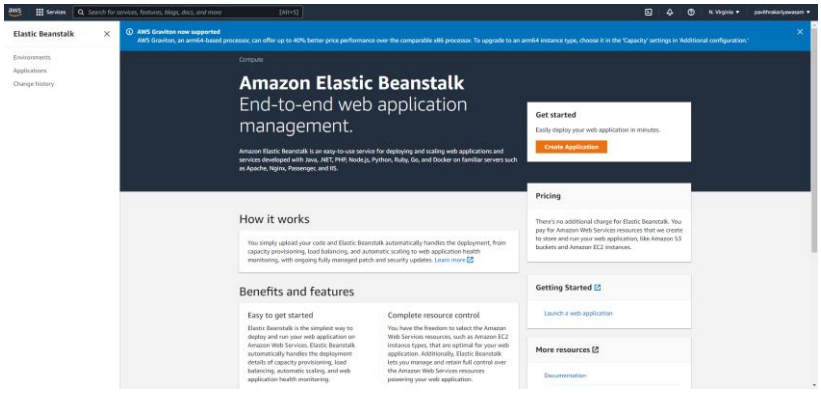
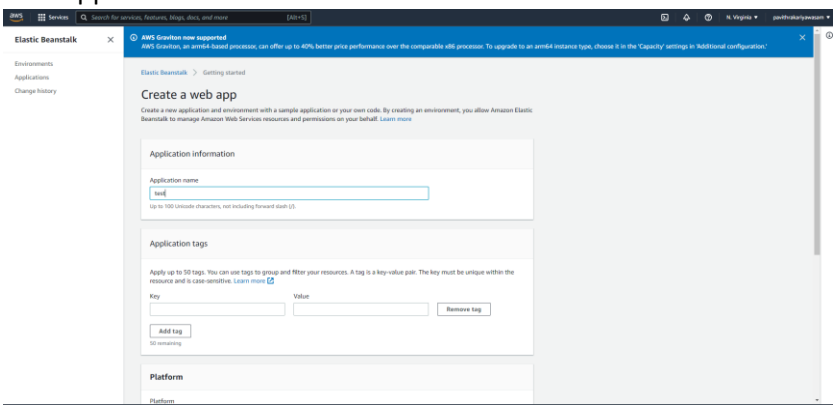
Elastic Beanstalk

https://us-east-1.console.aws.amazon.com/elasticbeanstalk

1. Create application



2. Give application name

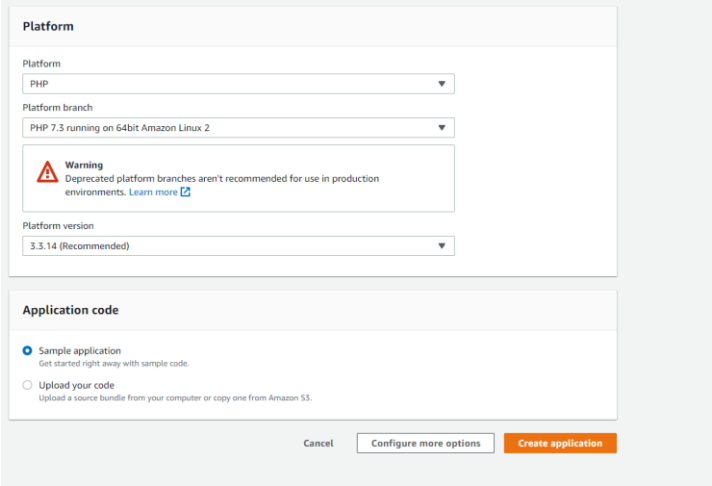


3. Select platform
   Platform – PHP

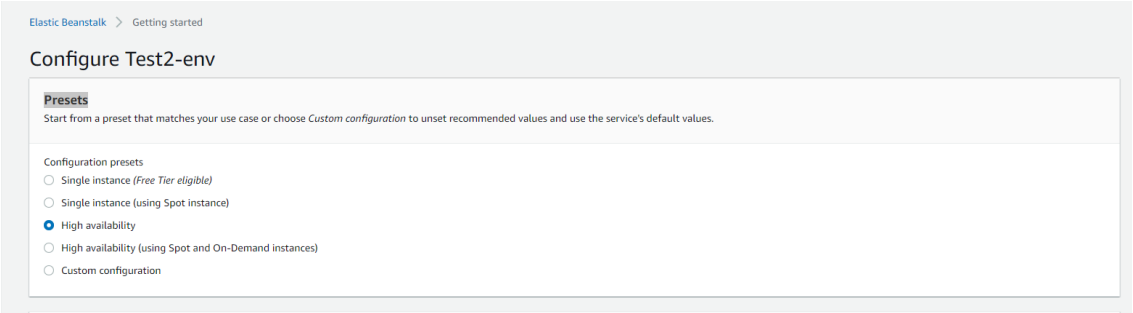
   Platform branch - PHP 7.3 running on 64bit Amazon Linux 2

   Application Code – sample application

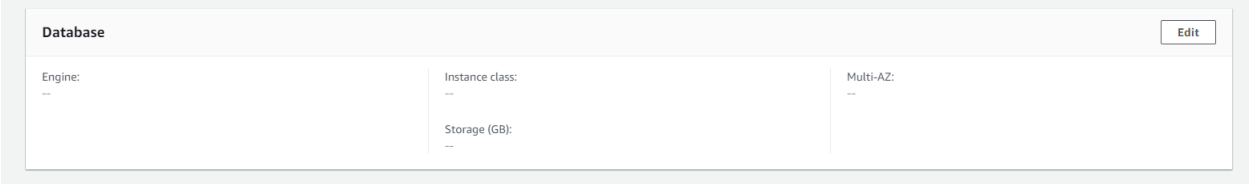
   Click configure more options

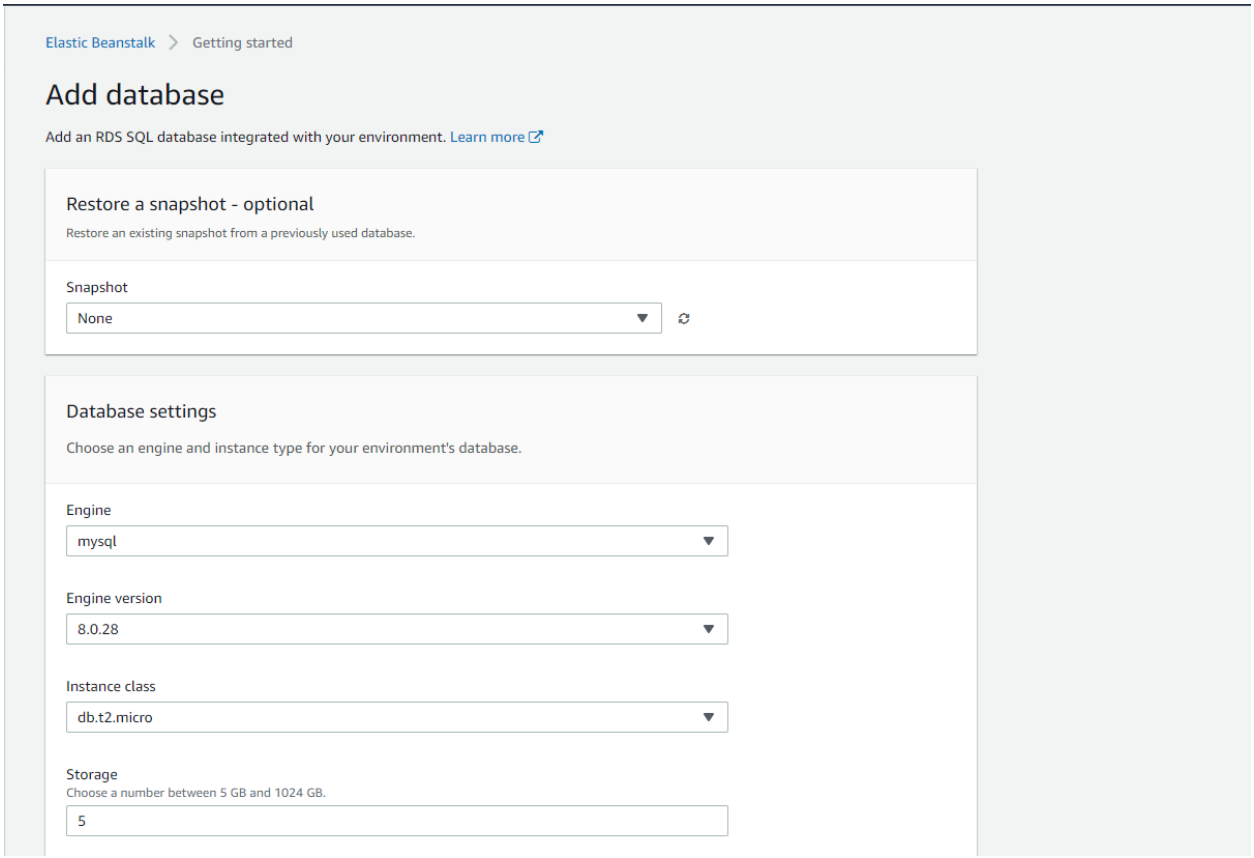


4. Presets

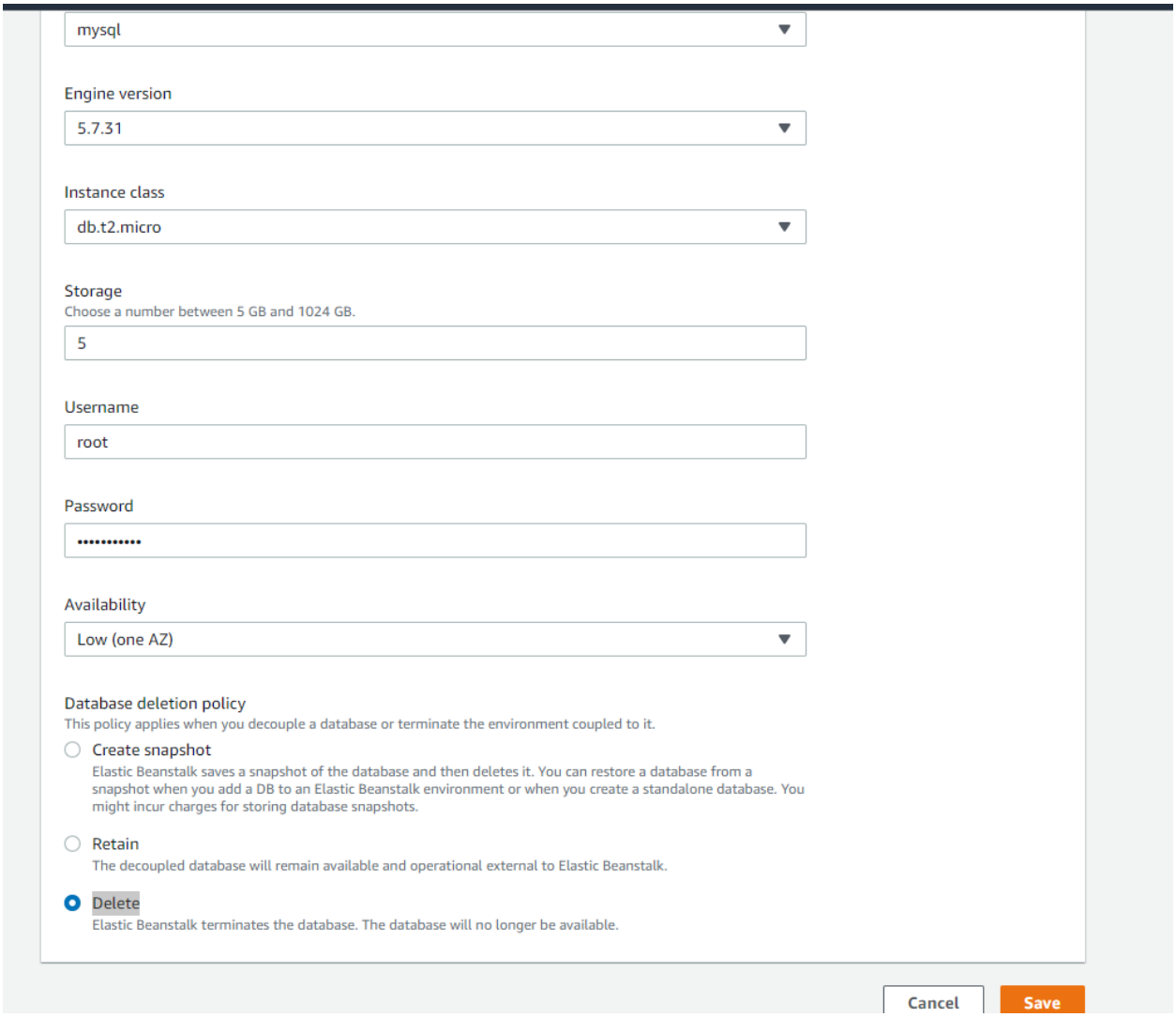


5. Database – edit





Engine – mysql
Engine version - 5.7.31
Give username and password
Database deletion policy – Delete
Keep other configurations as it is



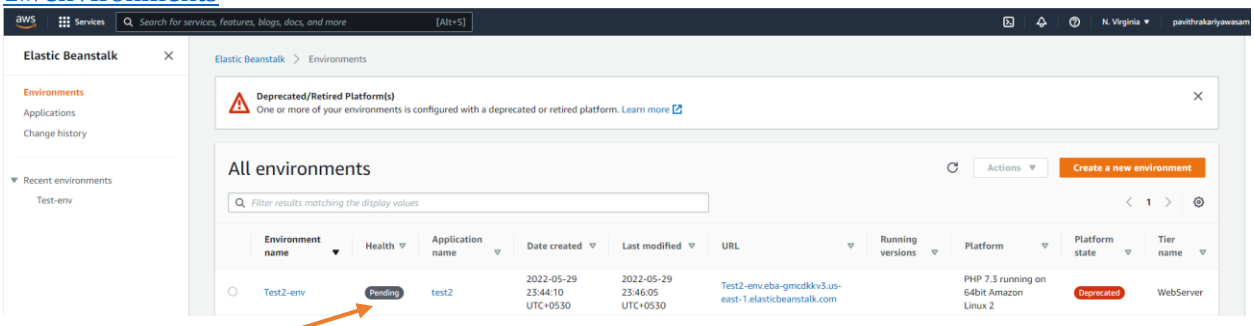
Save

Create app

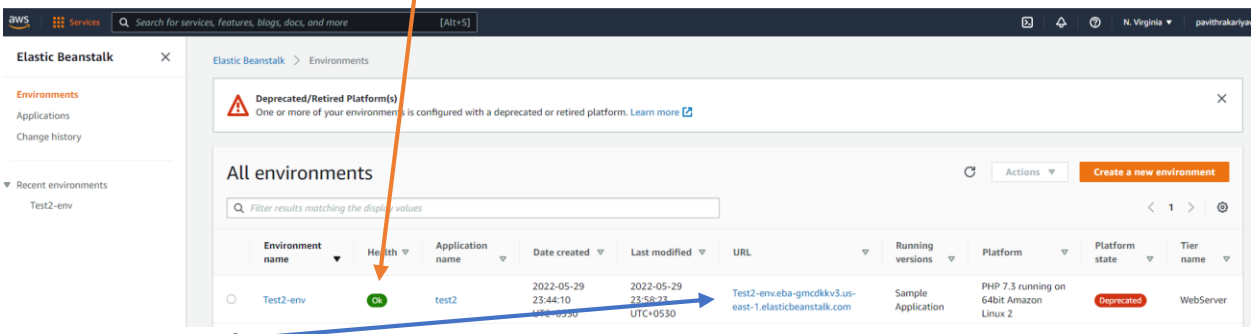
This will take some time to finish

6. Check on beanstallk dashboard

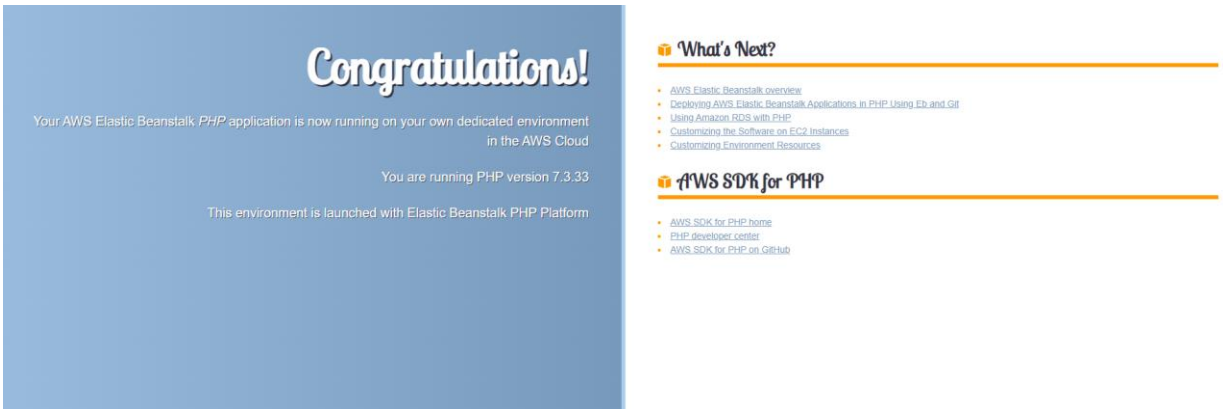
https://us-east-1.console.aws.amazon.com/elasticbeanstalk/home?region=us-east-1#/environments



Once finished (health become ok – green color)

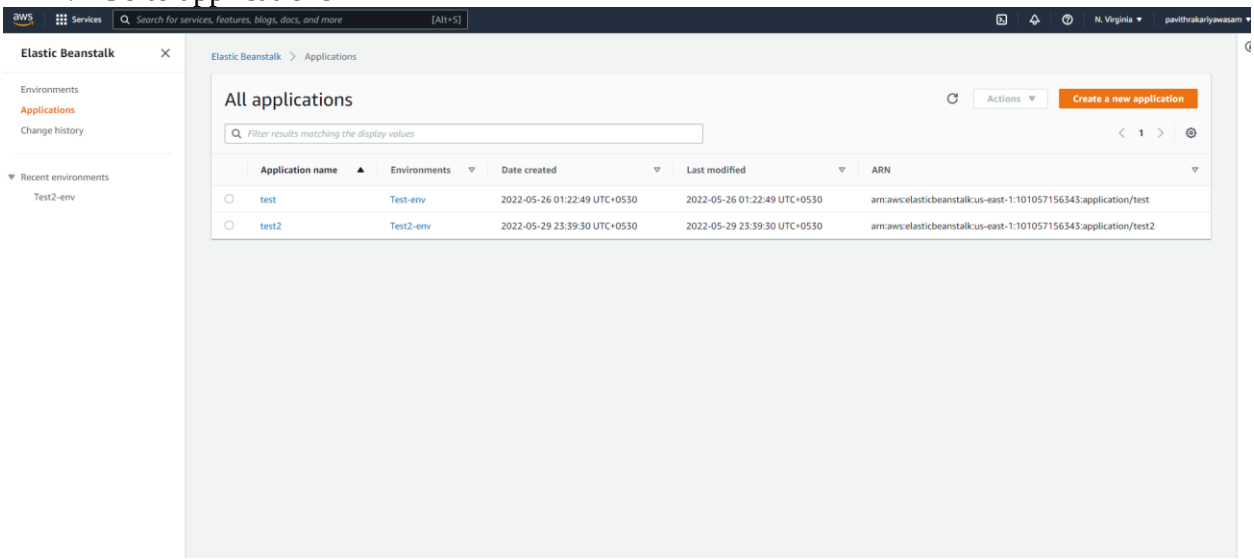


Click on the url of the environment, you should view web application like this



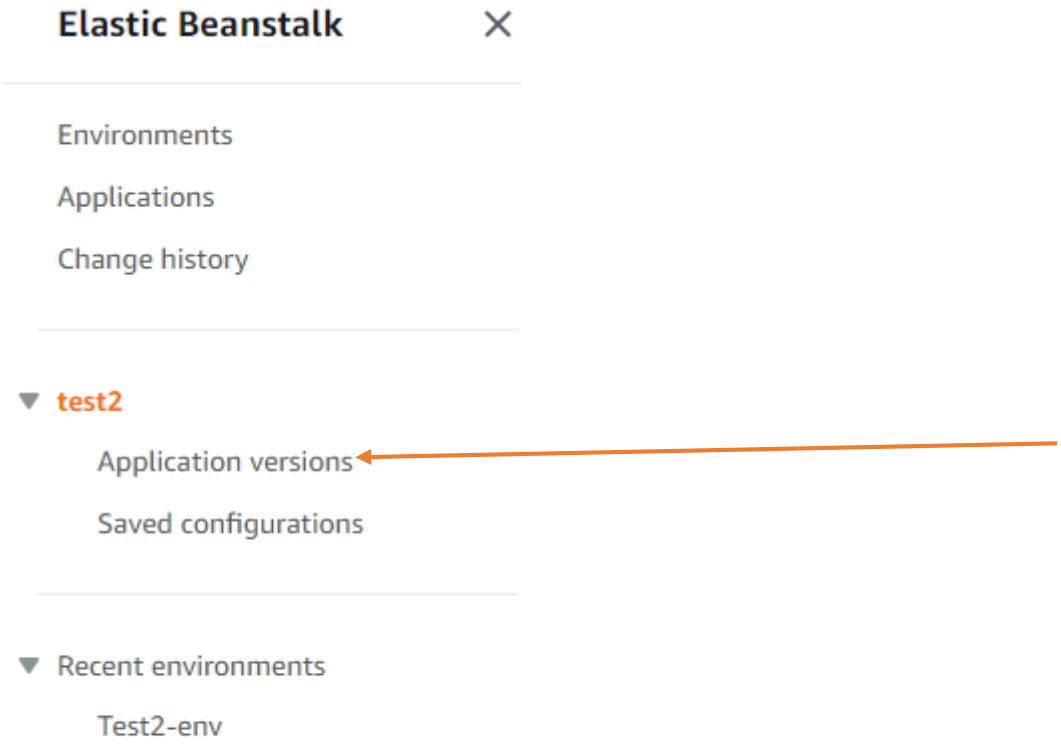
Go back

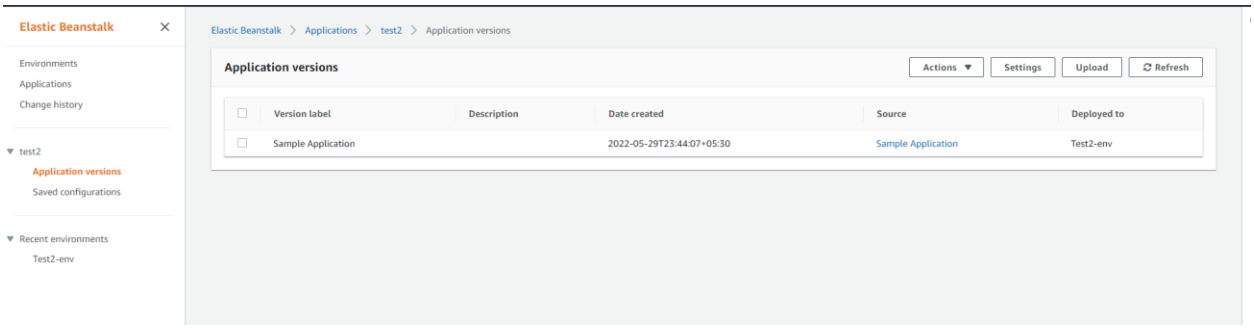
7. Go to applications



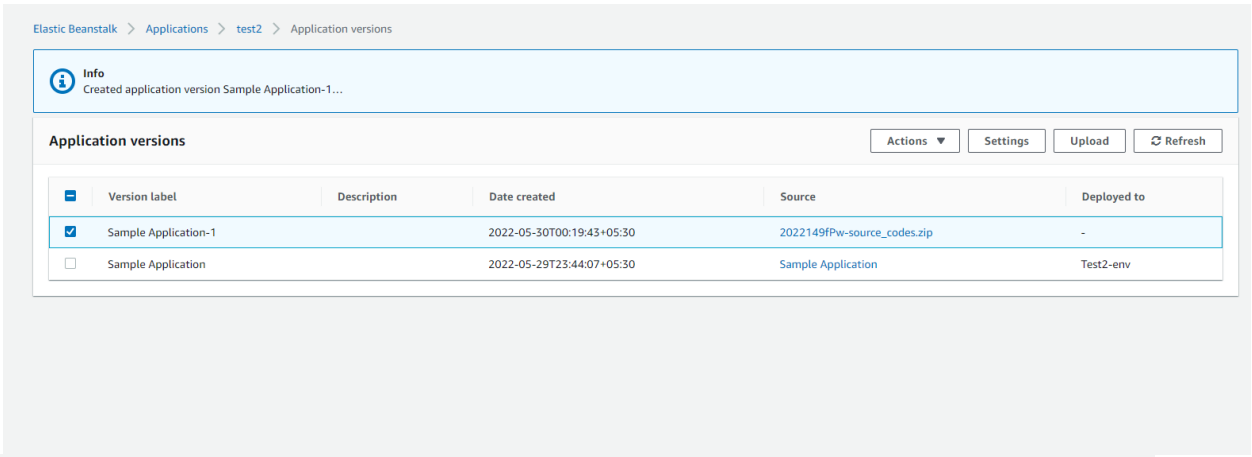
Click on name of the application you created

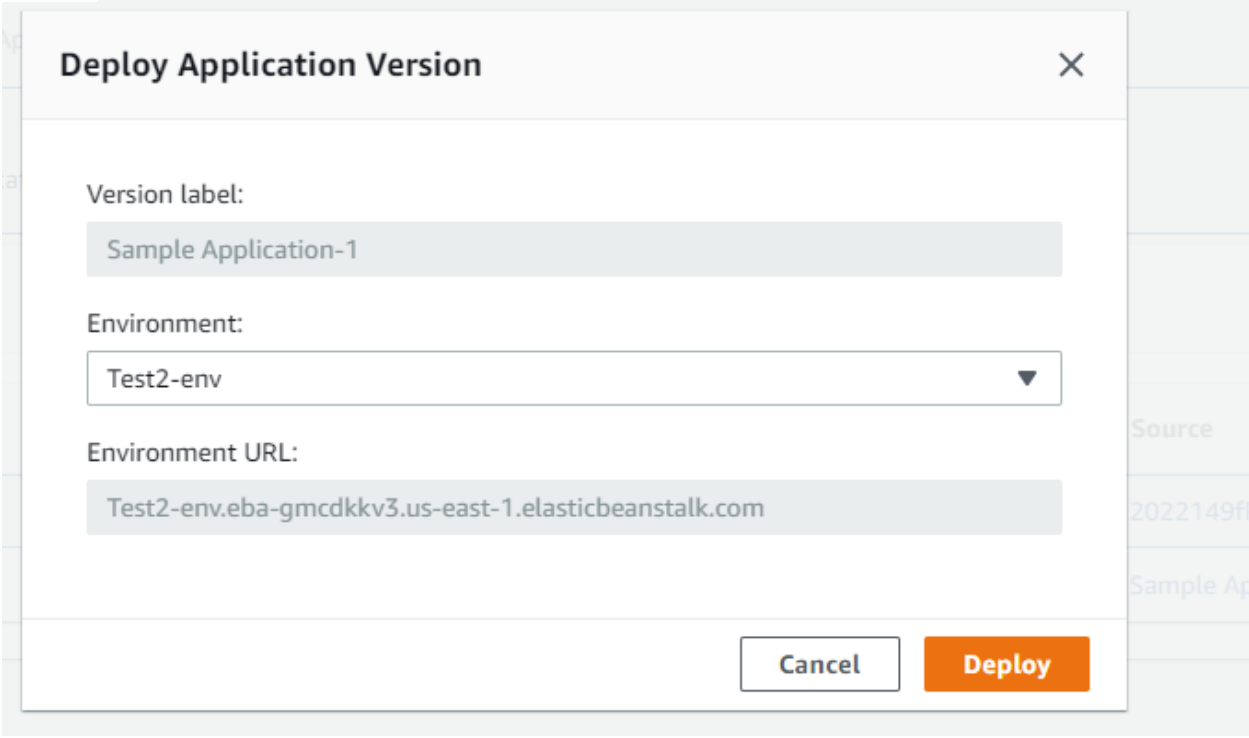
Go to application versions



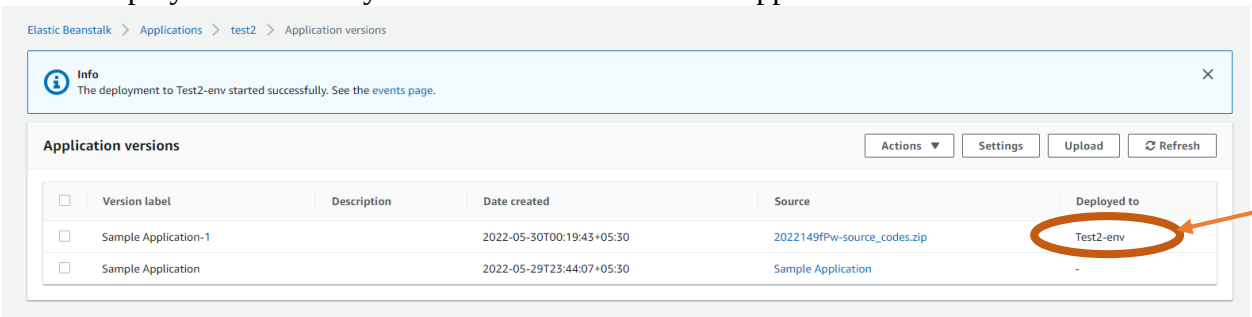


8. Upload
   Upload the source_codes.zip file here
   Select latest version -> actions -> deploy

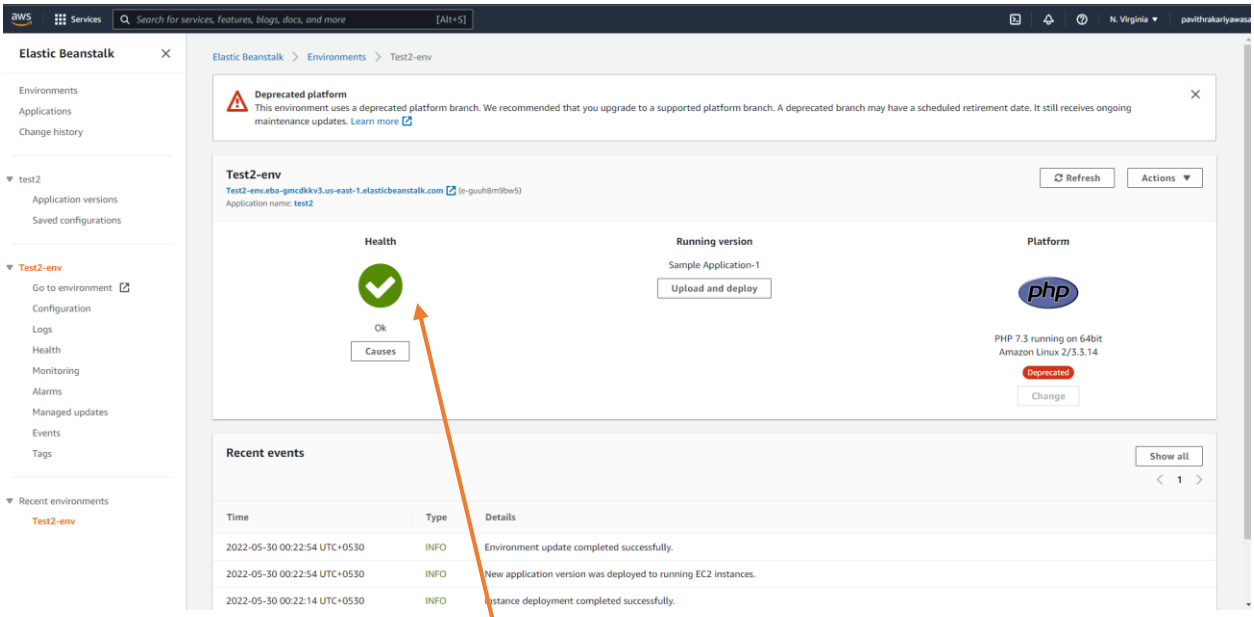




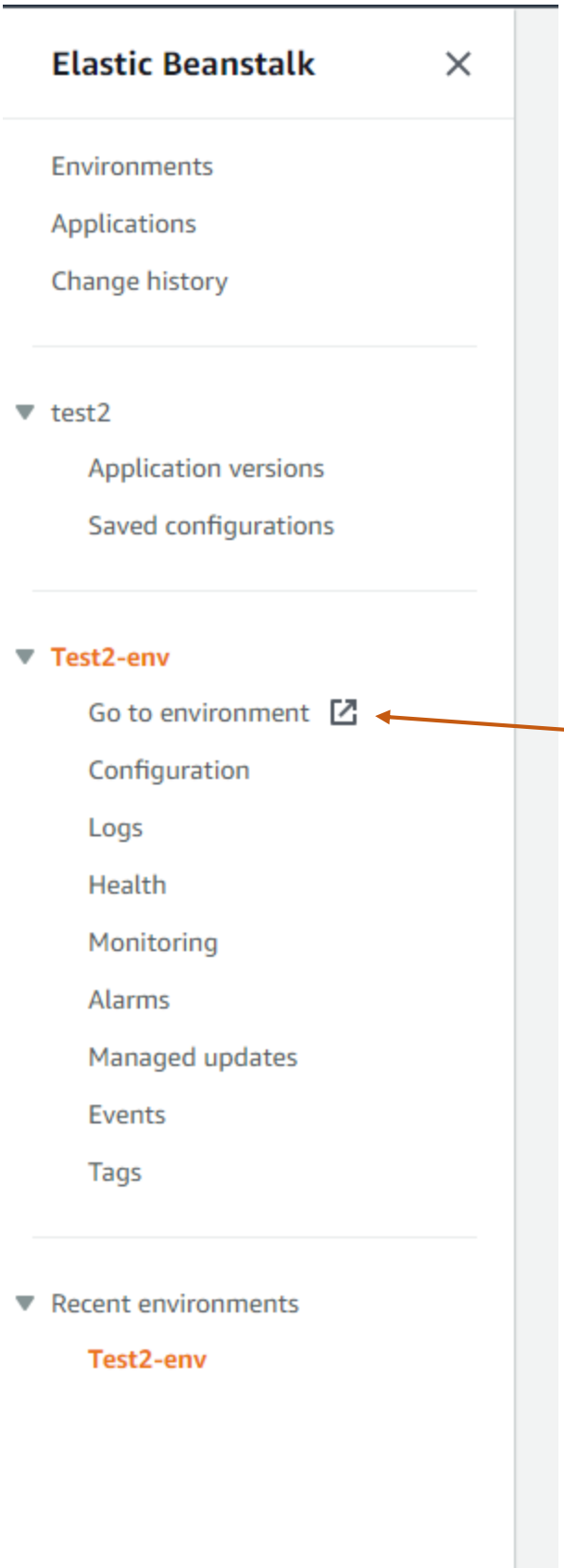
Wait few minutes.
Under deployed to section your environment name will appear



You can also check with this



Always health should be green color



Go to environment

Check ec2 - https://us-east-1.console.aws.amazon.com/ec2/v2/home?region=us-east-1#Home:
There will ec2 instances, load balancers created
Check RDS - https://us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#databases:
There will be a database

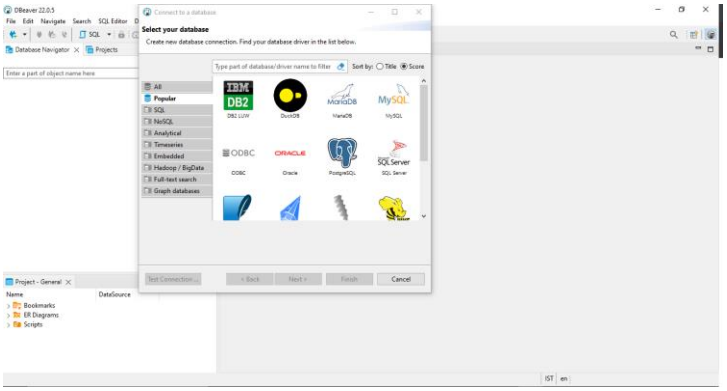
RDS

Get dbeaver (you can use any database tool)

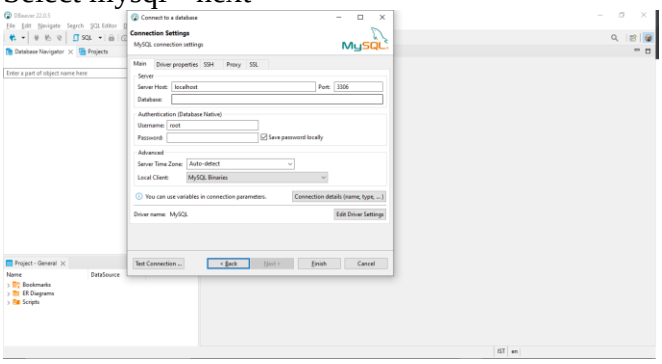
https://dbeaver.io/download/

install dbeaver

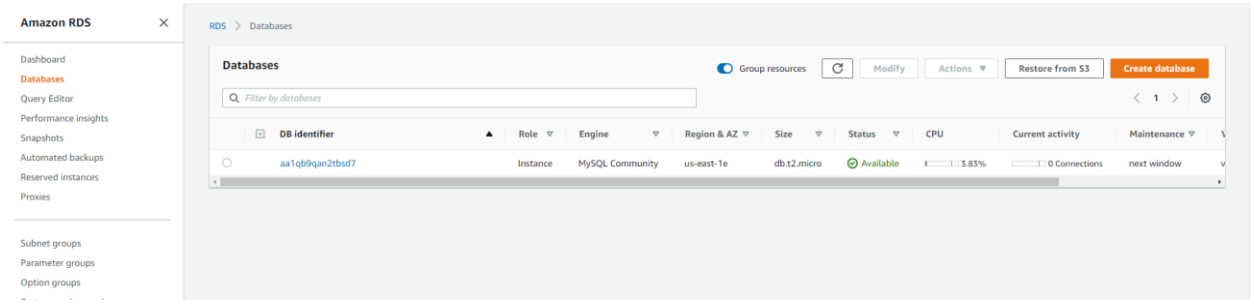
1. run dbeaver



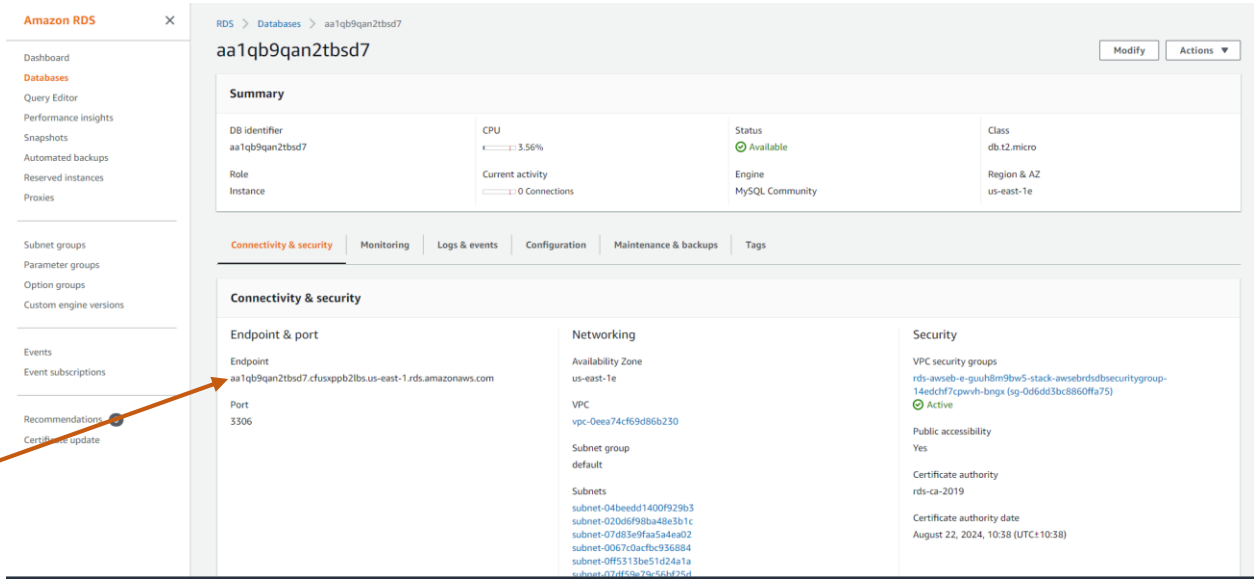
Select mysql->next



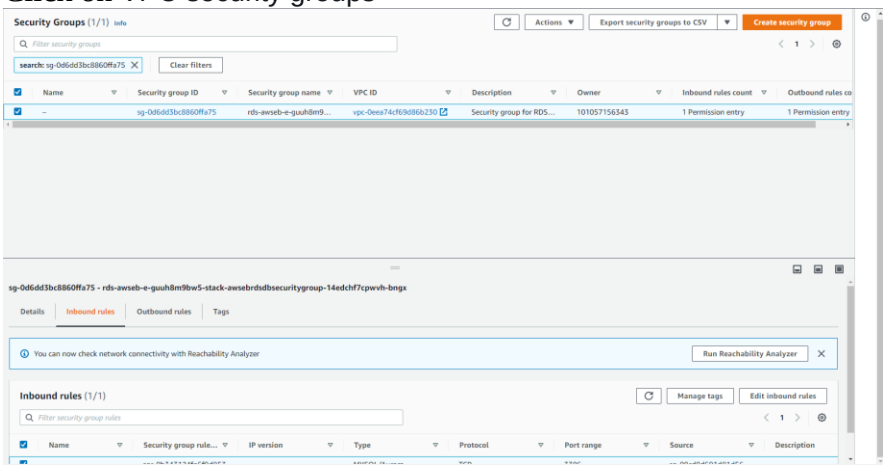
2. go to RDS dashboard https://us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#databases:



Click on db instance

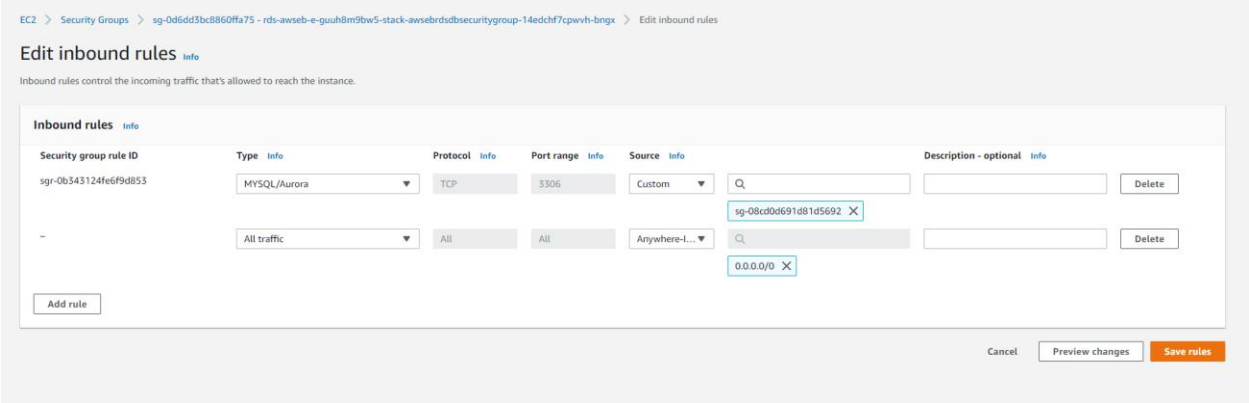


Click on VPC security groups



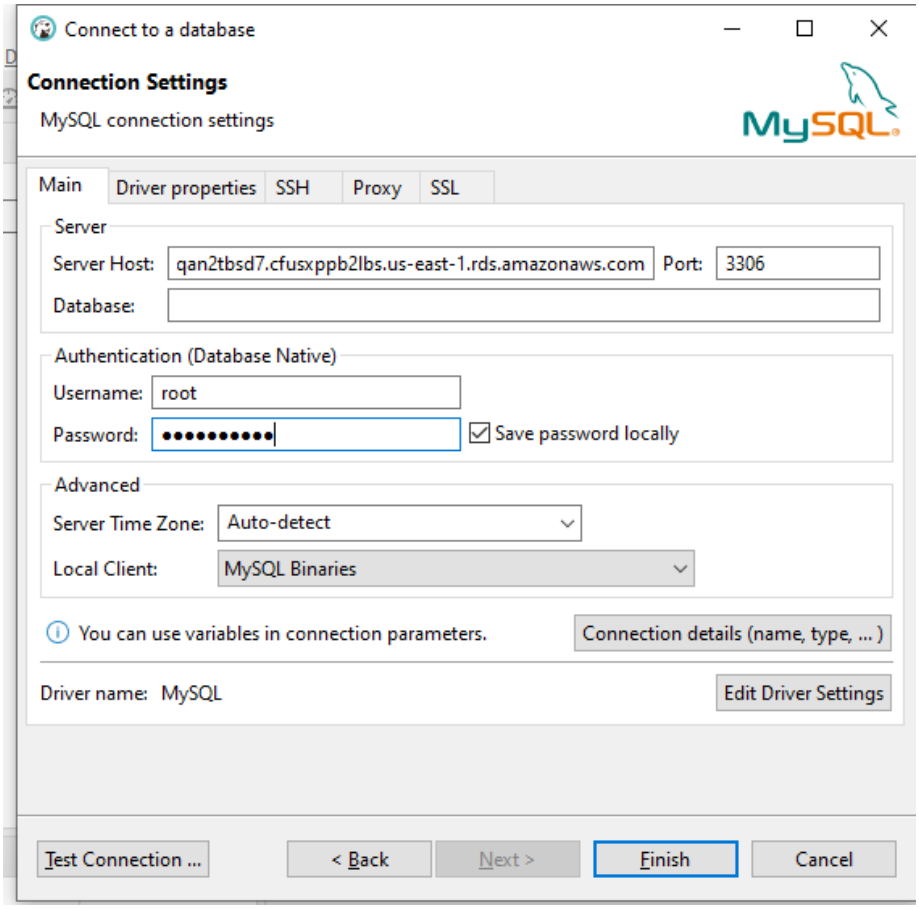
Edit inbound rules -> add rule
Type – all traffic

Source – Anywhere-IPV4



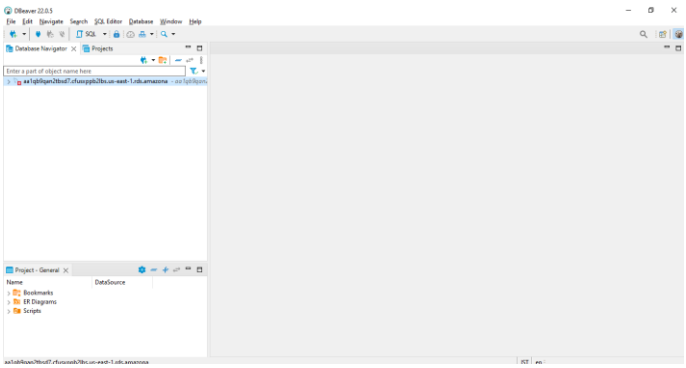
Save

Copy endpoint and go back to dbeaver



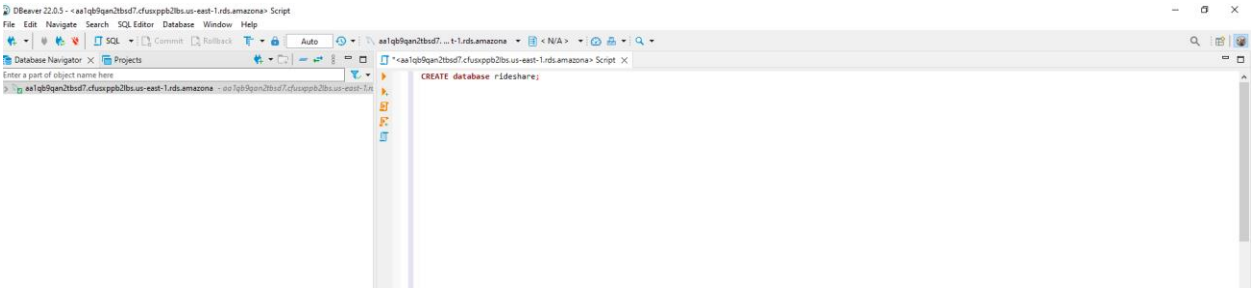
Server host – endpoint copied from rds
Username – assigned by you when creating RDS in beanstalk (step 5)
Password - assigned by you when creating RDS in beanstalk (step 5)

Finshed

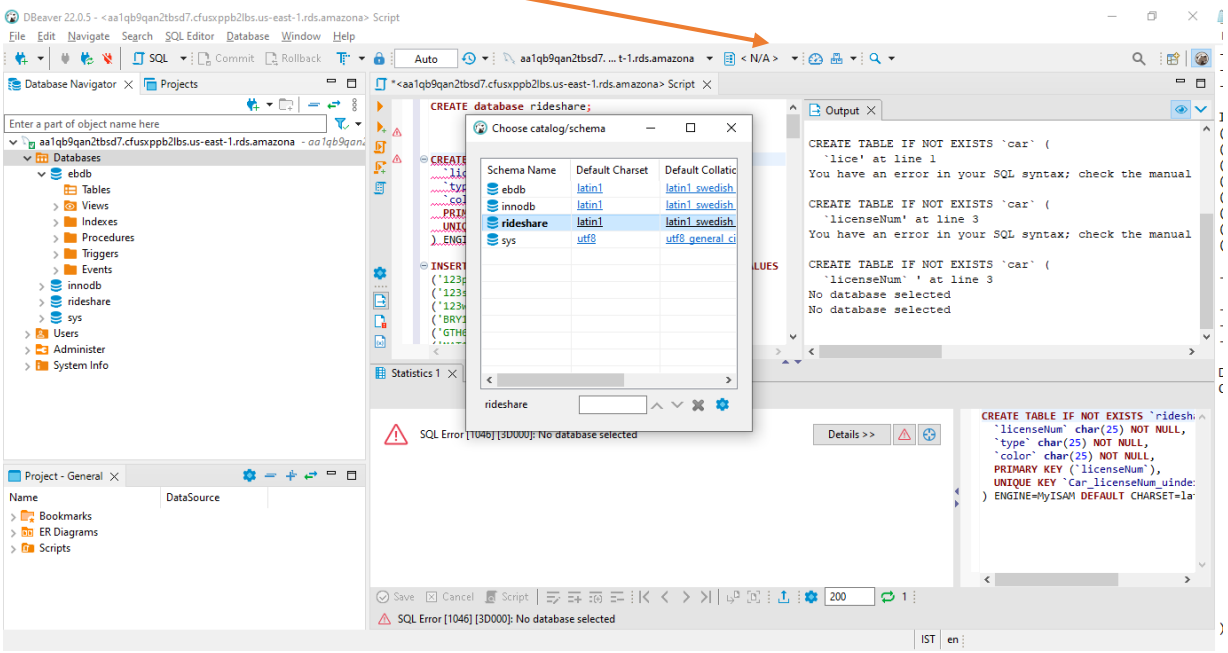


Right click on connection -> connect

3. create database
   right click on database object -> sql editor -> new sql script
   run following sql query
   ```
   CREATE database rideshare;
   ```



Select database



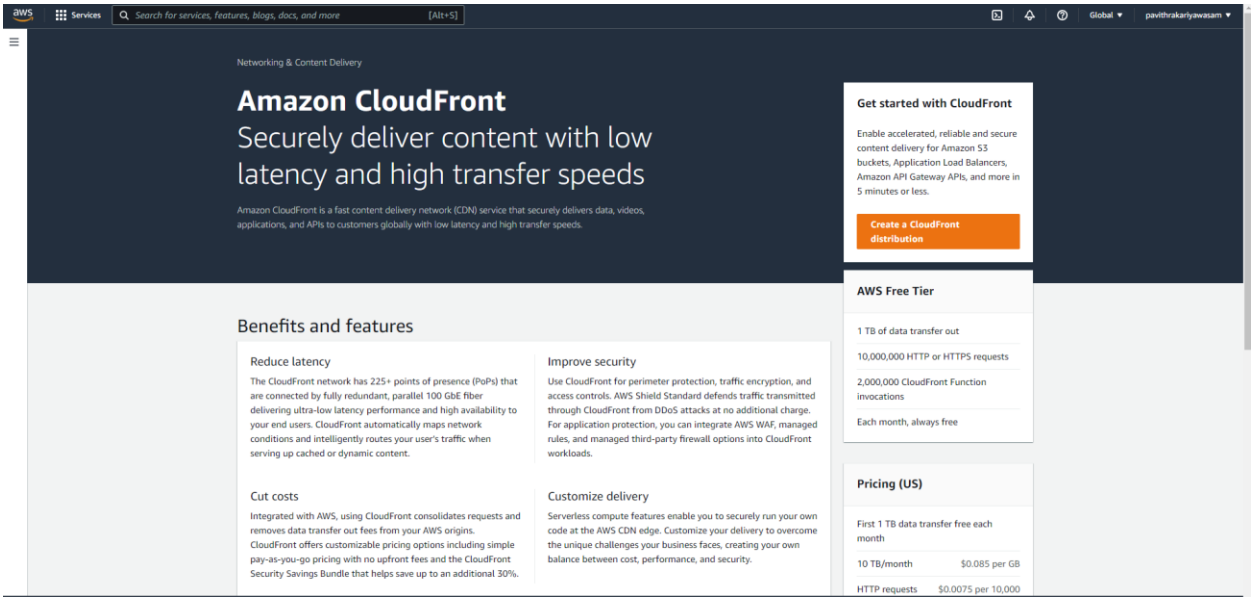
NOTE: If database rideshare didn’t appear right click and refresh database object

4. run sql queries one by one (script.sql)

cloudfront

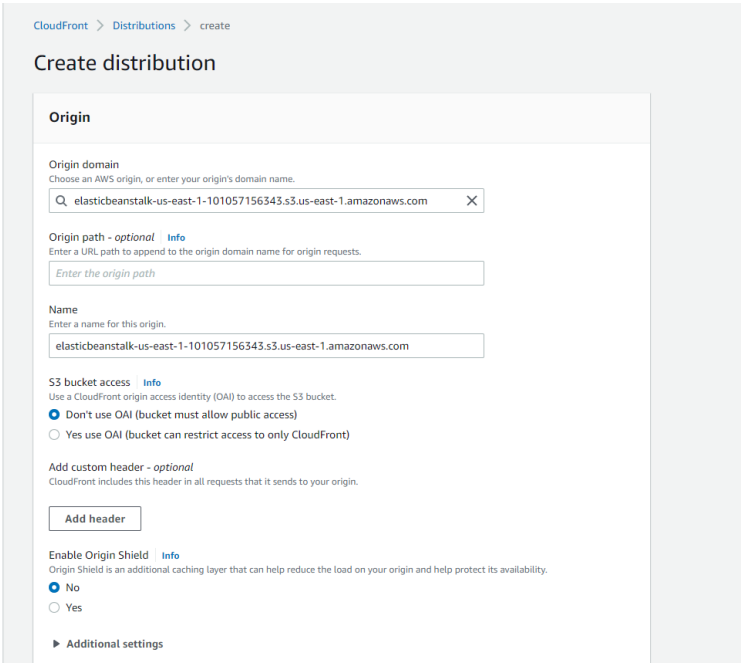
https://us-east-1.console.aws.amazon.com/cloudfront

1. create cloudfront distribution

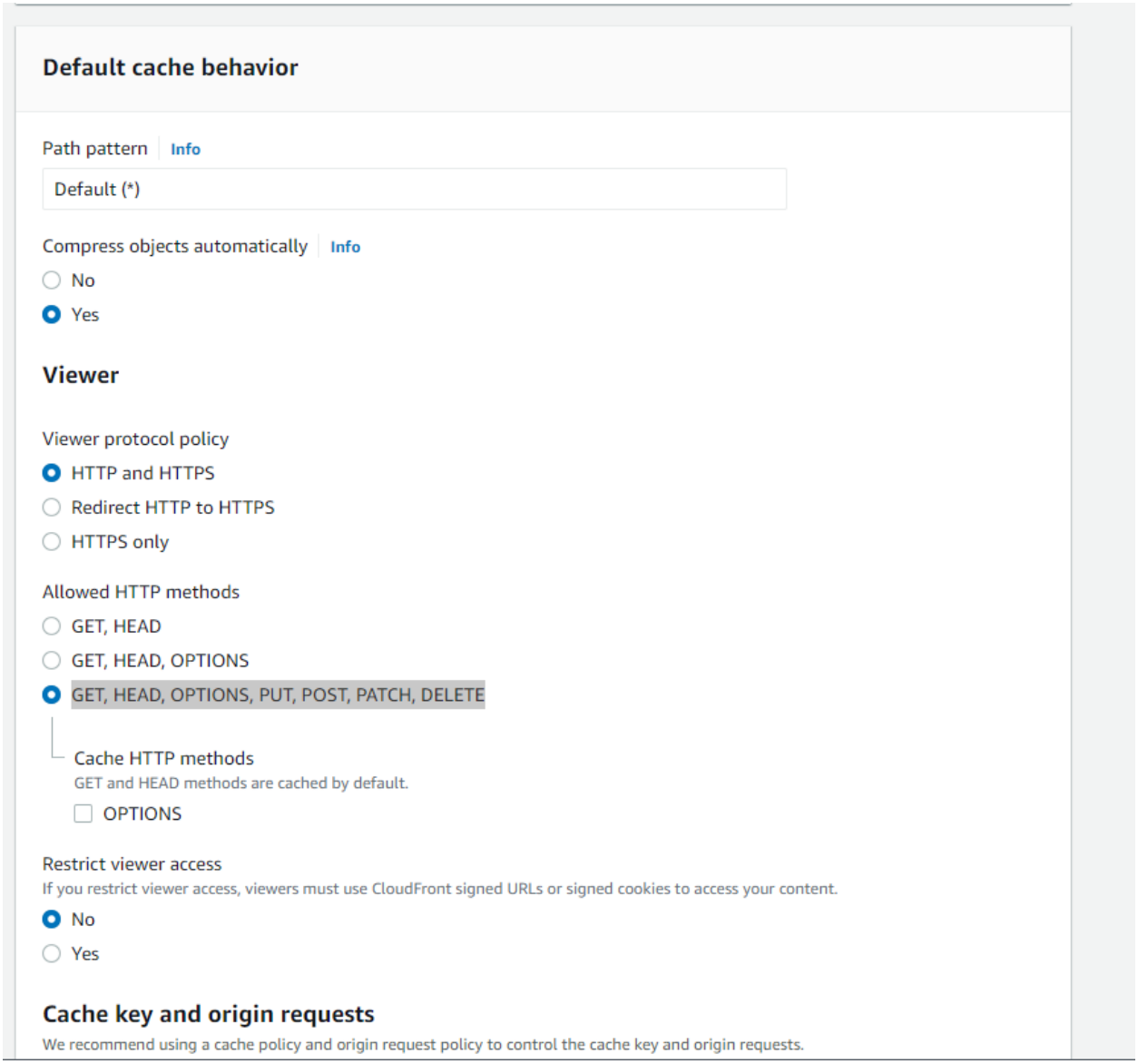


1. create distribution

select origin domain which url start with elasticbeanstalk….



In View section select - GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE.



Create distribution

Check s3
https://s3.console.aws.amazon.com/s3/buckets?region=us-east-1

there will be a S3 folder where name start with elasticbeanstalk

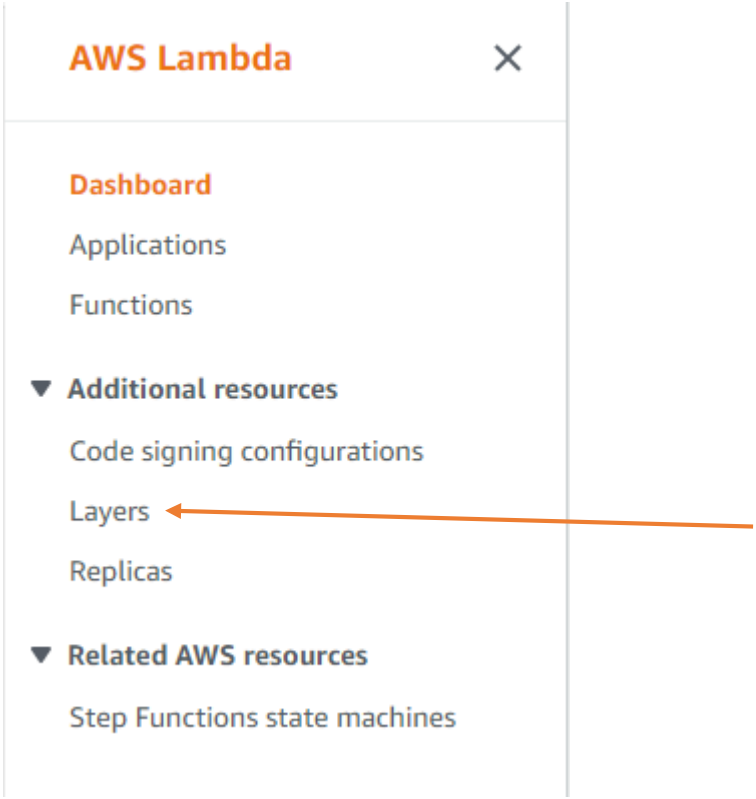
within folder your source code zip file will be available.

This folder created when steps completed in beanstalk and updated when cloufront steps completed.

LAMBDA

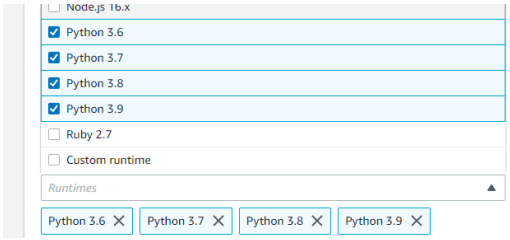
https://us-east-1.console.aws.amazon.com/lambda

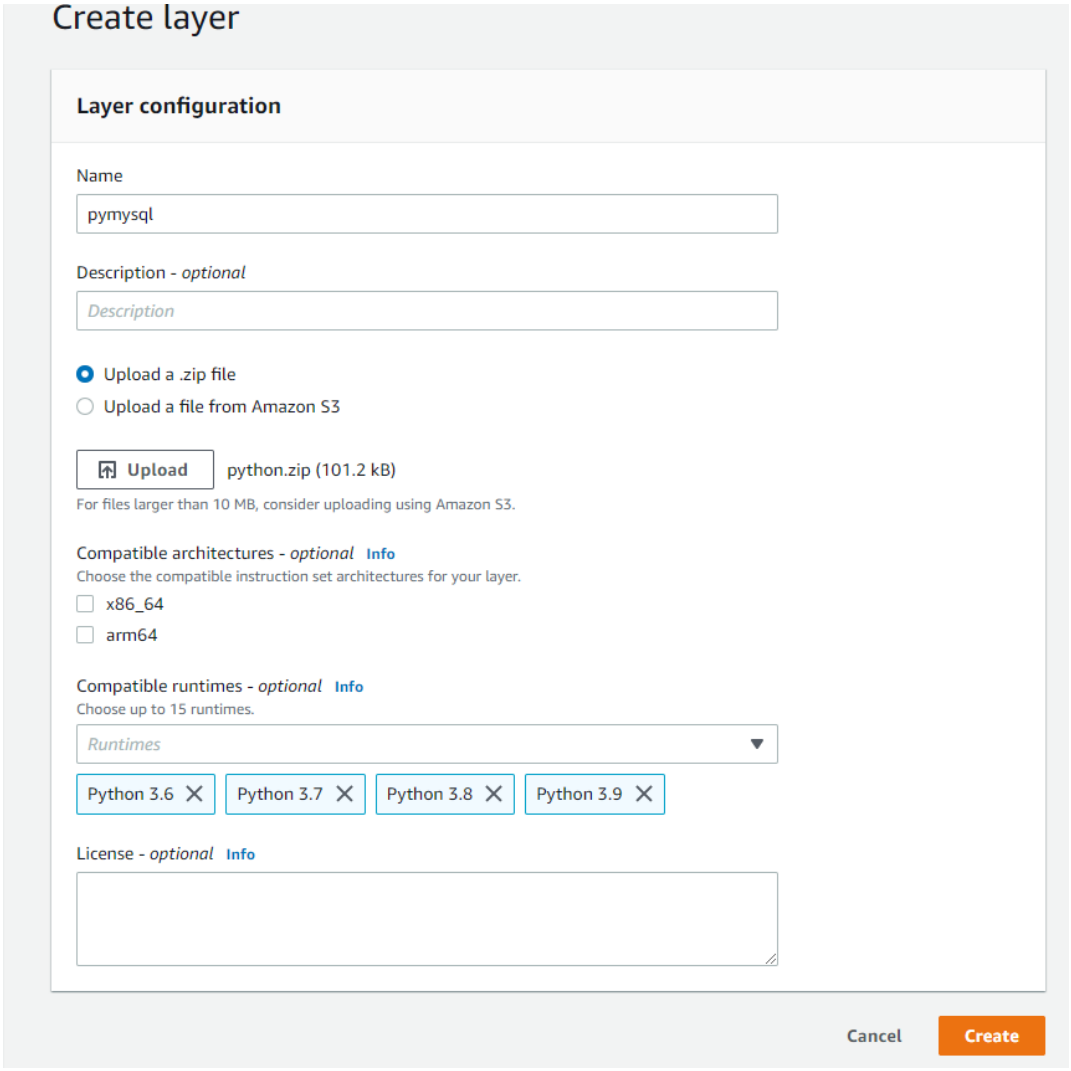
1. create layer



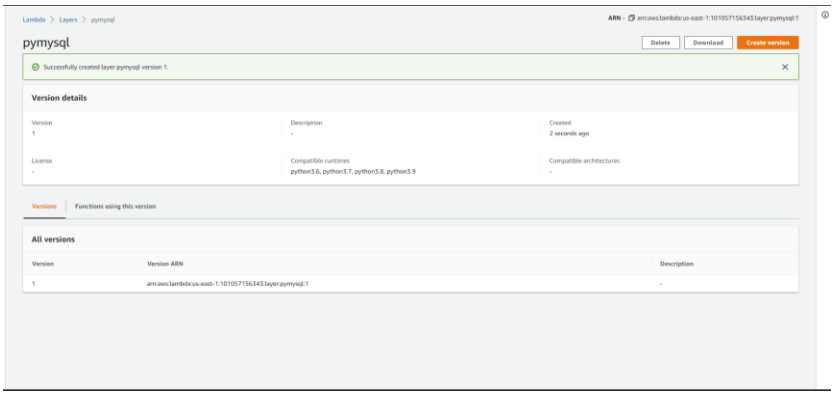
Create layer

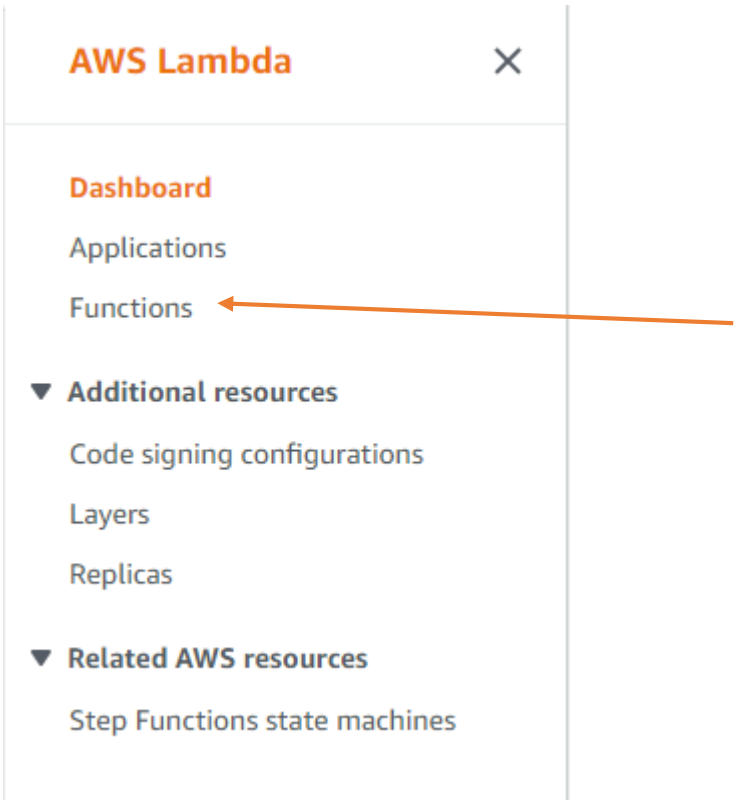
Layer name – pymysql
Upload – python.zip




Create layer
Layer configuration
Name
pymysql
Description - optional
Description
Upload a .zip file
Upload a file from Amazon S3
Upload
python.zip (101.2 kB)
For files larger than 10 MB, consider uploading using Amazon S3.
Compatible architectures - optional Info
Choose the compatible instruction set architectures for your layer.
x86_64
arm64
Compatible runtimes - optional Info
Choose up to 15 runtimes.
Runtimes
Python 3.6
Python 3.7
Python 3.8
Python 3.9
License - optional Info
Cancel
Create

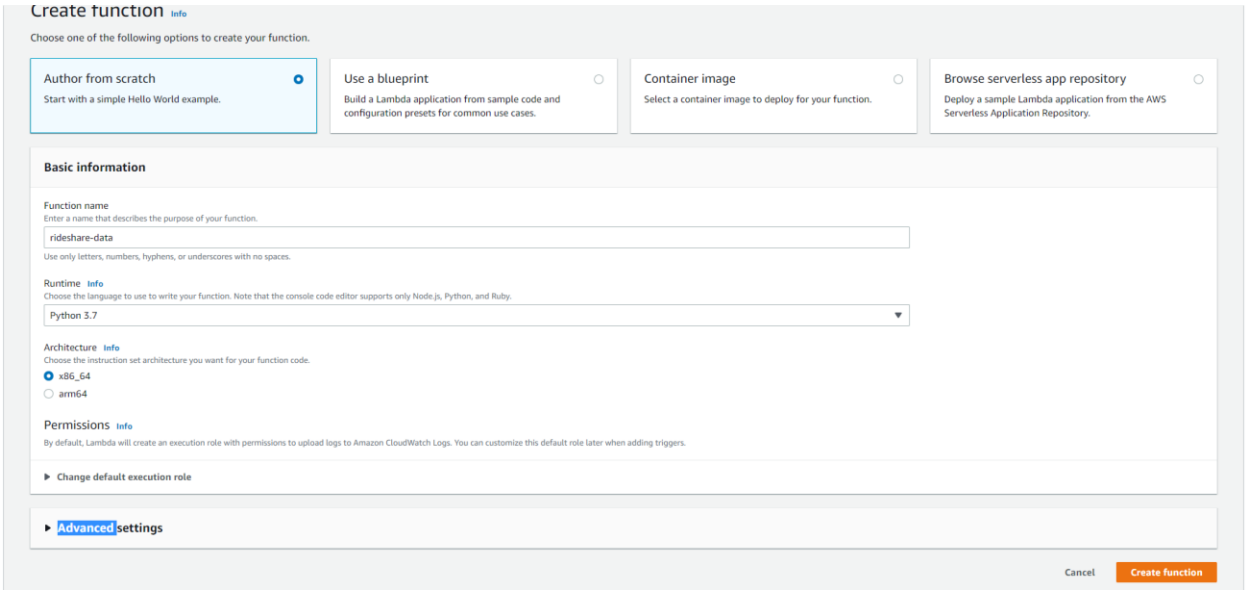
Create


pymysql
Delete
Download
Create version
Version details
All versions



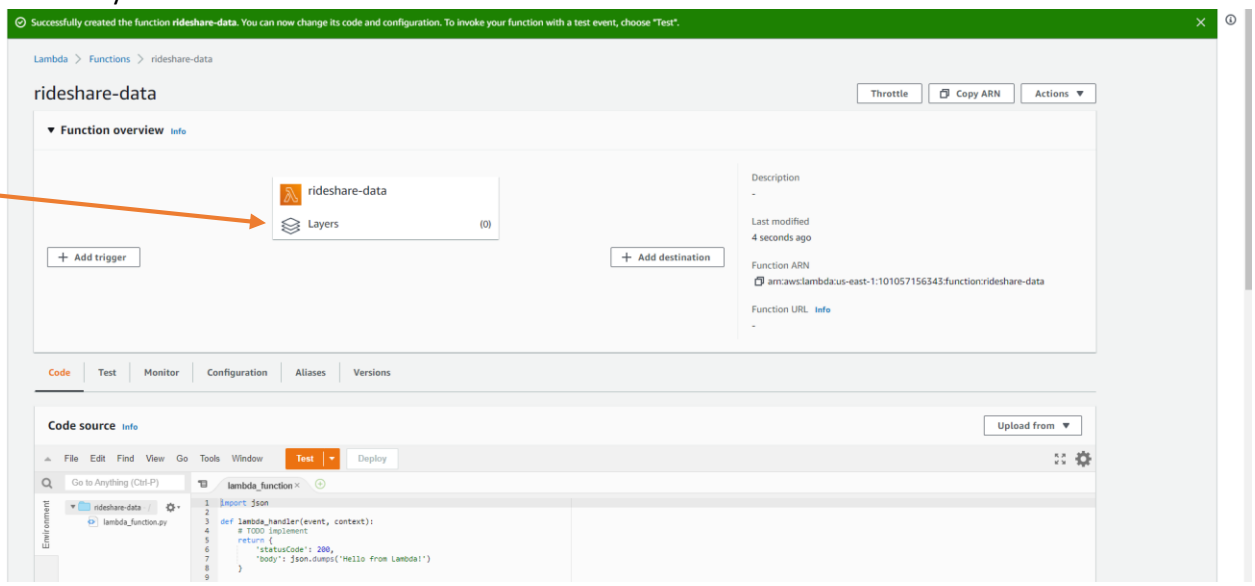
2. Create function

Function name - Any function name
Runtime – python 3.7

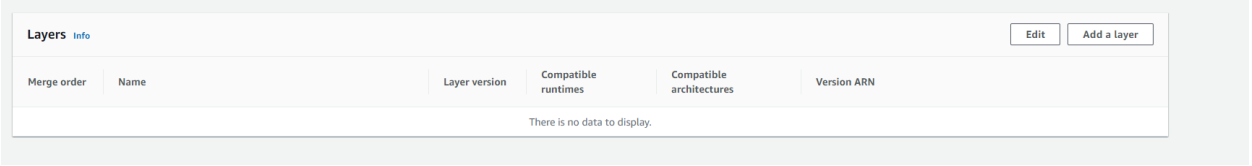


Create function

3. Attach layer



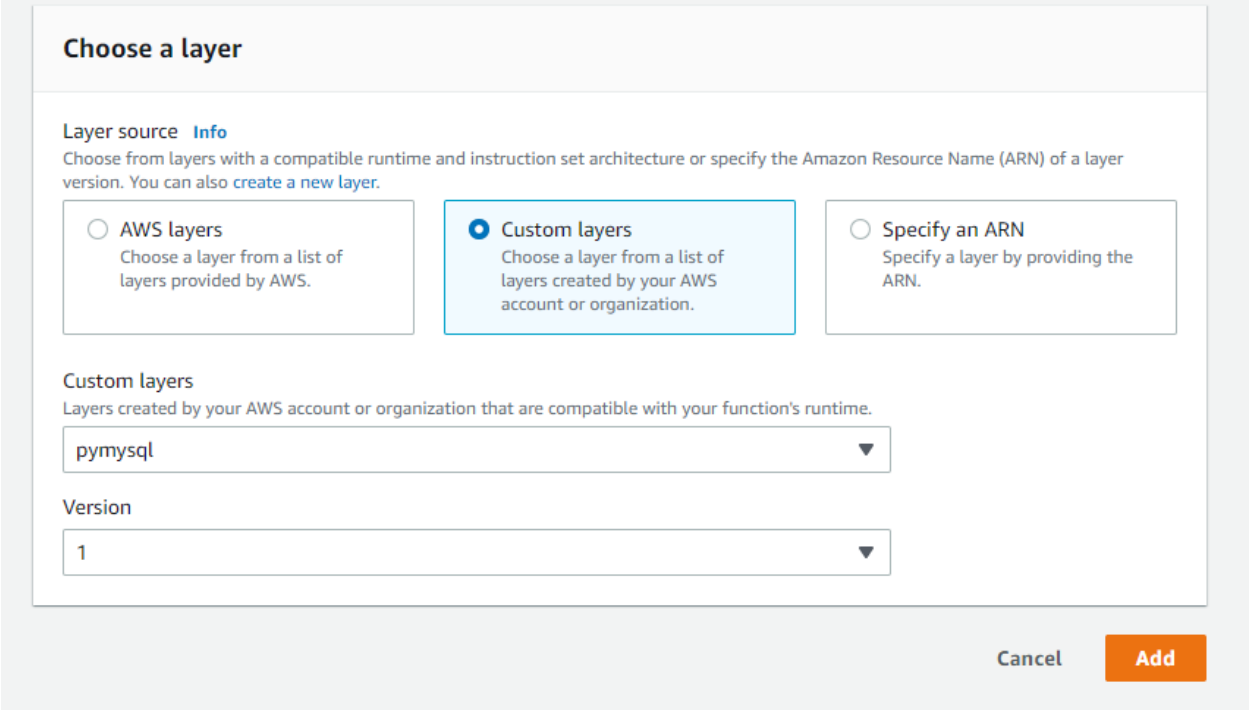
Add layer



Custom layers

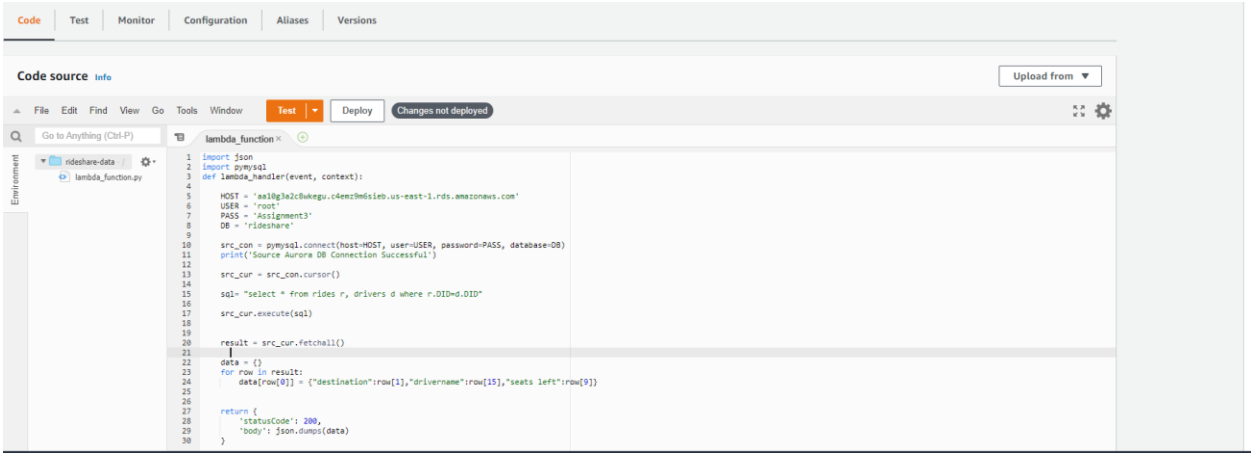
Select the layer name you create previously

Select latest version



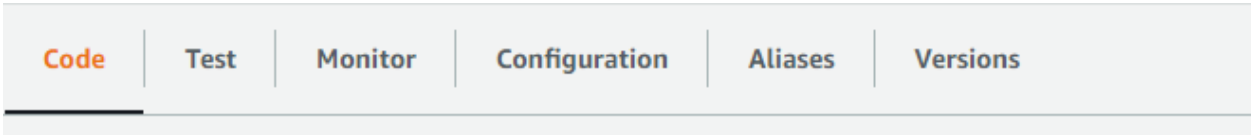
add

4. Add code
   Delete existing code and paste following(lambda.py code)
   Change database credentials (host, user, password, db)(RDS credentials)

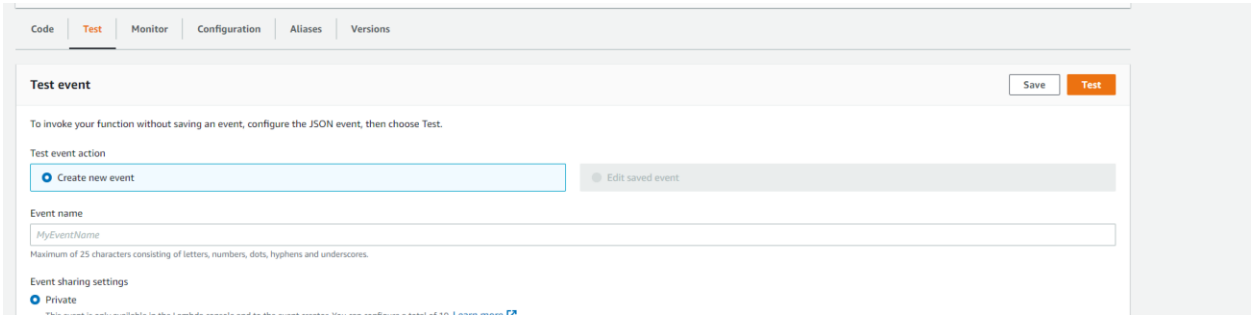


Deploy

5. Go to test



test



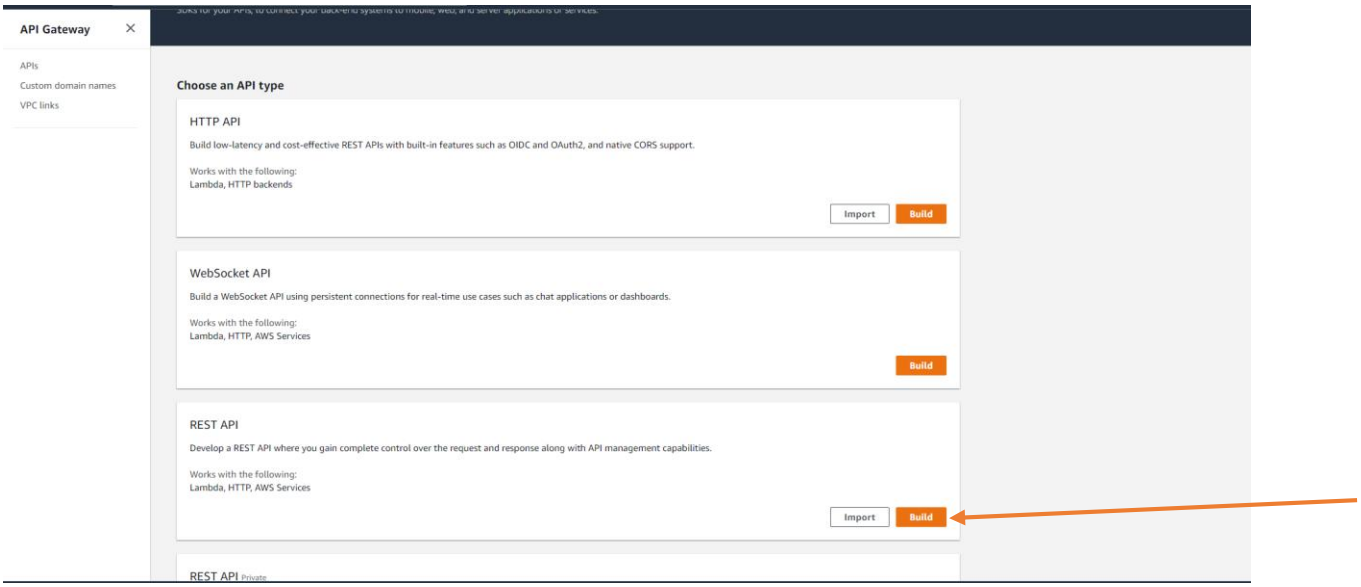
Function should run correctly

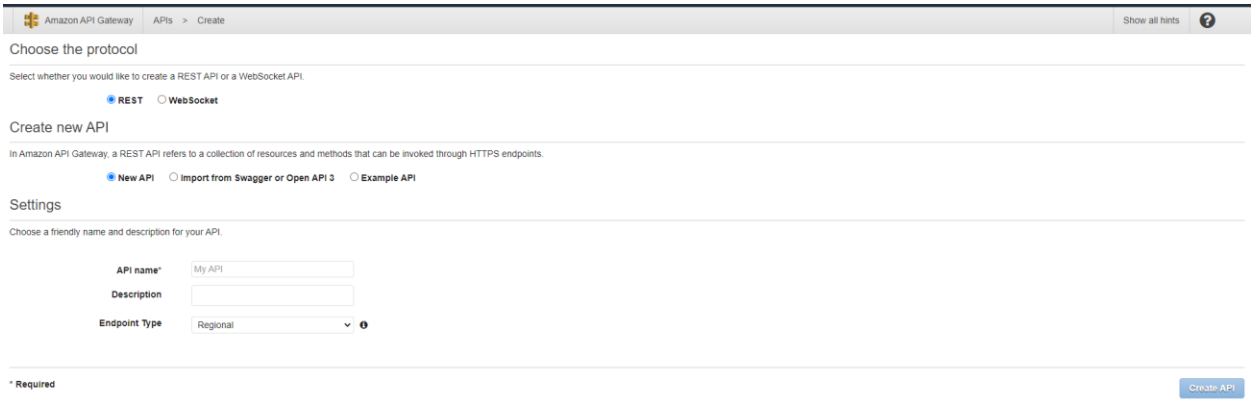
Similar output should come

API Gateway

https://us-east-1.console.aws.amazon.com/apigateway

1. Build – REST API



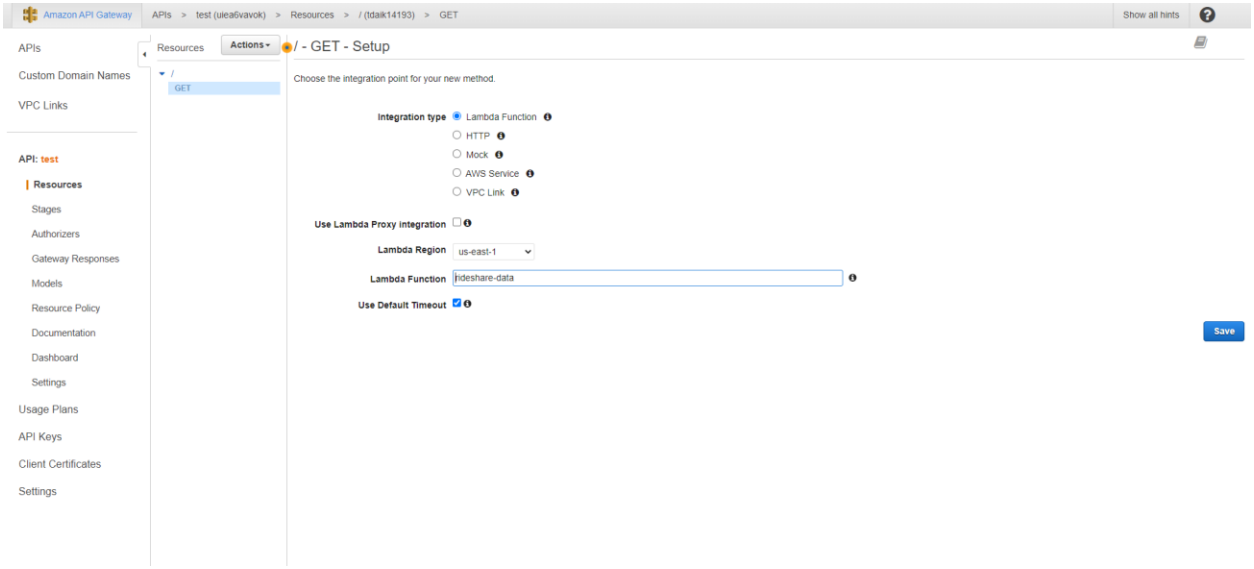


Give name to the API

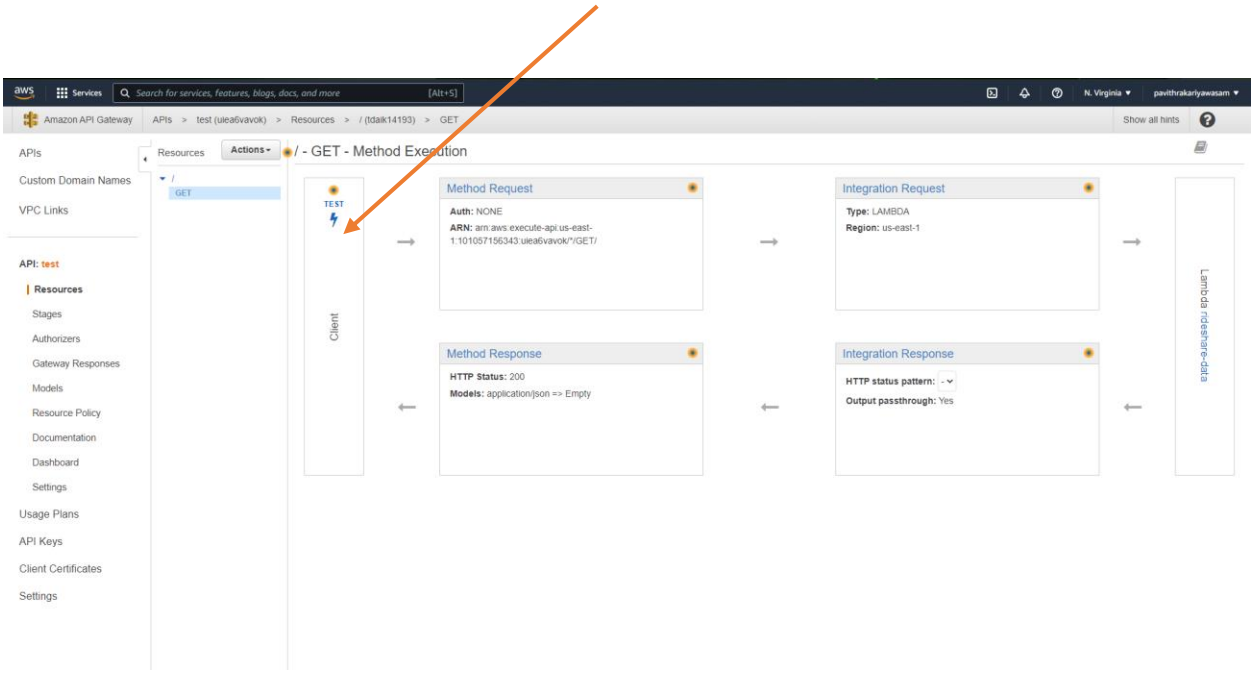
Create

Actions -> Create Method
Select method type as GET

Type name of the lambda function you created in previously

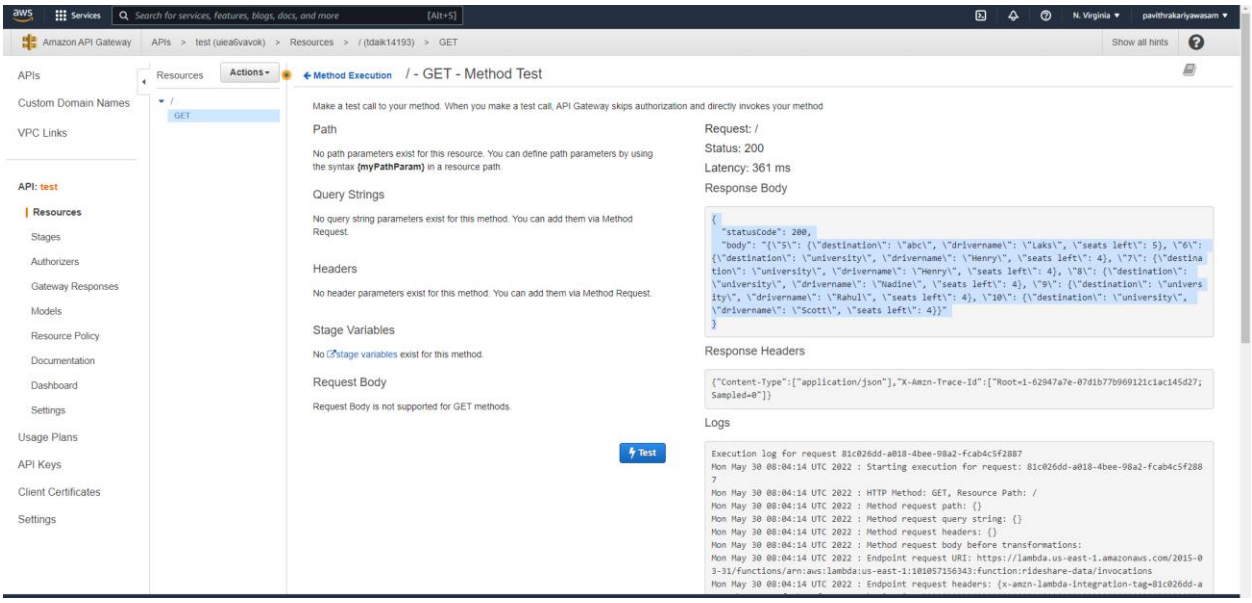


Save

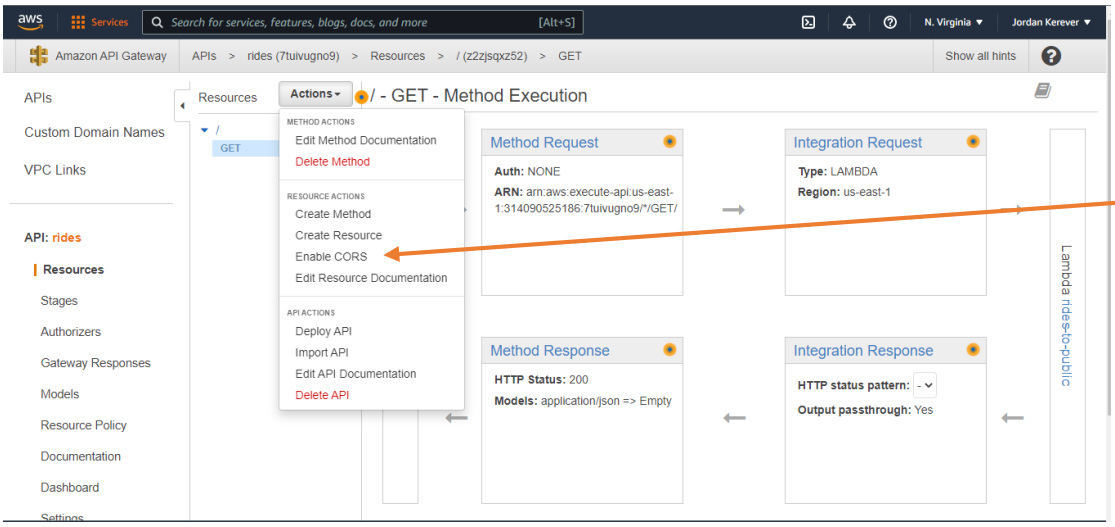


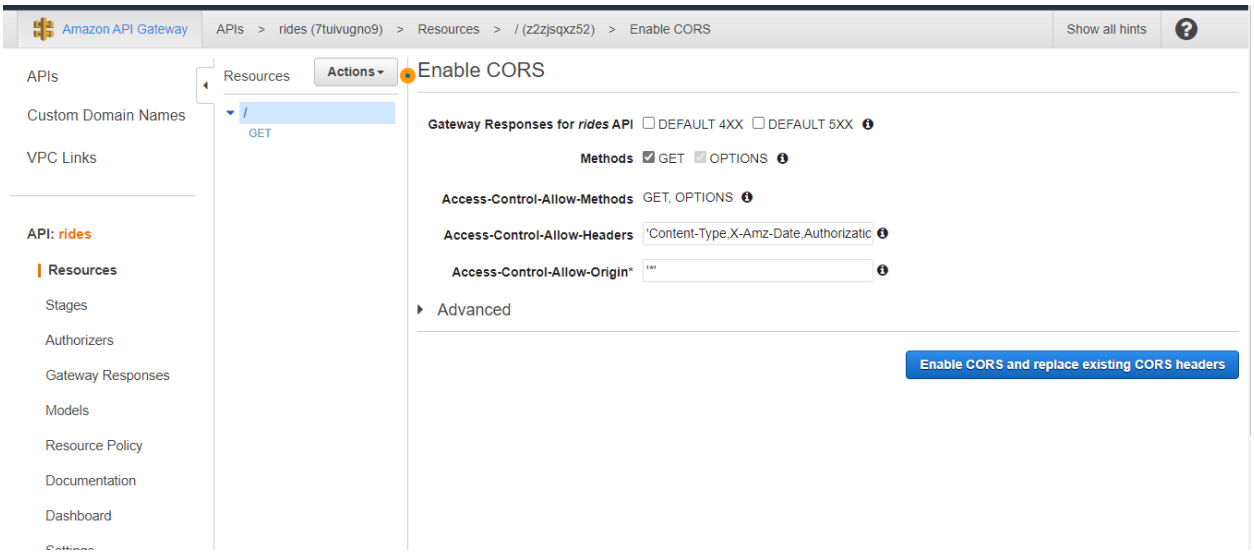
Click on test -> test

You will get a similar output

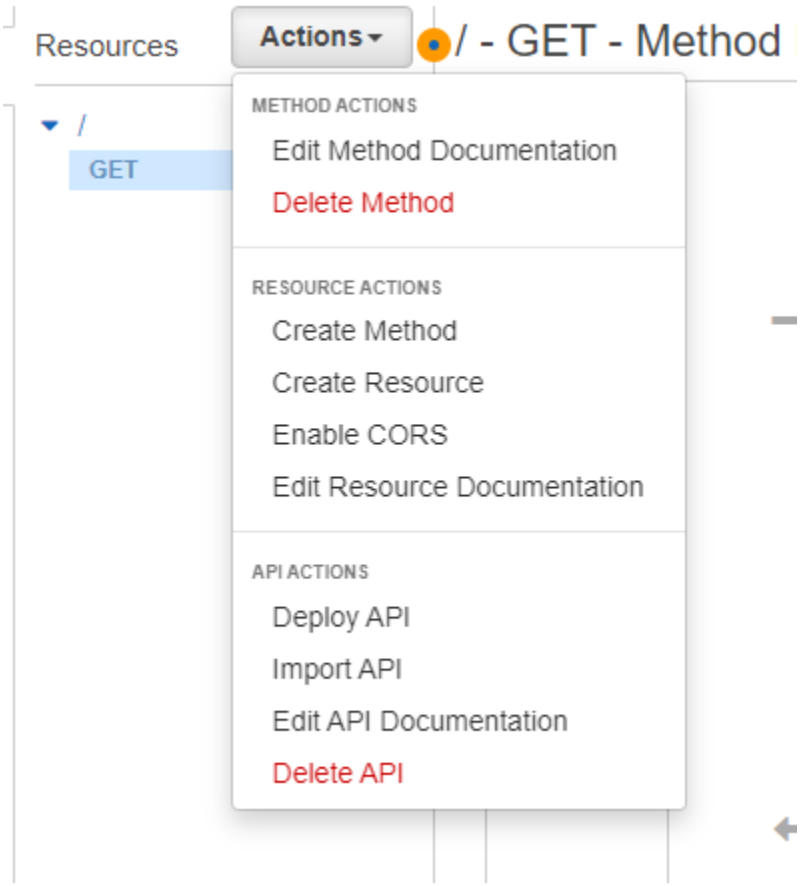


Actions -> Enable CORS

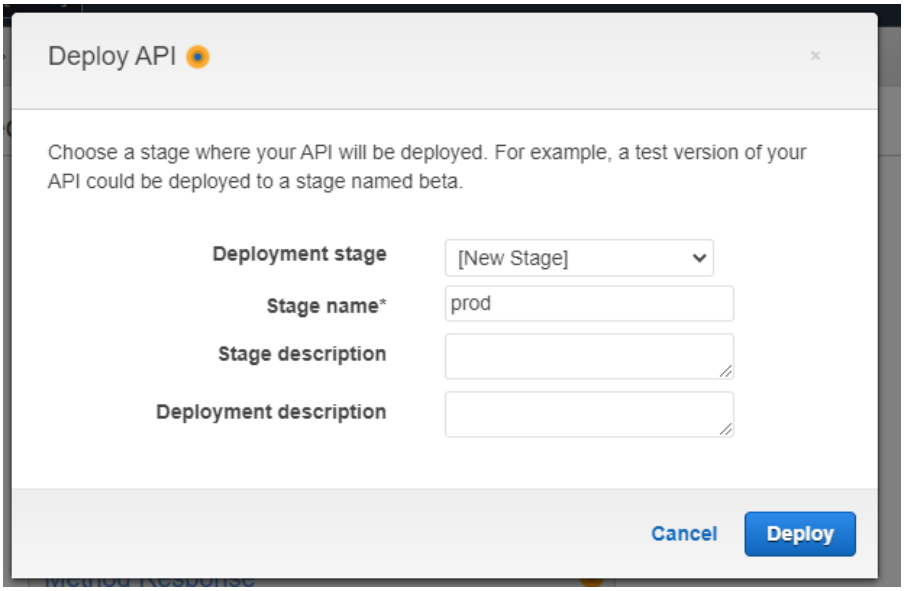




Action->Deploy API

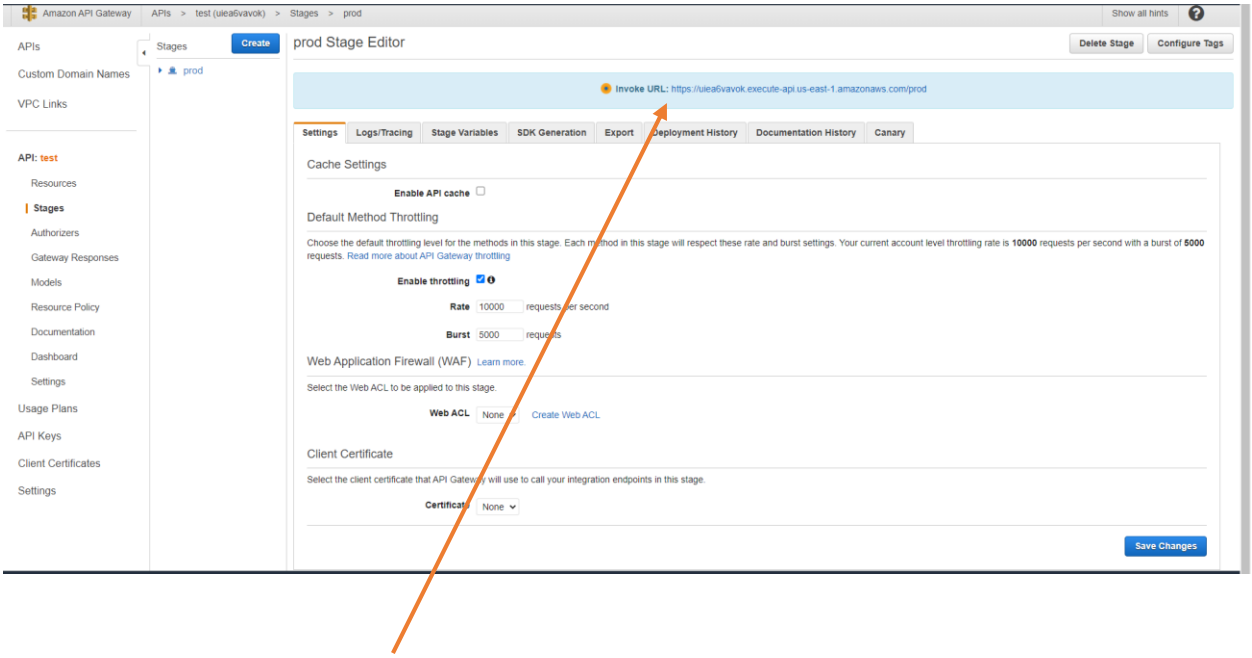


Development stage - select as new stage
Stage new – prod (any name stage name you prefer)



Deploy

Save changes



Check this url , data will be exposed
Keep a copy of this API URL

Update PHP project

Change API URL in the index.php line no 134

Change RDS credentials in db.php line no 6,7,8,9

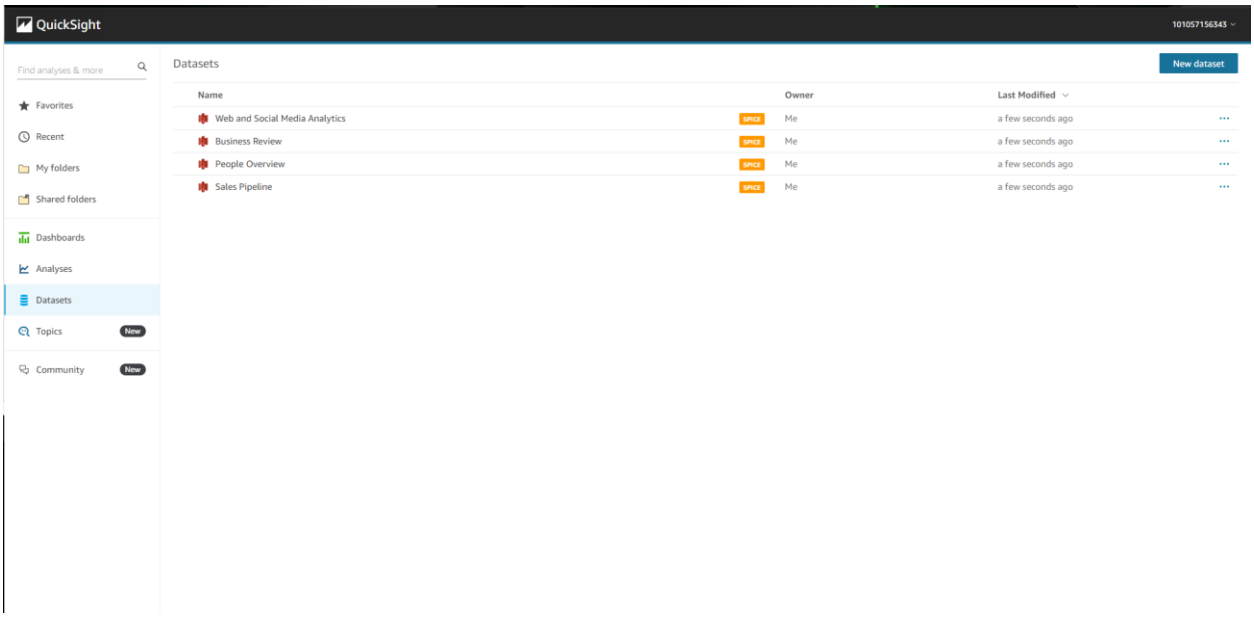
Zip all php files

Re do the step in Elastic Beanstalk step 7 (upload zip to beanstalk and deploy)

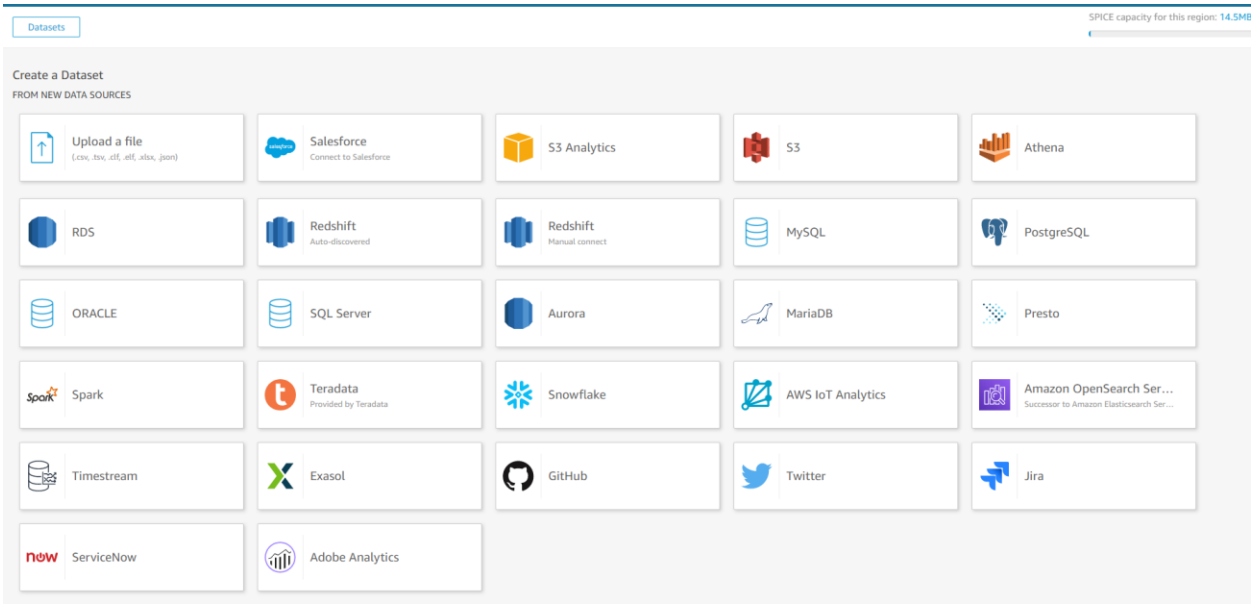
Quicksight

https://quicksight.aws.amazon.com/sn/start

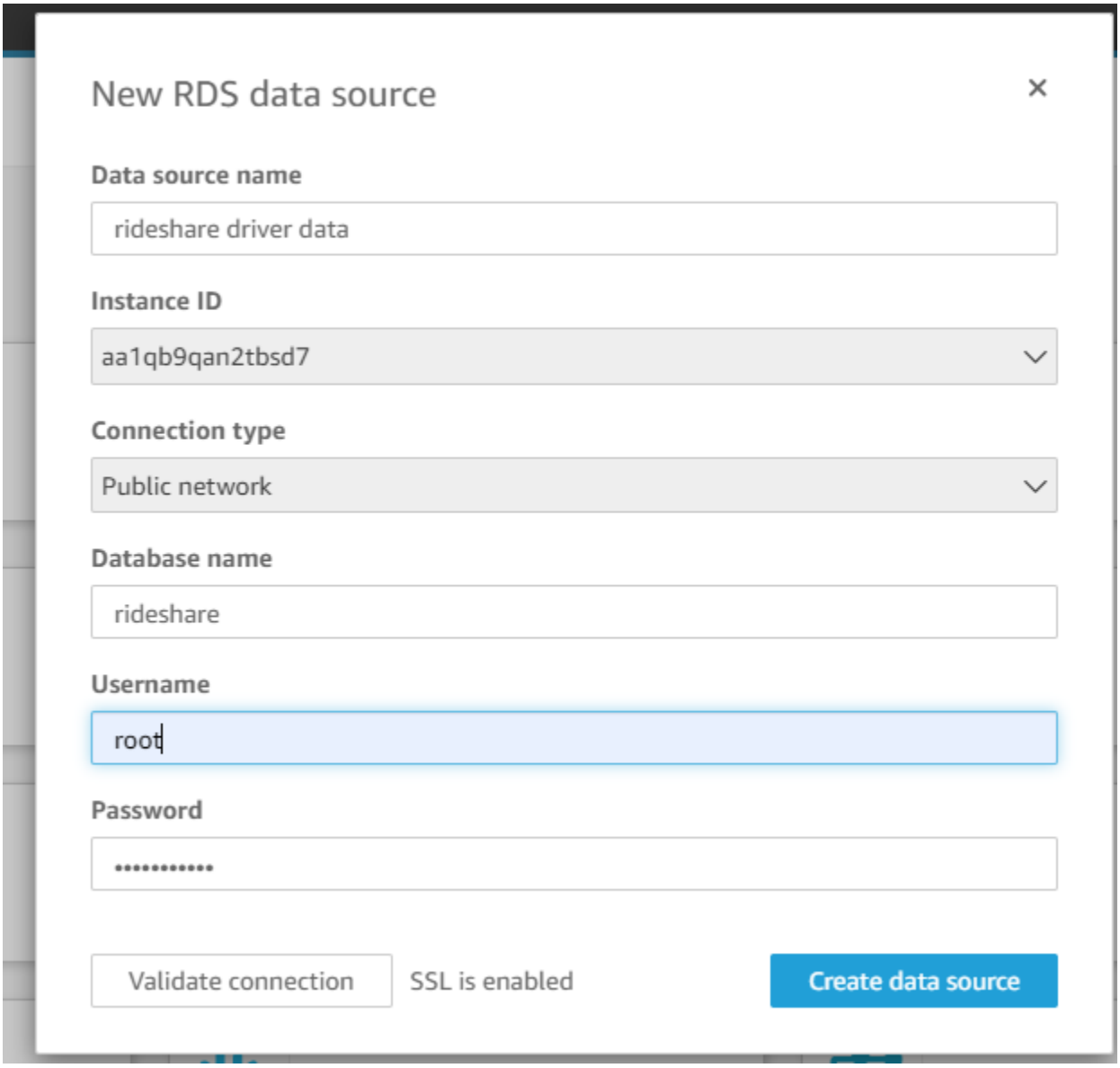
1. create dataset



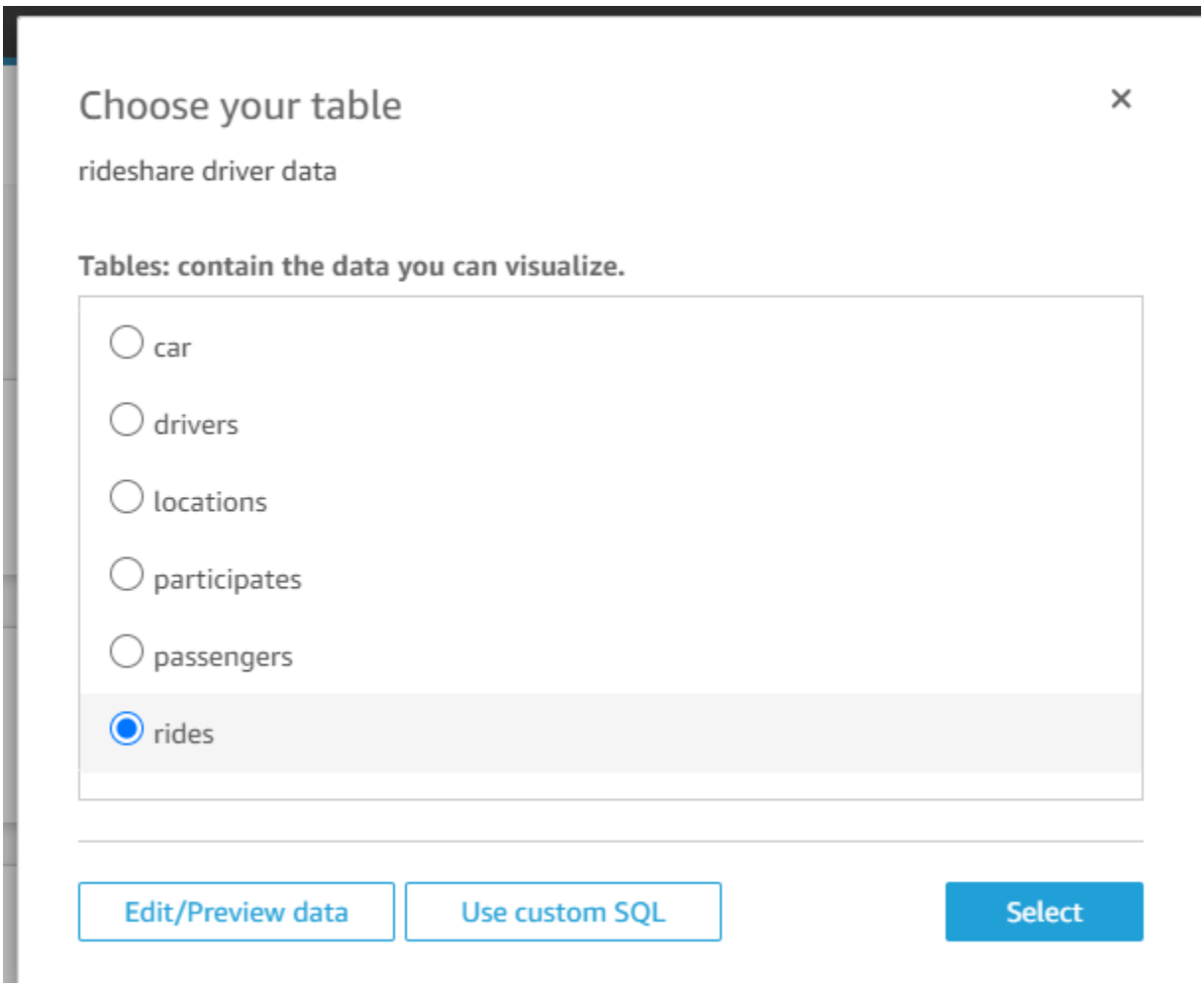
New dataset



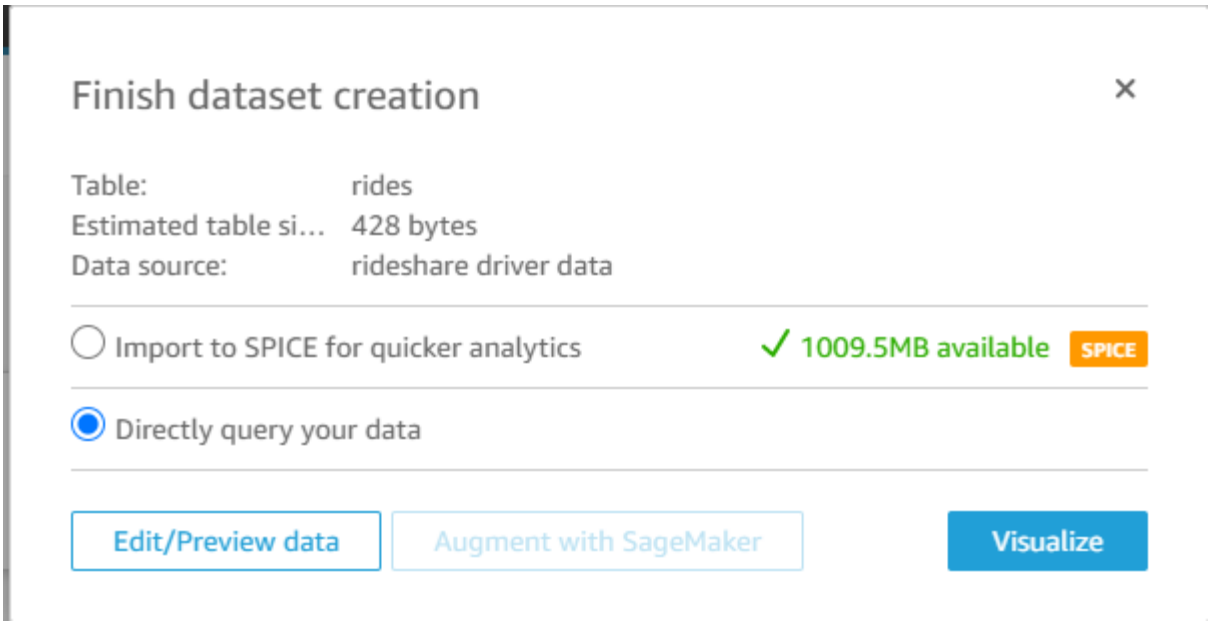
Select RDS



Give suitable name to dataset
Select RDS instance
Add database name, username and password
Create data source

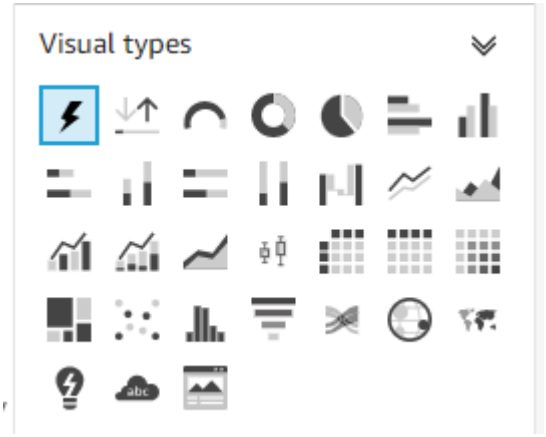


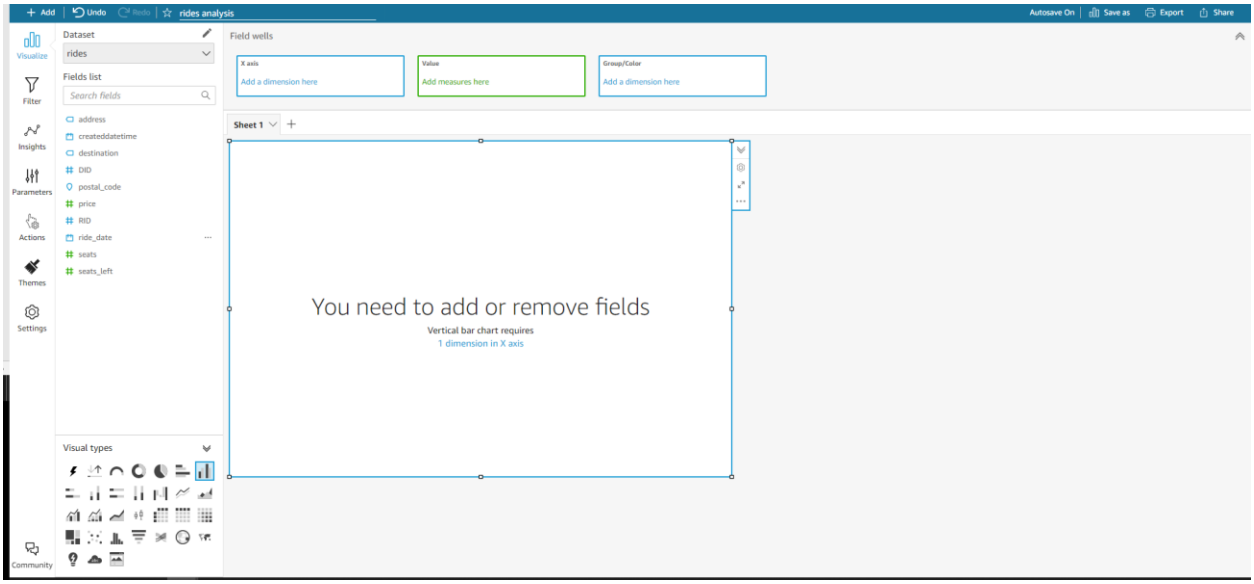
Select table you want
Select



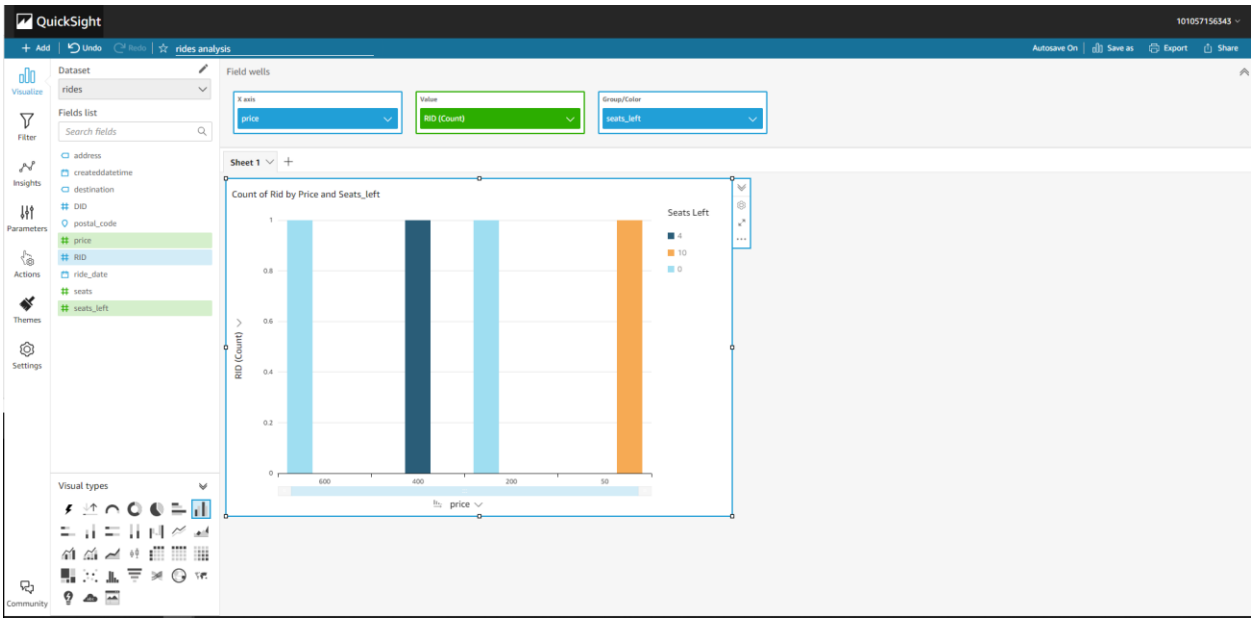
Select direct query your data
Visualize

Select any visual type (Example – select vertical bar graph)



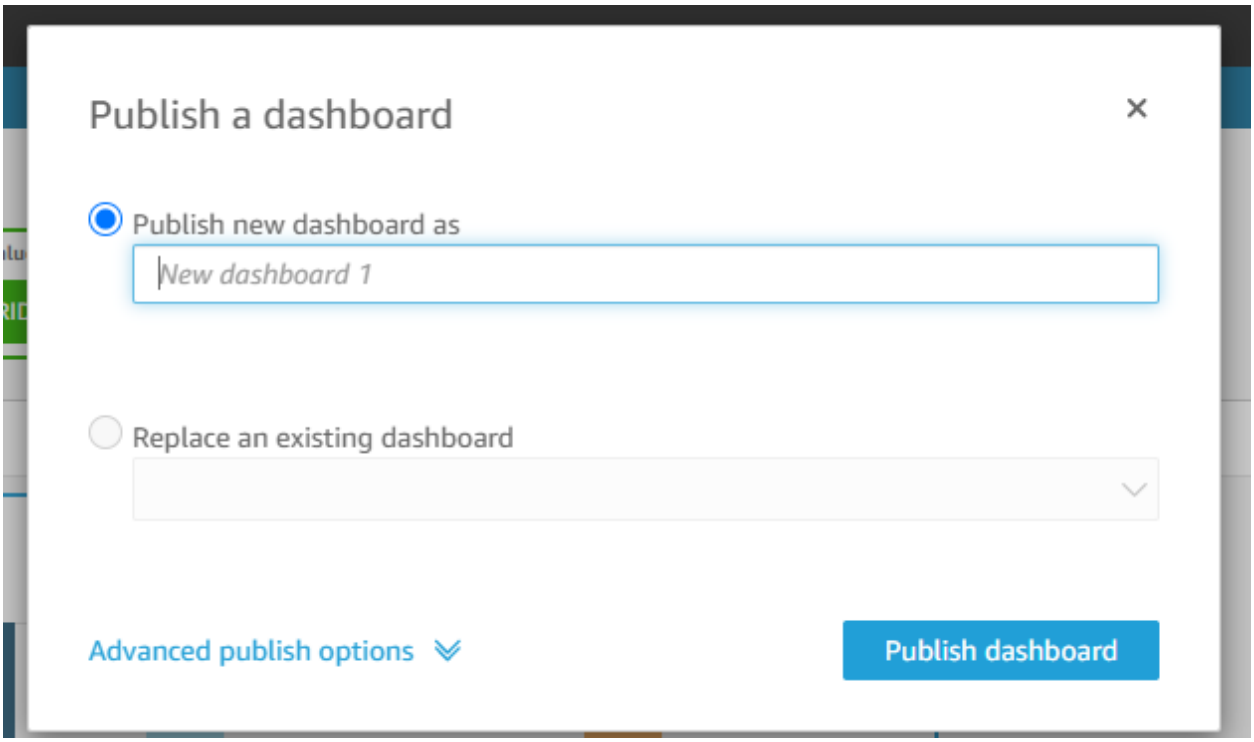


Select files for x axis, value (these changes according to visual selected)



Can get these type of charts

Share -> publish dashboard



Give name you prefer
Publish dashboard