users, or assigning permissions to users.

Question(s) 5:

- The policy grants you access to the AWS IAM service. The actions `iam:Get*` and `iam:List*` allow you to get and list IAM resources, such as users, groups, policies, and roles.
- The policy does not allow us to create any IAM resources. The only actions that are allowed are Get and List. This means that we can view the existing IAM resources, but we cannot create any new ones.
- Three specific actions that the `iam:Get*` action allows:
    - `iam:GetAccountSummary`

    - `iam:GetPolicy`
    - `iam:GetUser`

Question(s) 6:

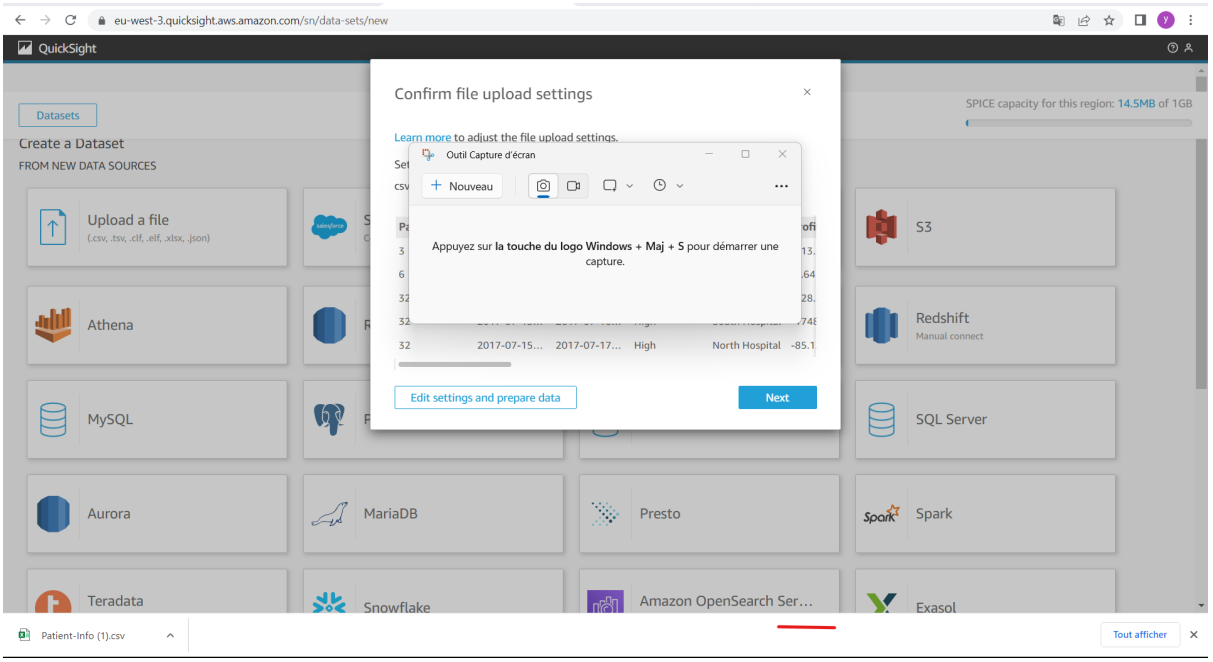
- The policy allows the following actions:
    - `ec2:RunInstances`
    - `ec2:StartInstances`
- The first policy will restrict the access granted by the second policy by denying the `ec2:RunInstances` and `ec2:StartInstances` actions if the instance type is “t2.micro” or “t2.small”. For example, if you tried to run an “t2.micro” instance, the first policy would deny the request, even though the second policy would allow it.
- If the policy included both the statement on the left and the statement in question 2, you could not terminate an m3.xlarge instance that existed in the account. The first policy would deny the ec2:TerminateInstances action if the instance type is t2.micro or t2.small, and the second policy would not allow the action for any other instance types.

Data Visualization With AWS QuickSight

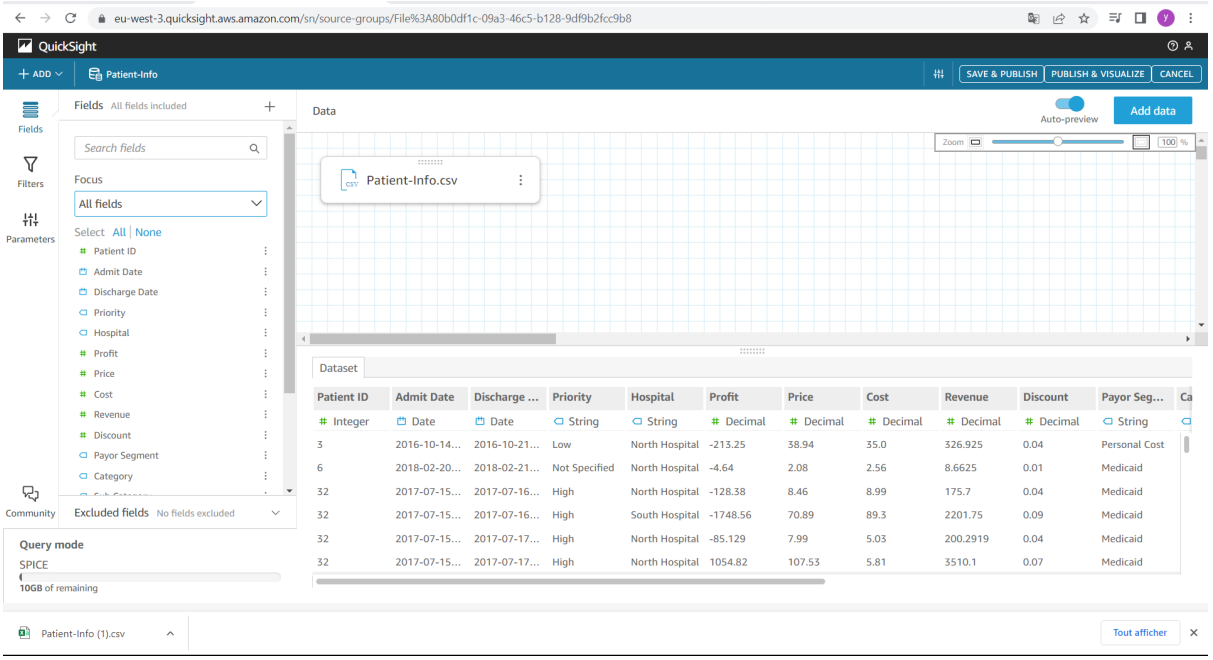
Dashboard Creation:

First step :

We have to enter quicksight and then we have to load our database 4

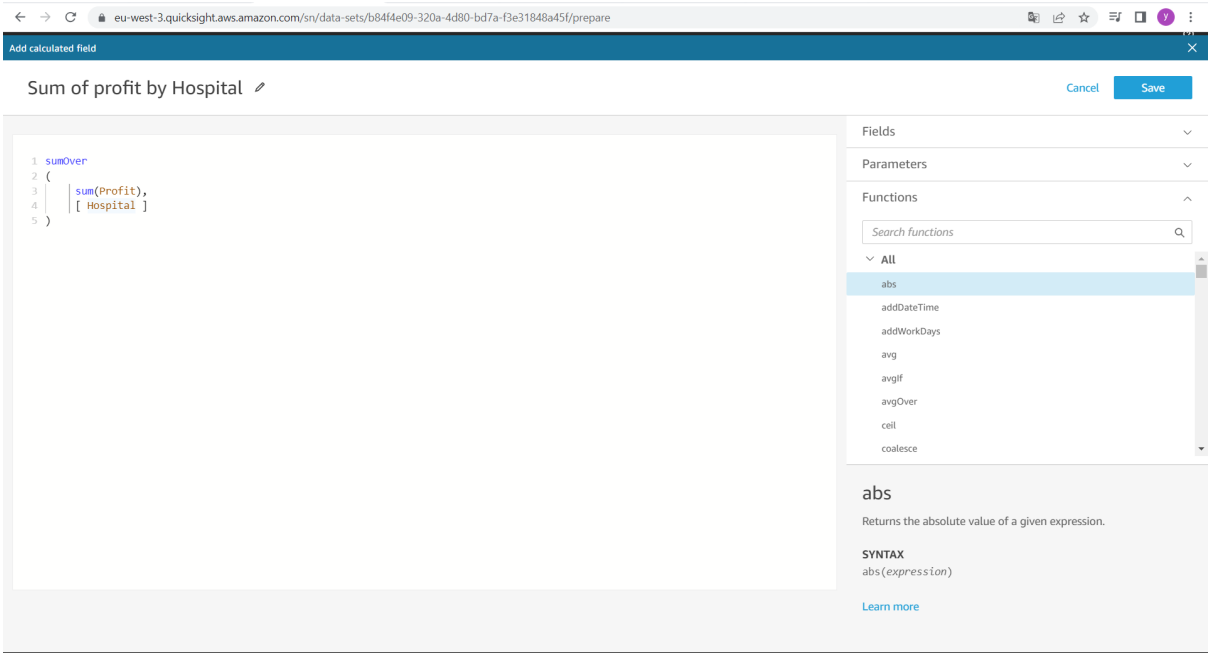


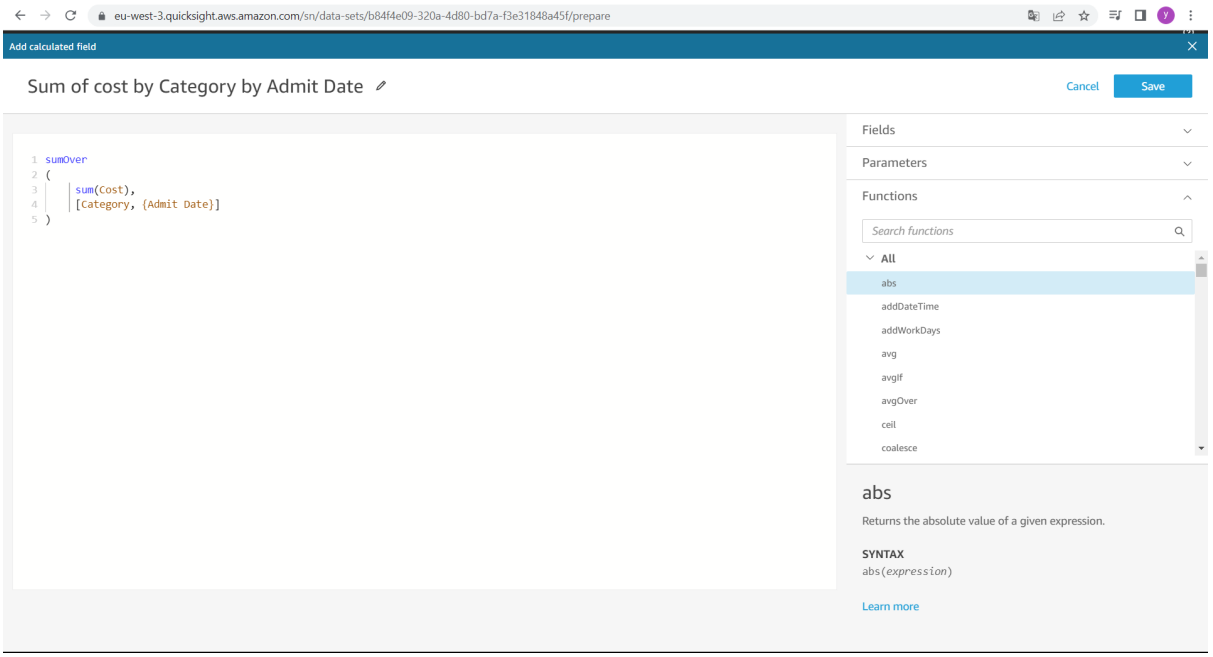
Then we have to edit the database to be sure that we have everything we need in it

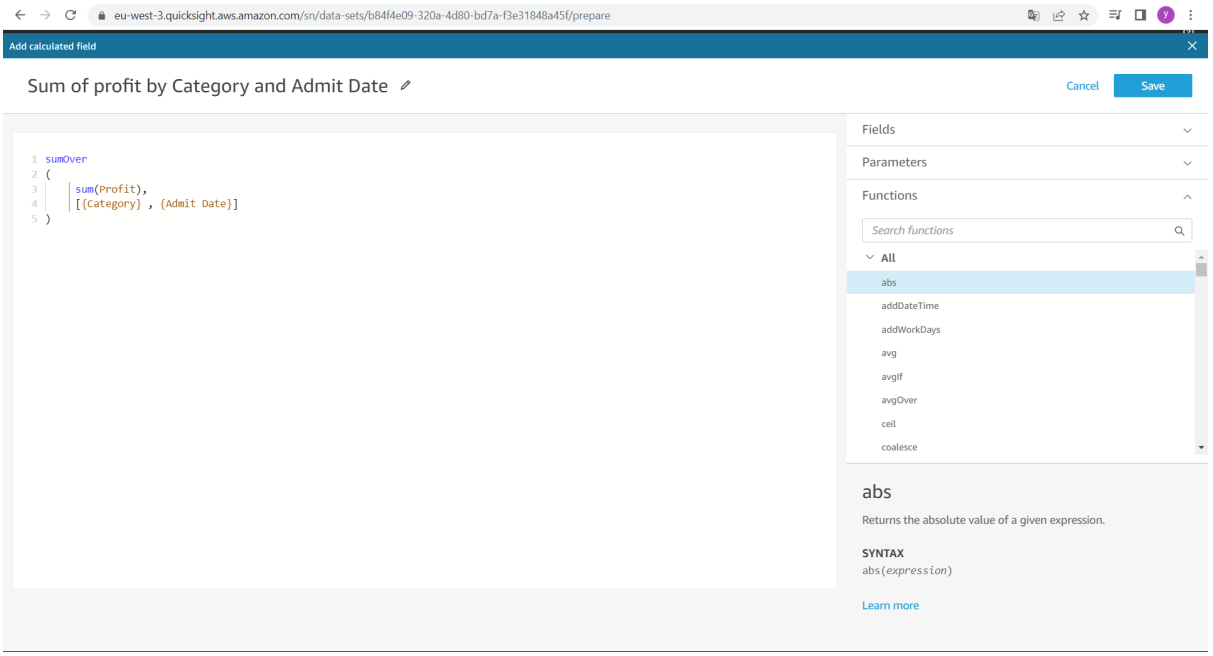


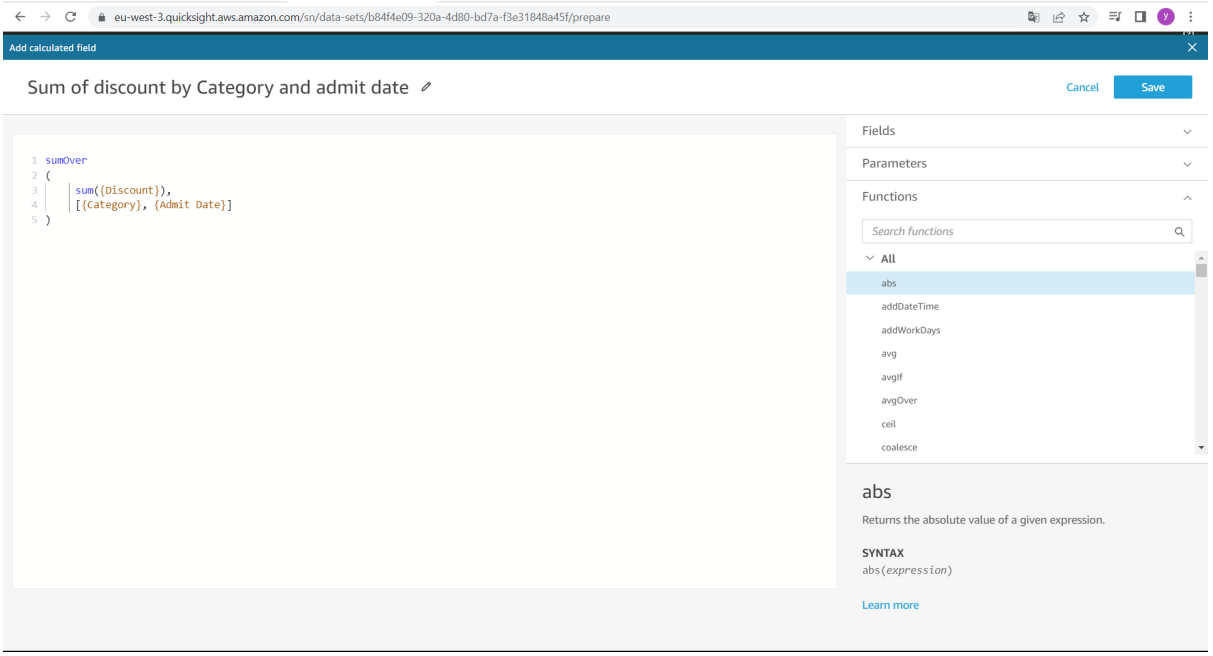
Then what we need to do is to calculate a new field according to the criteria that we need to meet to have all the fields needed to do the dashboard.

All the new field are Sums grouped by another field so the structure of the code is the same for each of them , we use the SumOver function that allow us to do a sum of a field for another specific field



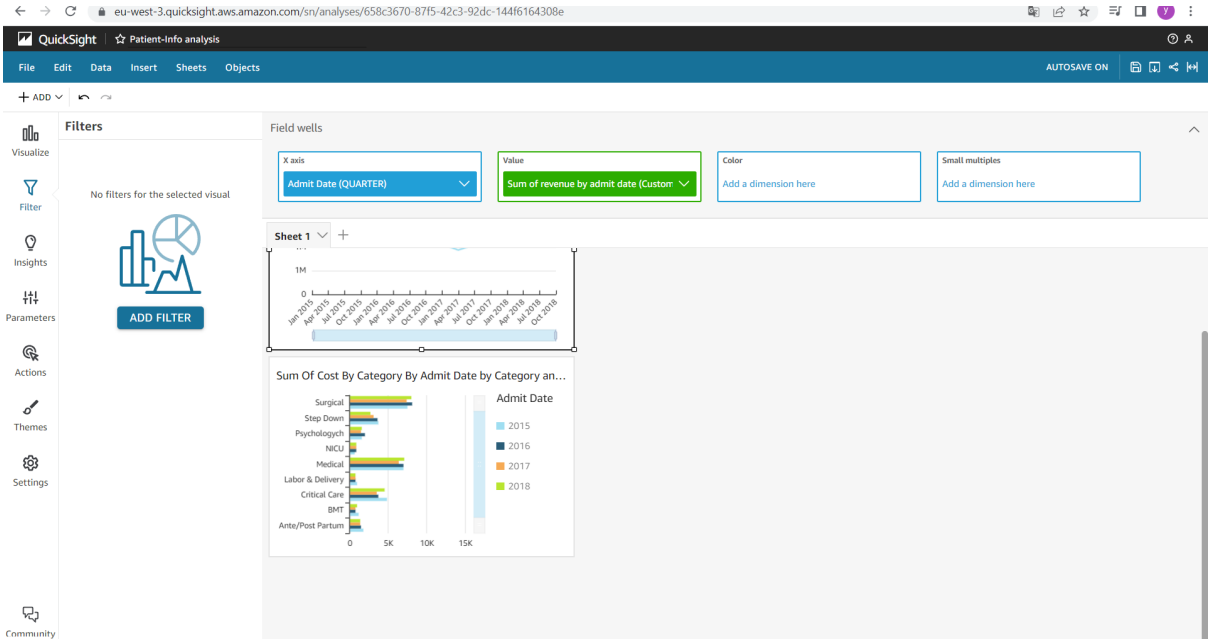




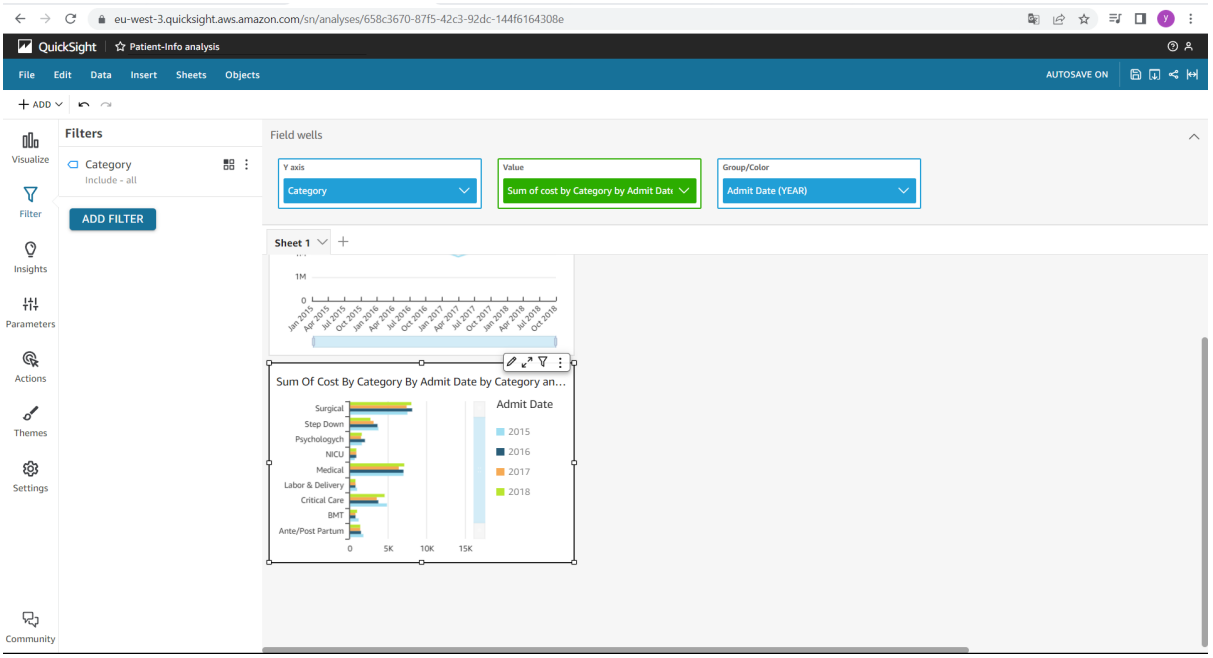


Then now that we have that we can go to the analyze panel.

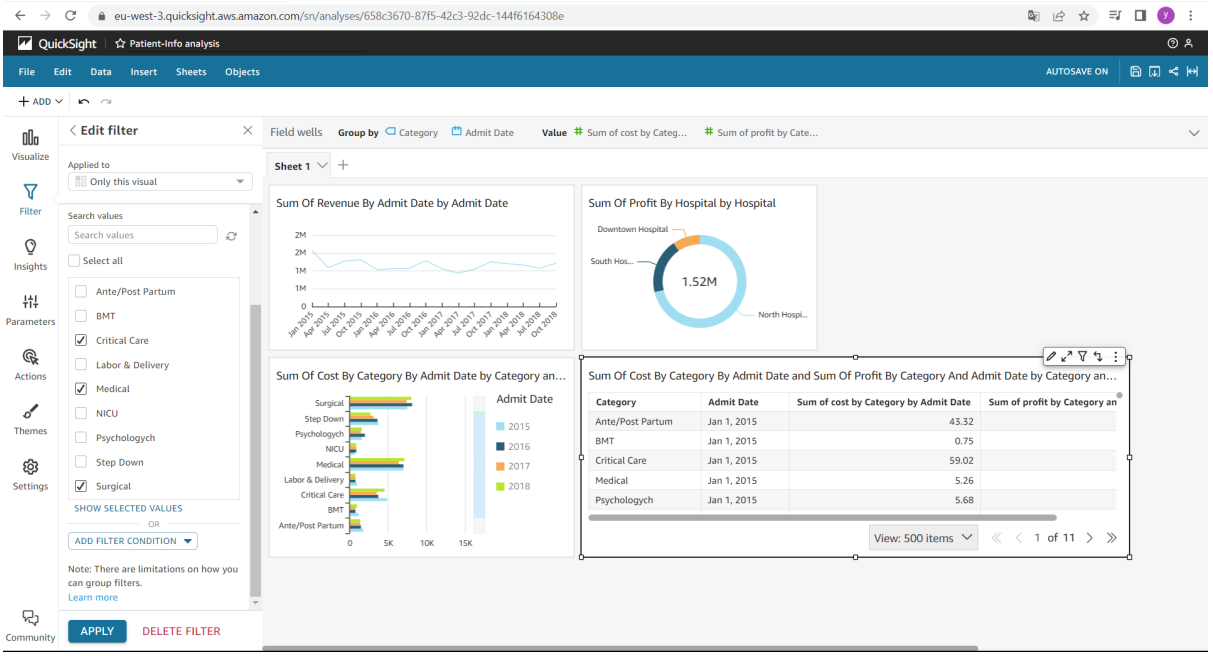
For the first graphic that we can see here, we have to specify the different axis and the scale, the process is the same for all the other graphs .



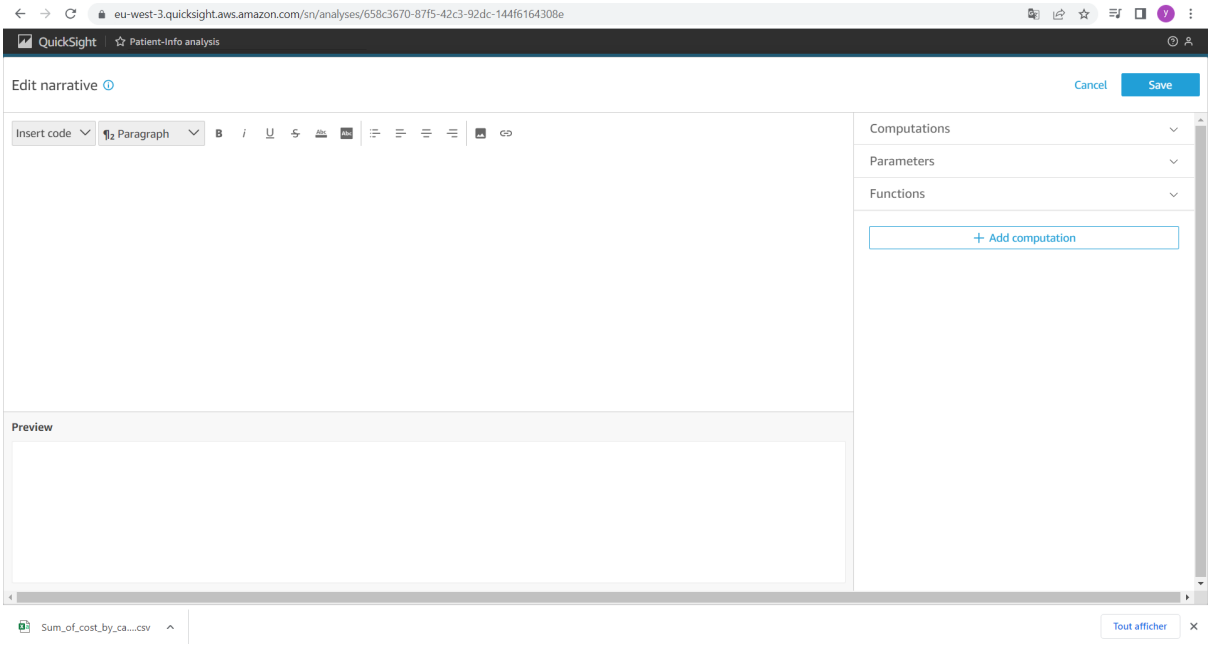
There we have to use some filter to specify the range that we want to visualize

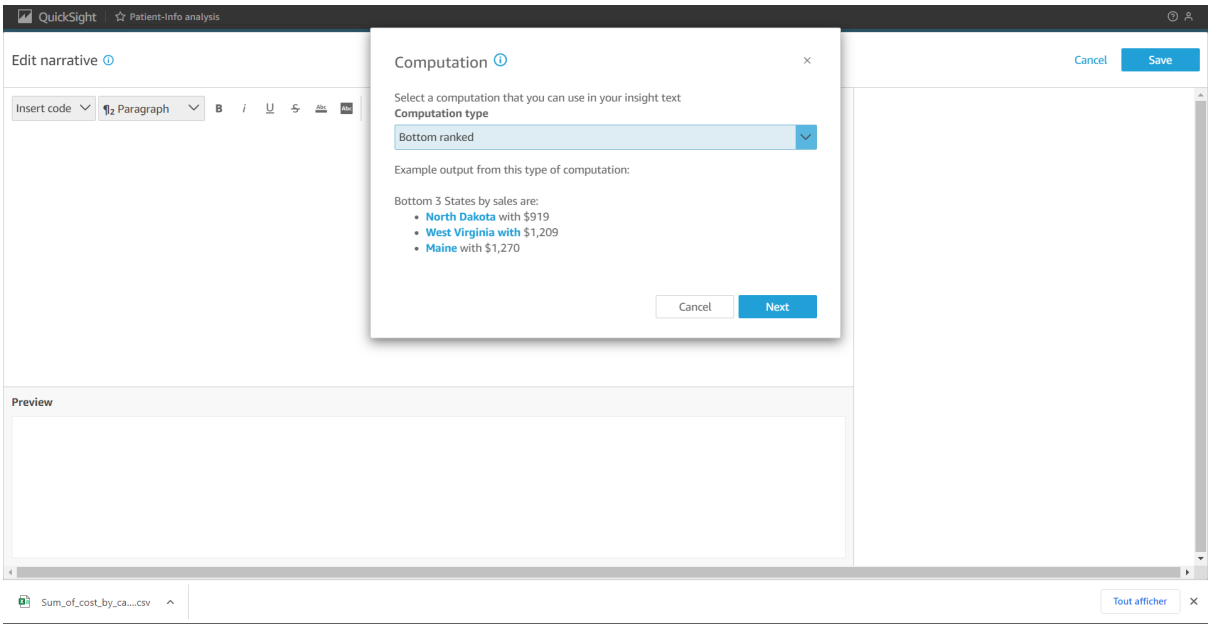


There the things are the same that in the graph one , we just used other types of graphs and changed titles for better visibility

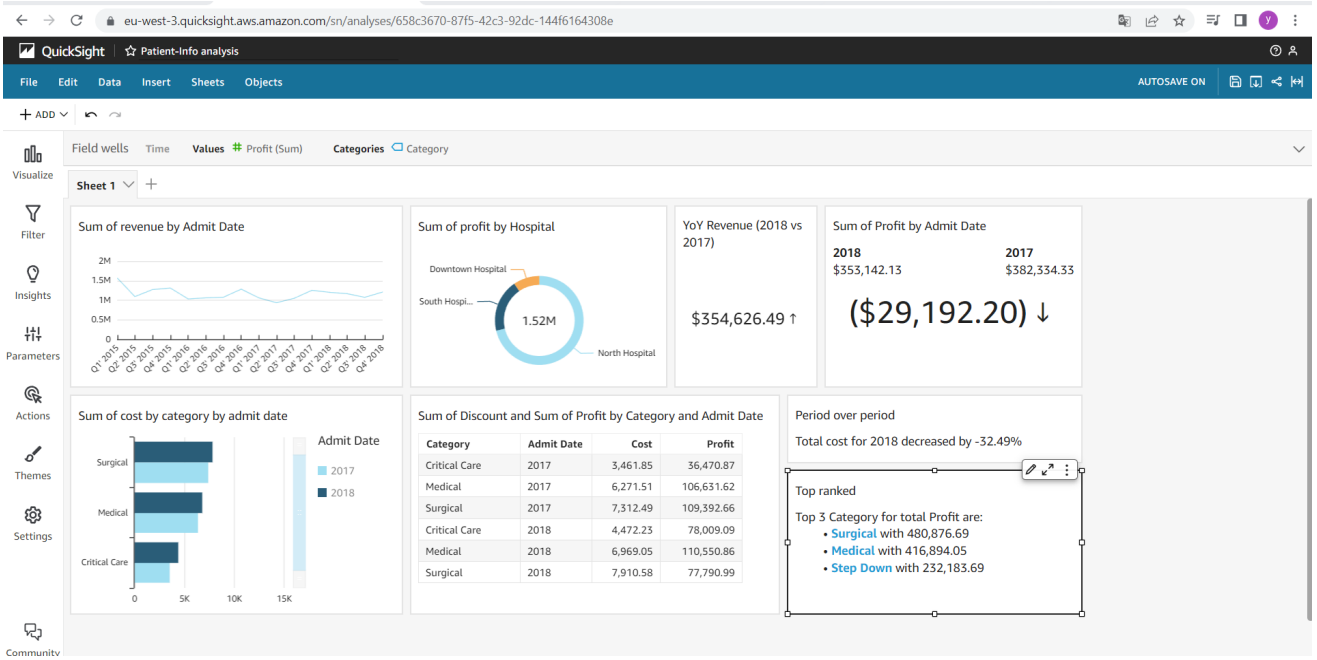


There we have to do some computations to retrieve insights did by quicksight and visualize them in the dashboard





So at the final we just have to refine titles , we also added comparisons over time using KPI visualization. And at the final we get THIS 🙂


eu-west-3.quicksight.aws.amazon.com/sn/analyses/658c3670-87f5-42c3-92dc-144f6164308e
QuickSight
Patient-Info analysis
File
Edit
Data
Insert
Sheets
Objects
AUTOSAVE ON
ADD
Visualize
Filter
Insights
Parameters
Actions
Themes
Settings
Community
Field wells
Time
Values
Profit (Sum)
Categories
Category
Sheet 1
Sum of revenue by Admit Date
2M
1.5M
1M
0.5M
0
Sum of profit by Hospital
Downtown Hospital
South Hospi...
1.52M
North Hospital
YoY Revenue (2018 vs 2017)
$354,626.49 ↑
Sum of Profit by Admit Date
2018
$353,142.13
2017
$382,334.33
($29,192.20) ↓
Sum of cost by category by admit date
Surgical
Medical
Critical Care
0
5K
10K
15K
Admit Date
2017
2018
Sum of Discount and Sum of Profit by Category and Admit Date
Category
Admit Date
Cost
Profit
Critical Care
2017
3,461.85
36,470.87
Medical
2017
6,271.51
106,631.62
Surgical
2017
7,312.49
109,392.66
Critical Care
2018
4,472.23
78,009.09
Medical
2018
6,969.05
110,550.86
Surgical
2018
7,910.58
77,790.99
Period over period
Total cost for 2018 decreased by -32.49%
Top ranked
Top 3 Category for total Profit are:
Surgical with 480,876.69
Medical with 416,894.05
Step Down with 232,183.69