

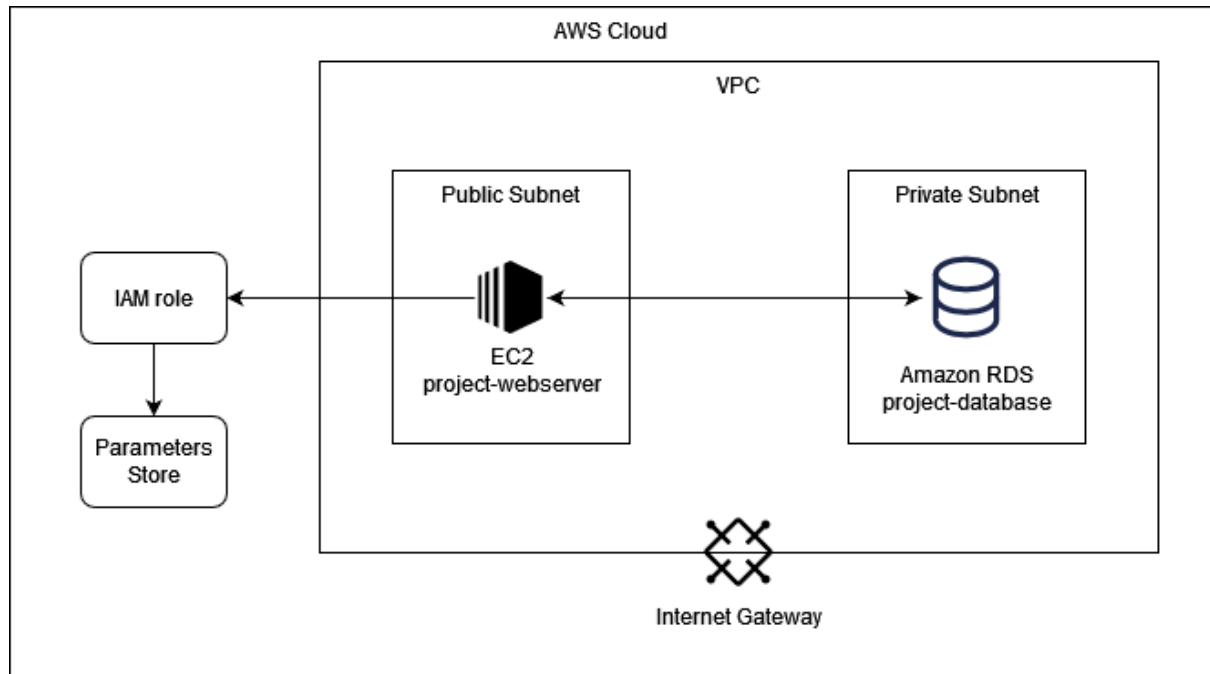
Capstone Project

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.

Vous avez créé avec succès vpc-012cb74a145a3fa09 / project-vpc

VPC > Vos VPC > vpc-012cb74a145a3fa09

vpc-012cb74a145a3fa09 / project-vpc

Actions ▼

Détails Infos

ID de VPC vpc-012cb74a145a3fa09	État Available	Noms d'hôte DNS Désactivé	Résolution DNS Activé
Location Default	Jeu d'options DHCP dopt-08c9a0fe1dfd4ad1c	Table de routage principale rtb-095b19f6ef6d0f200	ACL réseau principal acl-08482f3e7564fa0dd
VPC par défaut Non	CIDR IPv4 10.0.0.0/16	Groupe IPv6 -	CIDR IPv6 -
Métriques d'utilisation d'adresses réseau Désactivé	Groupes de règles du pare-feu DNS de Route 53 Resolver -	ID du propriétaire 634297156469	

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).

Sous-réseaux (7) Infos

🔍

Filter les sous-réseaux

↻

Actions ▼

Créer un sous-réseau

< 1 >

⚙️

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.

Internet gateway igw-075fd268868d42221 successfully attached to vpc-012cb74a145a3fa09

VPC > Internet gateways > igw-075fd268868d42221

igw-075fd268868d42221 / project-internet-gateway

Actions

Details Info

Internet gateway ID

igw-075fd268868d42221

State

Attached

VPC ID

vpc-012cb74a145a3fa09 | project-vpc

Owner

634297156469

Route tables (4) Info

Actions

Create route table

Find resources by attribute or tag

< 1 >

	Name	Route table ID	Explicit subnet associati...	Edge associatio
<input type="checkbox"/>	-	rtb-02988909b6fd1bb7c	-	-
<input type="checkbox"/>	-	rtb-095b19f6ef6d0f200	-	-
<input type="checkbox"/>	public-route-table	rtb-0ba4667459be7c00d	2 subnets	-
<input type="checkbox"/>	private-route-table	rtb-06bd399a45460af8a	2 subnets	-

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.

Instances | EC2 Management Console

https://eu-west-3.console.aws.amazon.com/ec2/home?region=eu-west-3#instances:

Search [Alt+S]

New EC2 Experience

EC2 Dashboard
EC2 Global View
Events

▼ Instances

Instances

Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations

▼ Images

AMIs
AMI Catalog

▼ Elastic Block Store

Volumes
Snapshots

Instances (1/1) info

Find instance by attribute or tag (case-sensitive)

Connect Instance state Actions Launch instances

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
project-web-server	i-0f3122420fe76236b	Running	t2.micro	-	No alarms	eu-west-3c	ec2-13-38-

Instance: i-0f3122420fe76236b (project-web-server)

Details Security Networking Storage Status checks Monitoring Tags

▼ Instance summary info

Instance ID
i-0f3122420fe76236b (project-web-server)

Public IPv4 address
13.38.8.34 | open address

Private IPv4 addresses
172.31.37.11

Instance state
Running

Public IPv4 DNS
ec2-13-38-8-34.eu-west-3.compute.amazonaws.com | open address

IPv6 address
-

Instance type
t2.micro

Hostname type
IP name: ip-172-31-37-11.eu-west-3.compute.internal

Private IP DNS name (IPv4 only)
ip-172-31-37-11.eu-west-3.compute.internal

Answer private resource DNS name
IPv4 (A)

Elastic IP addresses
-

CloudShell Feedback Language

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

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

EC2 Management Console

https://eu-west-3.console.aws.amazon.com/ec2/home?region=eu-west-3#securitygroups:

Search [Alt+S]

New EC2 Experience

EC2 Dashboard
EC2 Global View
Events

▼ Instances

Instances

Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations

▼ Images

AMIs
AMI Catalog

▼ Elastic Block Store

Volumes
Snapshots

Inbound security group rules successfully modified on security group (sg-0fb9fe8e7cd016373 | project-security-group)

Details

Security Groups (1/2) info

Filter security groups

Actions Export security groups to CSV Create security group

Name	Security group ID	Security group name	VPC ID	Description	Owner
-	sg-0fb9fe8e7cd016373	project-security-group	vpc-0effbde29839b34bc	project-security-group ...	634297156469

You can now check network connectivity with Reachability Analyzer

Run Reachability Analyzer

Inbound rules (3)

Filter security group rules

Name	Security group rule...	IP version	Type	Protocol	Port range
-	sgr-0849b2fa7dcd725f	IPv4	HTTPS	TCP	443
-	sgr-04be95197a9b6ac9c	IPv4	SSH	TCP	22
-	sgr-04f3023a6bf23f290	IPv4	HTTP	TCP	80

CloudShell Feedback Language

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

```
ubuntu@ip-172-31-37-11:~$ ssh -i "efrei-cloud-key.pem" ubuntu@ec2-13-38-8-34.eu-west-3.compute.amazonaws.com
PS C:\Users\Ayyoub\Downloads> ssh -i "efrei-cloud-key.pem" ubuntu@ec2-13-38-8-34.eu-west-3.compute.amazonaws.com
The authenticity of host 'ec2-13-38-8-34.eu-west-3.compute.amazonaws.com (13.38.8.34)' can't be established.
ED25519 key fingerprint is SHA256:zi22dGv93v24k5r/mArqsoSyzs0J+EUX9ErV43okYAK.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-13-38-8-34.eu-west-3.compute.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.19.0-1025-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Fri Jul 21 17:03:16 UTC 2023

System load:  0.02/783203125      Processes:    100
Usage of /:   5.4% of 28.89GB     Users logged in: 0
Memory usage: 24%                IPv4 address for eth0: 172.31.37.11
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-37-11:~$ |
```

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):

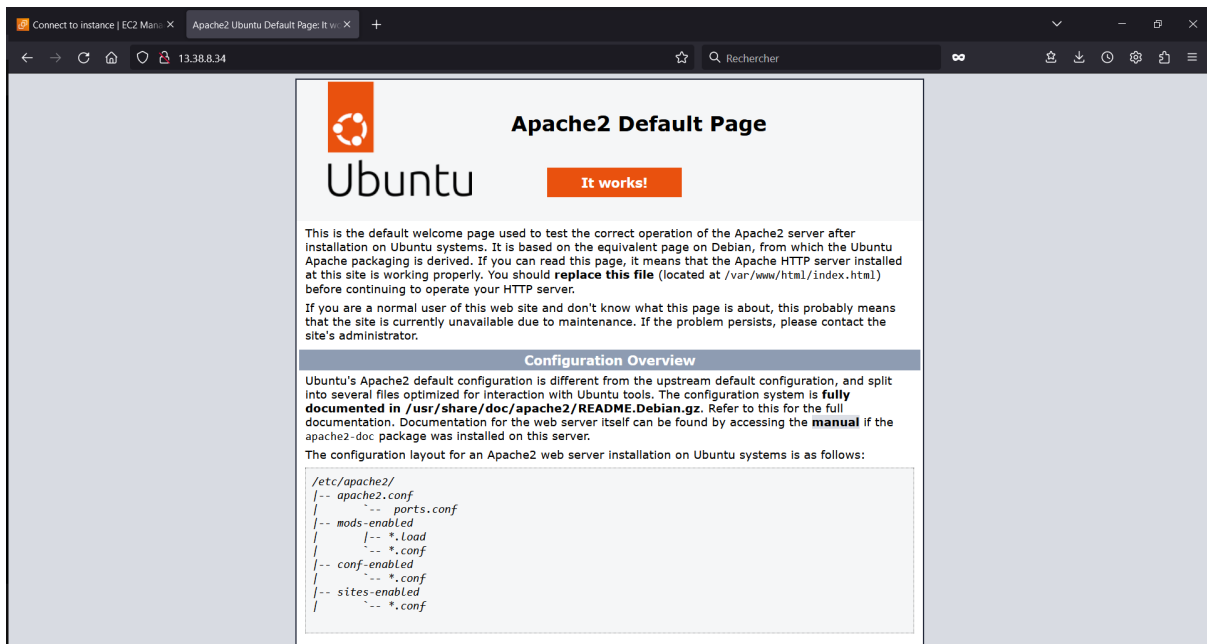
```
sudo apt-get update -y
sudo apt-get upgrade -y
sudo apt-get install apache2 -y
sudo apt-get install php -y
sudo apt-get install mariadb-server -y
```

Example: checking MariaDB installation

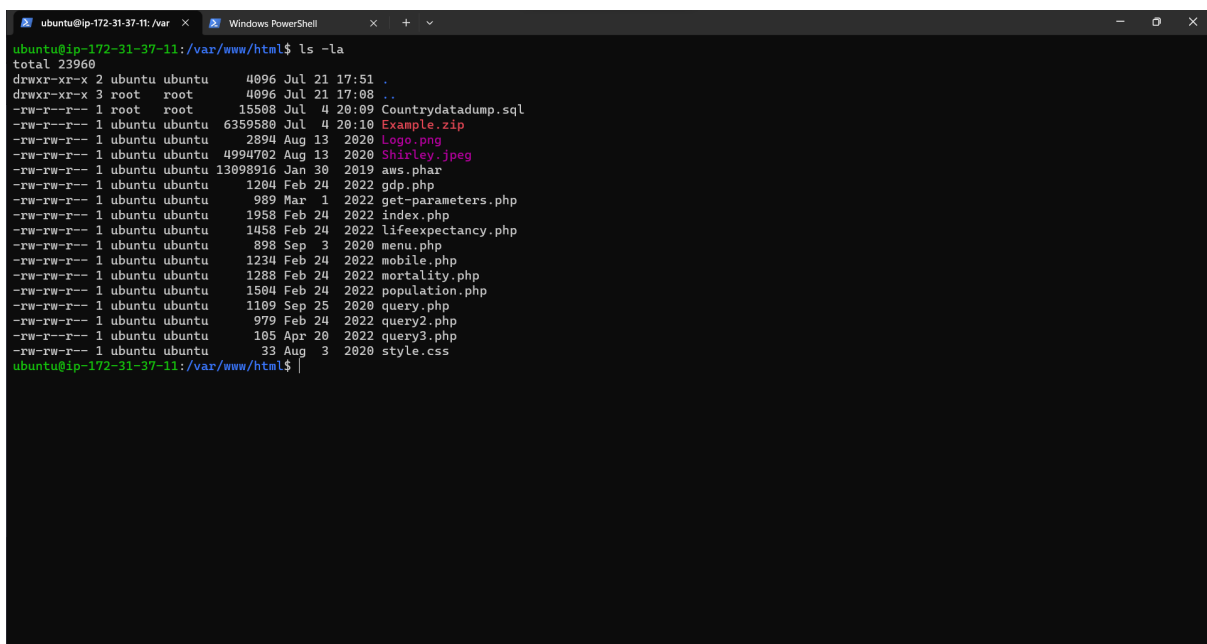
```
ubuntu@ip-172-31-37-11:~$ /usr/bin/mariadb -V
/usr/bin/mariadb from 11.0.2-MariaDB, client 15.2 for debian-linux-gnu (x86_64) using EditLine wrapper
ubuntu@ip-172-31-37-11:~$ sudo systemctl status mariadb
● mariadb.service - MariaDB 11.0.2 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset: enabled)
   Drop-In: /etc/systemd/system/mariadb.service.d
            └─migrated-from-my.cnf-settings.conf
   Active: active (running) since Fri 2023-07-21 17:20:20 UTC; 1min 1s ago
     Docs: man:mariadb(8)
           https://mariadb.com/kb/en/library/systemd/
   Process: 24307 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /var/run/mysqld (code=exited, status=0/SUCCESS)
   Process: 24308 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_START_POSITION (code=exited, status=0/SUCCESS)
   Process: 24310 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] && VAR= [[ VAR=cd /usr/bin/..; /usr/bin/galera_recovery' ; [ $? -eq 0 ] &&
   Process: 24353 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP_START_POSITION (code=exited, status=0/SUCCESS)
   Process: 24355 ExecStartPost=/etc/mysql/debian-start (code=exited, status=0/SUCCESS)
  Main PID: 24339 (mariabdd)
    Status: "Taking your SQL requests now..."
     Tasks: 10 (limit: 1141)
  Memory: 78.9M
     CPU: 413ms
    CGroup: /system.slice/mariadb.service
            └─24339 /usr/sbin/mariabdd

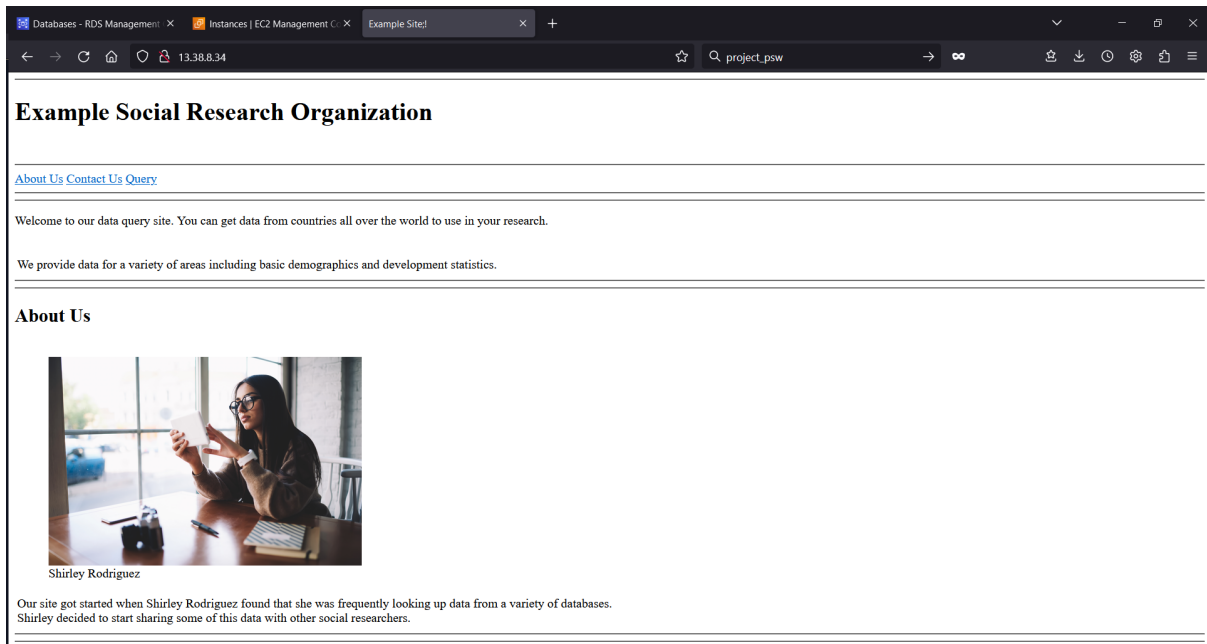
Jul 21 17:20:19 ip-172-31-37-11 mariabdd[24339]: 2023-07-21 17:20:19 0 [Note] Plugin 'wsrep-provider' is disabled.
Jul 21 17:20:19 ip-172-31-37-11 mariabdd[24339]: 2023-07-21 17:20:19 0 [Note] Server socket created on IP: '127.0.0.1'.
Jul 21 17:20:19 ip-172-31-37-11 mariabdd[24339]: 2023-07-21 17:20:19 0 [Note] InnoDB: Loading buffer pool(s) from /var/lib/mysql/ib_buffer_pool
Jul 21 17:20:20 ip-172-31-37-11 mariabdd[24339]: 2023-07-21 17:20:20 0 [Note] InnoDB: Buffer pool(s) load completed at 230721 17:20:20
Jul 21 17:20:20 ip-172-31-37-11 mariabdd[24339]: 2023-07-21 17:20:20 0 [Note] /usr/sbin/mariabdd: ready for connections.
Jul 21 17:20:20 ip-172-31-37-11 mariabdd[24339]: Version: '11.0.2-MariaDB-1:11.0.2+maria-ubu2204' socket: '/run/mysqld/mysqld.sock' port: 3306 mariadb.o
Jul 21 17:20:20 ip-172-31-37-11 systemd[1]: Started MariaDB 11.0.2 database server.
Jul 21 17:20:20 ip-172-31-37-11 /etc/mysql/debian-start[24357]: Upgrading MySQL tables if necessary.
Jul 21 17:20:20 ip-172-31-37-11 /etc/mysql/debian-start[24368]: Checking for insecure root accounts.
Jul 21 17:20:20 ip-172-31-37-11 /etc/mysql/debian-start[24372]: Triggering myisam-recover for all MyISAM tables and aria-recover for all Aria tables
lines 1-30/30 (END)
```

Apache default home page, our server is working!



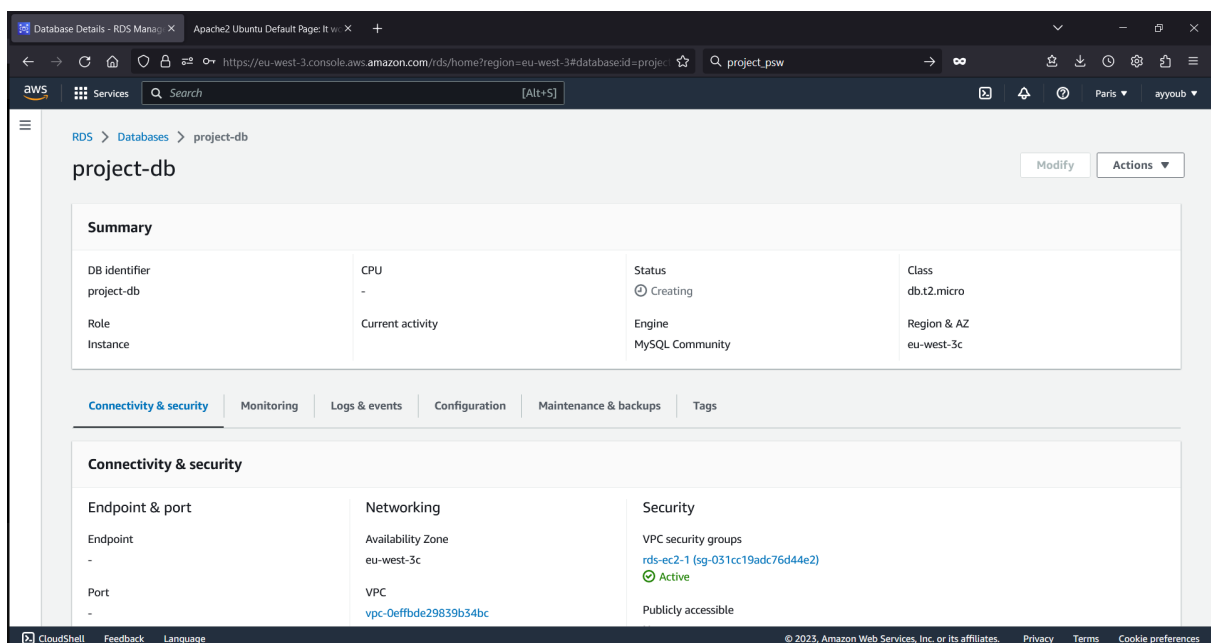
Next, we upload our files to the server using the `wget` command, then extract all the files into the `/var/www/html/` folder, taking care to delete the `index.html` file created by default when the Apache server was installed.





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 ~$ sudo mysql -u admin -p --host project-db.csrubyhxmkm.eu-west-3-rds.amazonaws.com
mysql: Deprecating 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.

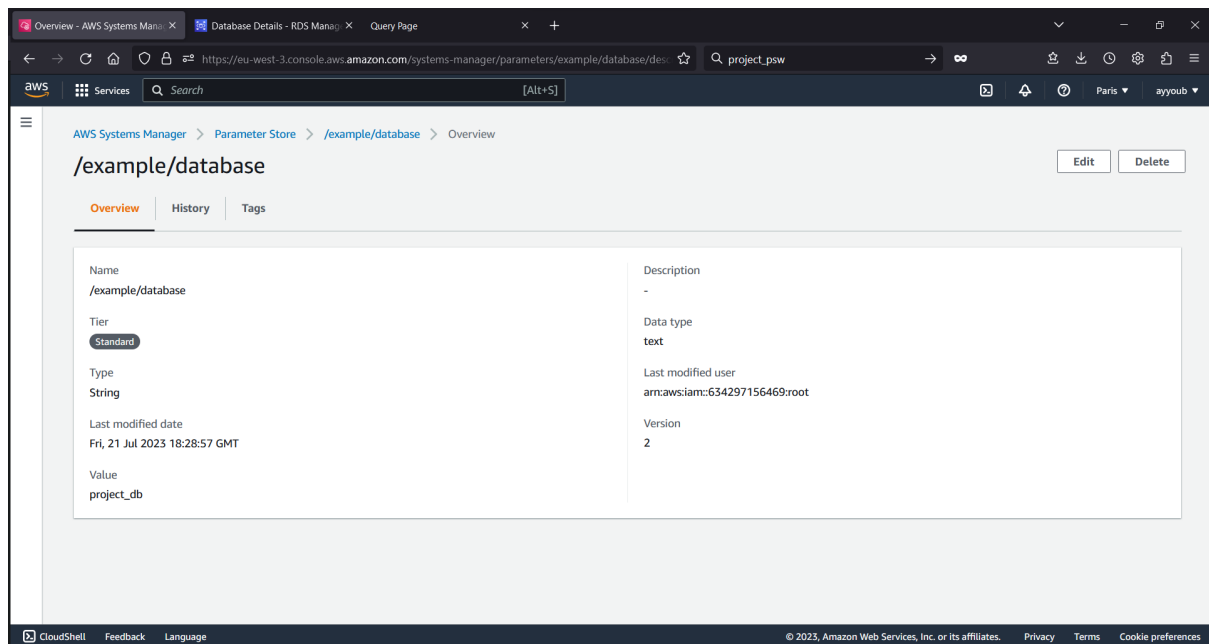
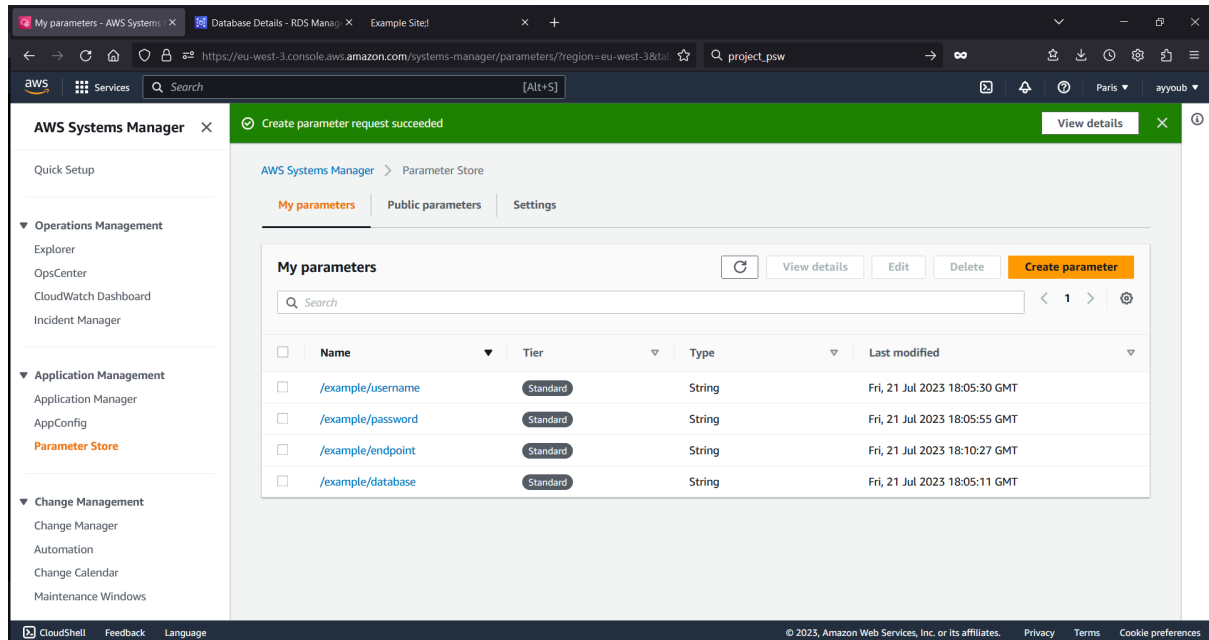
MySQL [(none)]> USE project_db;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MySQL [project_db]> SELECT * from countrydata_final;
```

name	mobilephones	mortalityunder5	healthexpenditurepercapita	healthexpenditurepercentGDP	population	populationurban	birthrate	lifeexpectancy	GDP
Afghanistan	0	150	11	9	26697430	5771984	50	46	2401666315
Albania	29791	29	75	6	3071856	1280964	17	74	368649387
Algeria	80000	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	20928	21	0	802526701
Argentina	6407920	20	689	9	36930709	3374569	10	74	280204000000
Armenia	17486	33	41	7	3076098	2002500	13	71	1911563665
Aruba	15000	0	0	0	90271	42157	14	74	1858659293
Australia	8560000	6	1728	8	19153000	15701416	13	80	416880000000
Austria	6117000	6	2361	10	8011566	5271610	10	78	192071800000
Azerbaijan	420400	67	30	5	8000000	4120883	15	67	5272617196
Bahamas, The	31524	17	1072	6	297651	244074	18	72	6327552000
Bahrain	205727	12	400	4	630193	560163	22	74	7970000000
Bangladesh	279000	86	9	3	129592275	30583777	27	65	47124925462
Barbados	20407	17	640	6	267511	97100	12	76	2558850040
Belarus	49353	14	66	6	10005000	6993495	9	69	12736856405
Belgium	5620000	6	2025	9	10251250	9953960	11	78	232073000000
Belize	16012	27	123	4	200000	119400	30	74	632072465
Benin	55476	143	15	4	6517010	2406321	43	53	2254830605
Bermuda	13000	0	0	0	62100	62100	14	78	3400210000
Bhutan	0	0	53	7	571262	345101	27	62	427000017
Bolivia	582620	82	61	6	8307200	5133079	31	63	8397858200
Bosnia and Herzegovina	93306	10	100	7	3693698	1595670	11	75	5505900405
Botswana	222100	96	155	5	1757925	935216	27	51	5632391130
Brazil	23100171	36	260	7	170425307	141633010	21	70	644702000000
Brunei Darussalam	95000	9	541	3	327036	232523	23	77	6001153318
Bulgaria	730000	21	95	6	8170172	5629249	9	72	12903546570
Burkina Faso	25245	191	11	5	12294012	2040800	45	50	2610045549
Burundi	10320	164	7	7	6374347	529071	30	46	836334007

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 Manager

Database Details - RDS Manager

Example Site!

← → ↺ ↻ 🔒

https://eu-west-3.console.aws.amazon.com/systems-manager/parameters/example/endpoint/overview

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

Edit

Delete

Overview

History

Tags

Name

/example/endpoint

Tier

Standard

Type

String

Last modified date

Fri, 21 Jul 2023 18:10:27 GMT

Value

project-db.csrxythxmmk.eu-west-3.rds.amazonaws.com

Description

-

Data type

text

Last modified user

arn:aws:iam::634297156469:root

Version

1

CloudShell

Feedback

Language

© 2023, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

Overview - AWS Systems Manager

Database Details - RDS Manager

Example Site!

← → ↺ ↻ 🔒

https://eu-west-3.console.aws.amazon.com/systems-manager/parameters/example/username/overview

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

Edit

Delete

Overview

History

Tags

Name

/example/username

Tier

Standard

Type

String

Last modified date

Fri, 21 Jul 2023 18:05:30 GMT

Value

admin

Description

-

Data type

text

Last modified user

arn:aws:iam::634297156469:root

Version

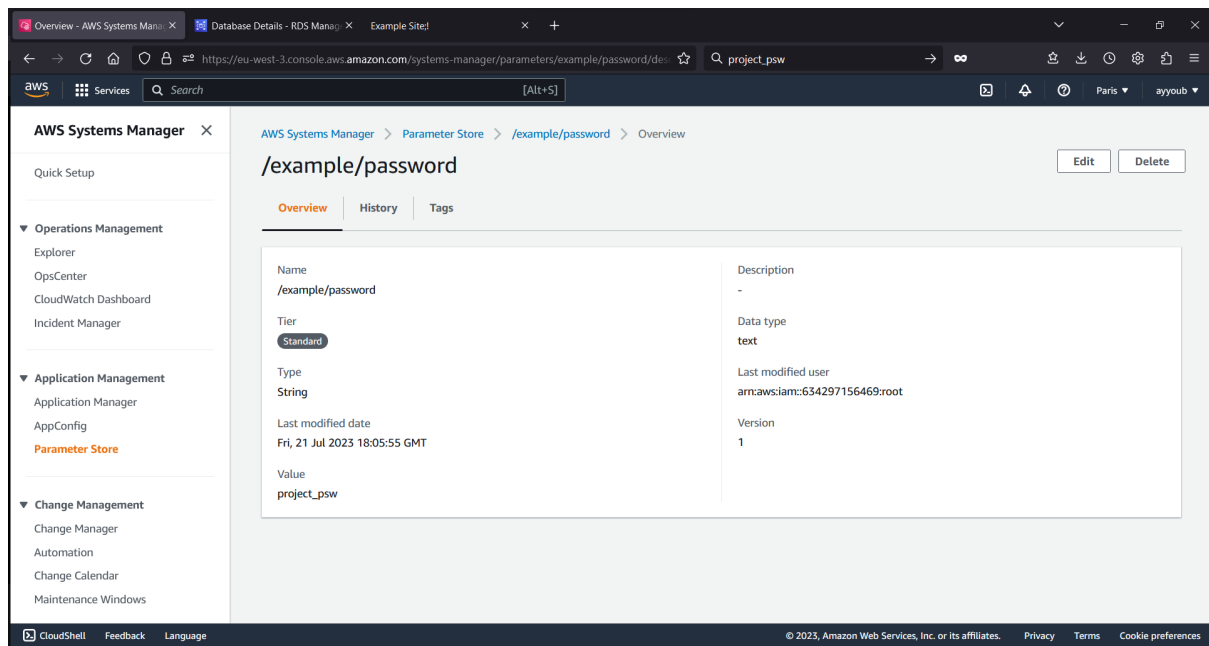
1

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:

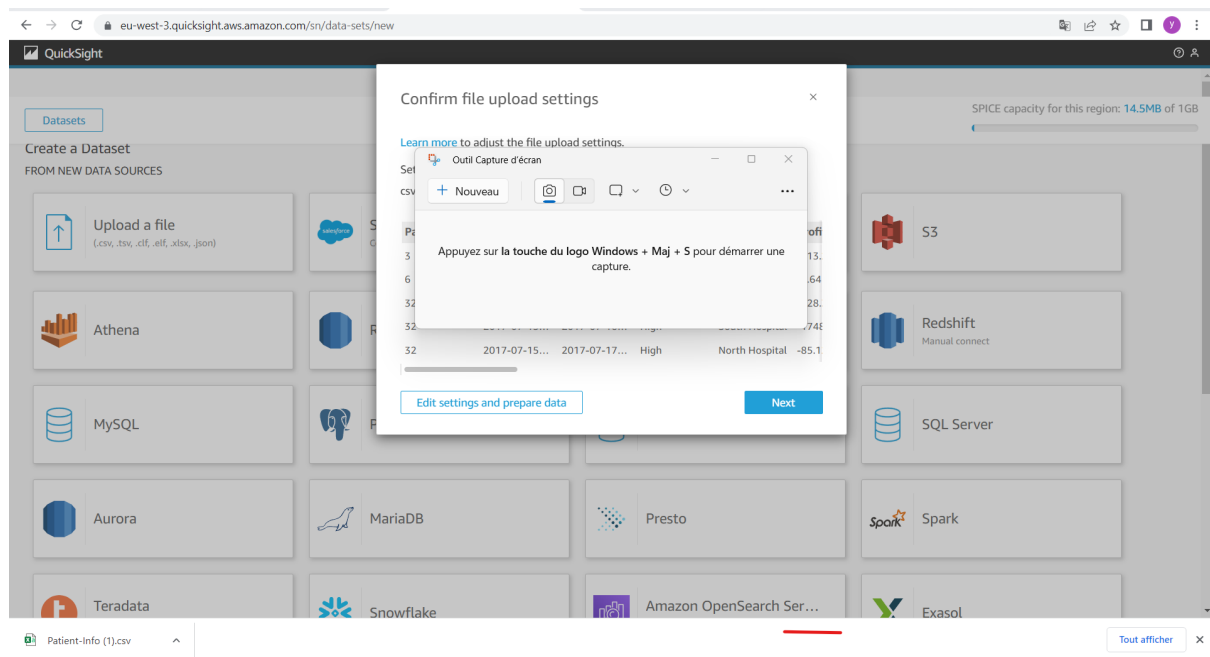
- The policy allows the following actions:
 - `ec2:RunInstances`
 - `ec2:StartInstances`
- The first policy will restrict the access granted by the second policy by denying the `ec2:RunInstances` and `ec2:StartInstances` actions if the instance type is “t2.micro” or “t2.small”. For example, if you tried 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

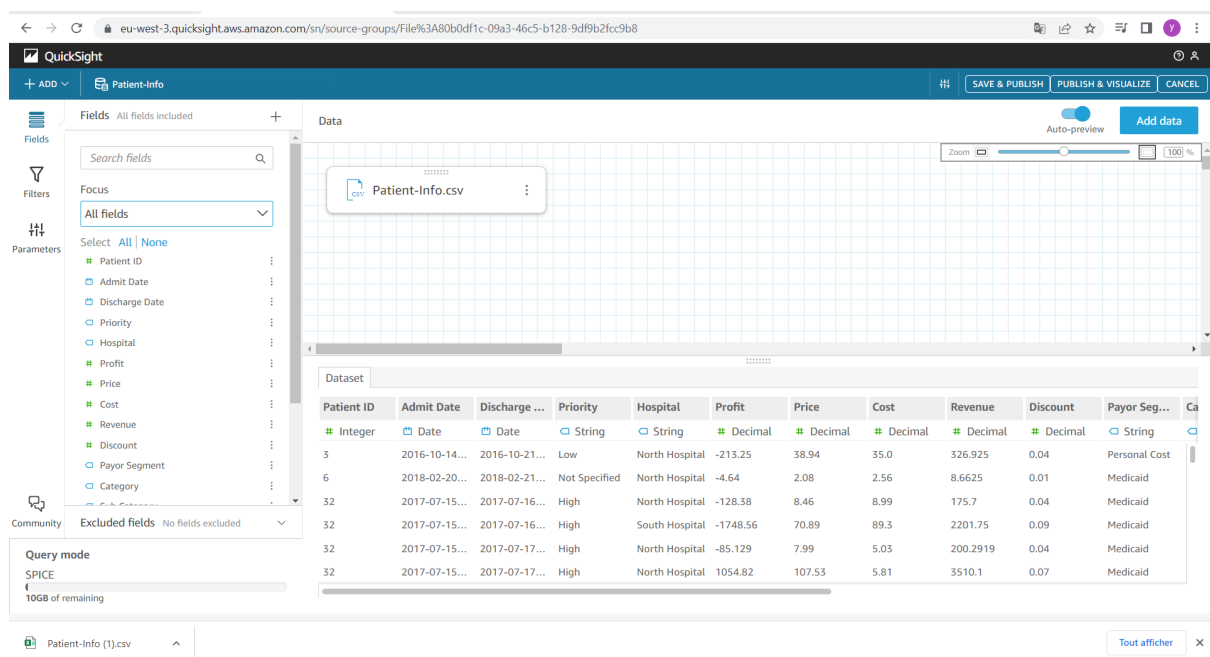
Dashboard Creation:

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

The image displays two screenshots of the AWS QuickSight 'Add calculated field' dialog, illustrating the configuration of calculated fields using the `sumOver` function.

Top Screenshot: Sum of profit by Hospital

The title bar indicates the calculated field is 'Sum of profit by Hospital'. The expression editor shows the following code:

```
1 sumOver
2 (
3   sum(Profit),
4   [ Hospital ]
5 )
```

The right-hand panel shows the 'Functions' list with 'abs' selected. The description for 'abs' is: 'Returns the absolute value of a given expression. SYNTAX: abs(expression)'. A 'Learn more' link is also present.

Bottom Screenshot: Sum of cost by Category by Admit Date

The title bar indicates the calculated field is 'Sum of cost by Category by Admit Date'. The expression editor shows the following code:

```
1 sumOver
2 (
3   sum(Cost),
4   [Category, {Admit Date}]
5 )
```

The right-hand panel shows the 'Functions' list with 'abs' selected. The description for 'abs' is: 'Returns the absolute value of a given expression. SYNTAX: abs(expression)'. A 'Learn more' link is also present.

eu-west-3.quicksight.aws.amazon.com/sv/data-sets/b84f4e09-320a-4d80-bd7a-f3e31848a45f/prepare

Add calculated field

Sum of profit by Category and Admit Date

Cancel Save

```
1 sumOver
2 (
3   | sum(Profit),
4   | [{Category}, {Admit Date}]
5 )
```

Fields

Parameters

Functions

Search functions

abs

addDateTime

addWorkDays

avg

avgIf

avgOver

cell

coalesce

abs

Returns the absolute value of a given expression.

SYNTAX

abs(expression)

[Learn more](#)

eu-west-3.quicksight.aws.amazon.com/sv/data-sets/b84f4e09-320a-4d80-bd7a-f3e31848a45f/prepare

Add calculated field

Sum of discount by Category and admit date

Cancel Save

```
1 sumOver
2 (
3   | sum({Discount}),
4   | [{Category}, {Admit Date}]
5 )
```

Fields

Parameters

Functions

Search functions

abs

addDateTime

addWorkDays

avg

avgIf

avgOver

cell

coalesce

abs

Returns the absolute value of a given expression.

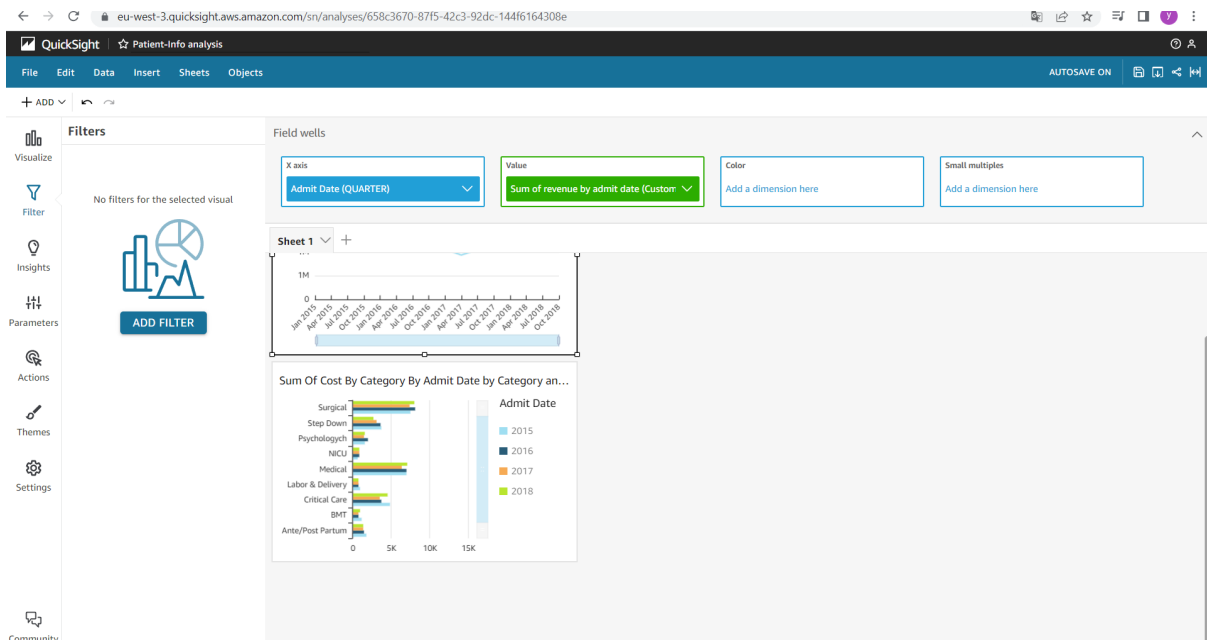
SYNTAX

abs(expression)

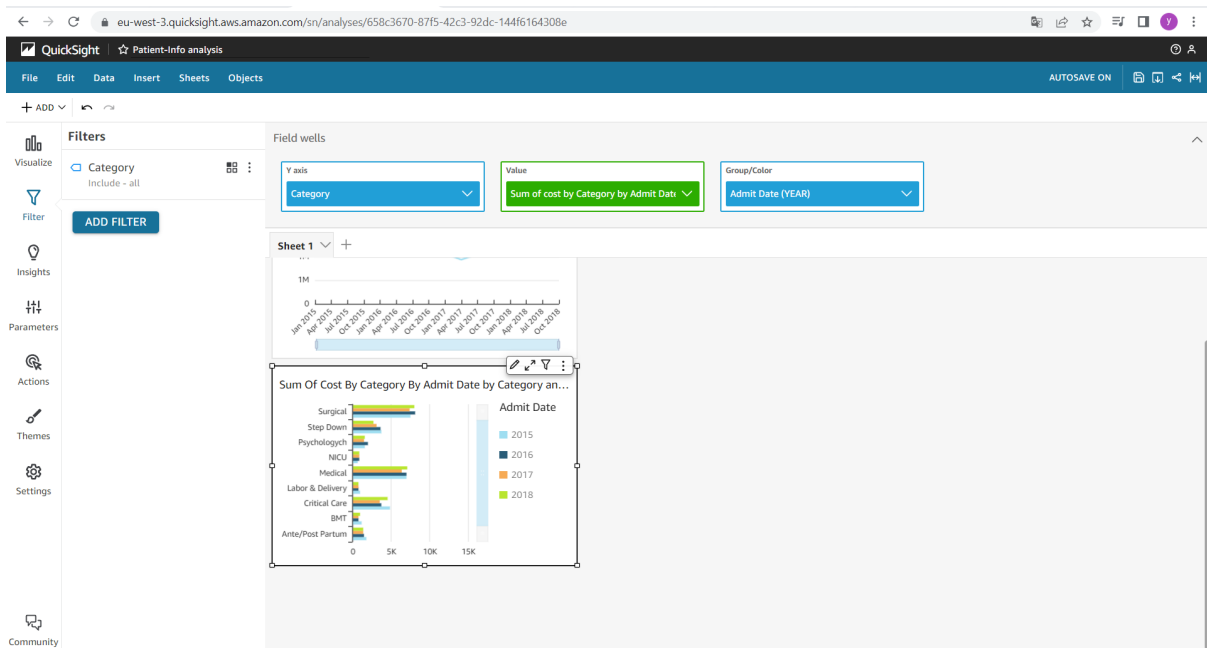
[Learn more](#)

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

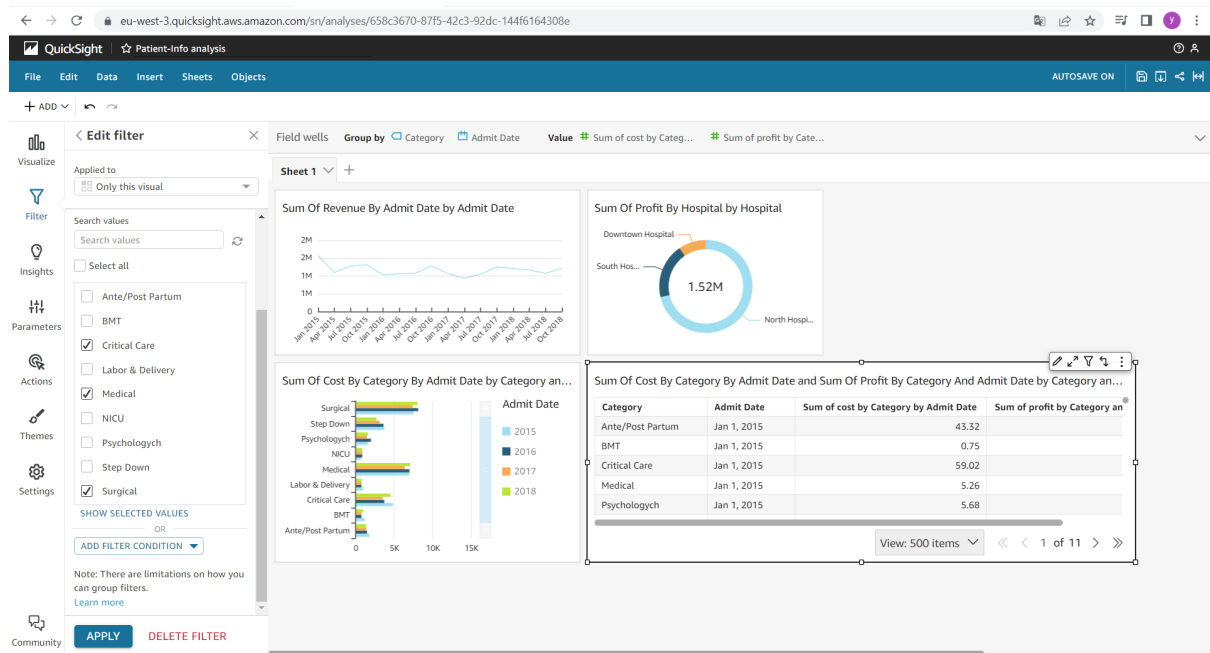
For the first graphic that we can see here, we have to specify the different axis and the scale, the process is the same for all the other graphs .



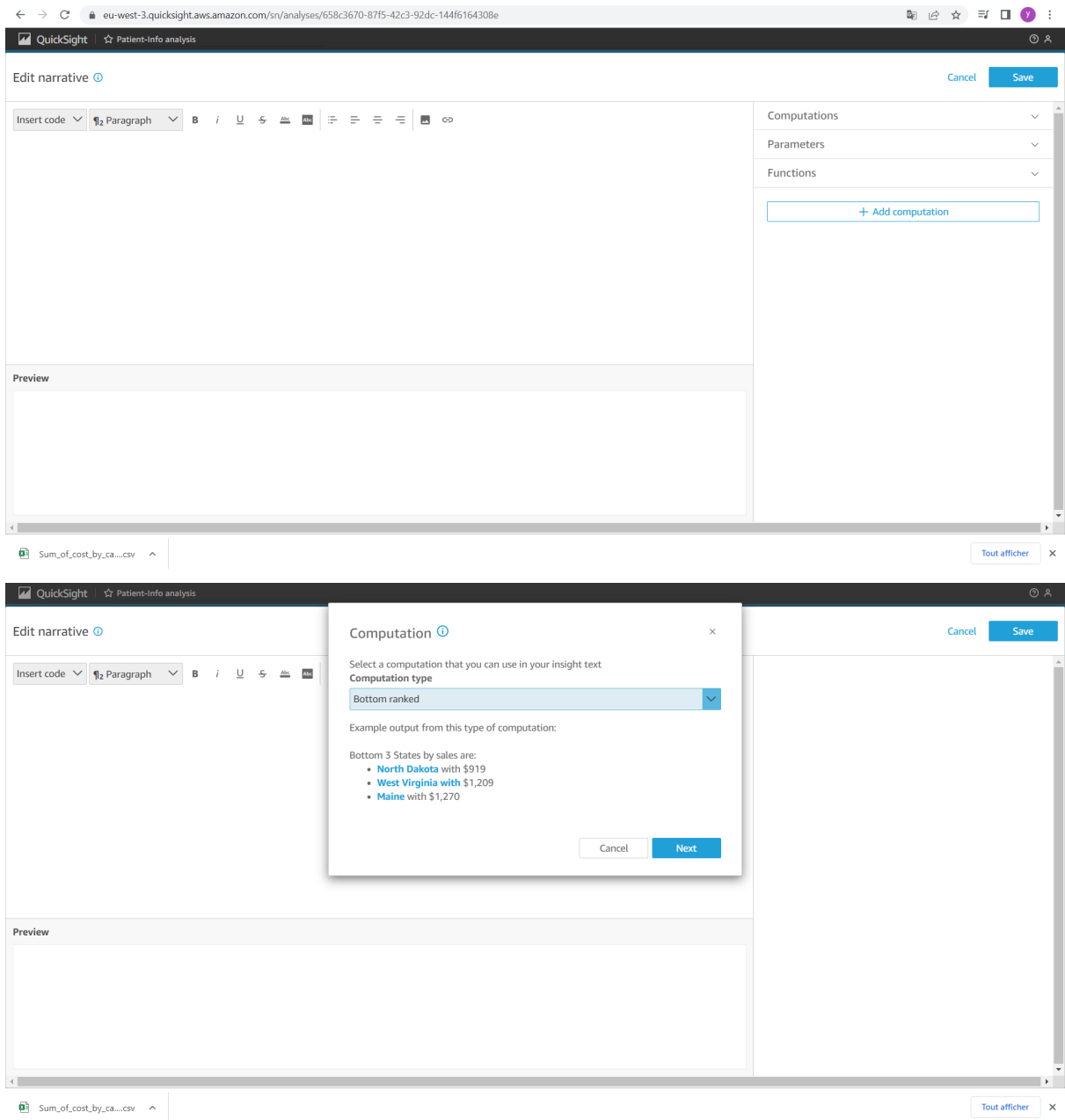
There we have to use some filter to specify the range that we want to visualize



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 😊

