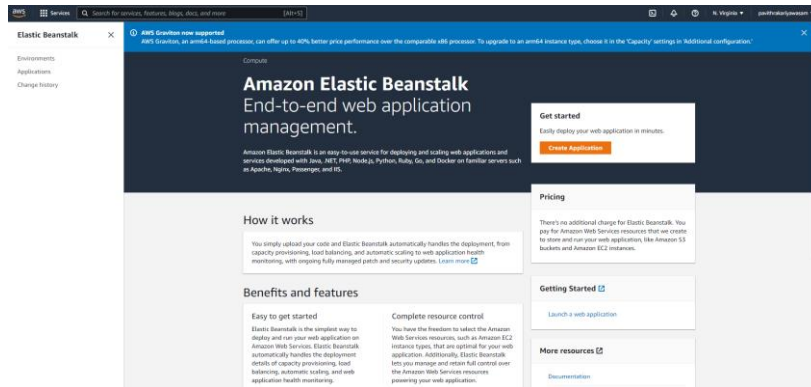


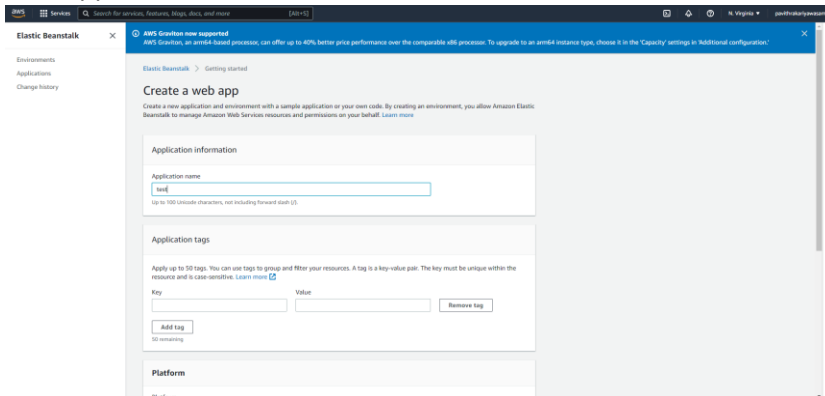
Elastic Beanstalk

<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

Platform

Platform
PHP

Platform branch
PHP 7.3 running on 64bit Amazon Linux 2

 **Warning**
Deprecated platform branches aren't recommended for use in production environments. [Learn more](#)

Platform version
3.3.14 (Recommended)

Application code

☒ **Sample application**
Get started right away with sample code.

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

Cancel
Configure more options
Create application

4. Presets

Elastic Beanstalk > Getting started

Configure Test2-env

Presets

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

☐ Single instance (*Free Tier eligible*)
☐ Single instance (using Spot instance)
☒ High availability
☐ High availability (using Spot and On-Demand instances)
☐ Custom configuration

5. Database – edit

Database
Edit

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

Add database

Add an RDS SQL database integrated with your environment. [Learn more](#)

Restore a snapshot - optional

Restore an existing snapshot from a previously used database.

Snapshot

None



Database settings

Choose an engine and instance type for your environment's database.

Engine

mysql

Engine version

8.0.28

Instance class

db.t2.micro

Storage

Choose a number between 5 GB and 1024 GB.

5

Engine – mysql

Engine version - 5.7.31

Give username and password

Database deletion policy – Delete

Keep other configurations as it is

mysql

Engine version

5.7.31

Instance class

db.t2.micro

Storage

Choose a number between 5 GB and 1024 GB.

5

Username

root

Password

.....

Availability

Low (one AZ)

Database deletion policy

This policy applies when you decouple a database or terminate the environment coupled to it.

Create snapshot

Elastic Beanstalk saves a snapshot of the database and then deletes it. You can restore a database from a snapshot when you add a DB to an Elastic Beanstalk environment or when you create a standalone database. You might incur charges for storing database snapshots.

Retain

The decoupled database will remain available and operational external to Elastic Beanstalk.

Delete

Elastic Beanstalk terminates the database. The database will no longer be available.

Cancel

Save

Save

Create app

This will take some time to finish

6. Check on beanstalk dashboard

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

Elastic Beanstalk

Environments

Applications

Change history

Recent environments

Test-env

Elastic Beanstalk > Environments

Depreciated/Retired Platform(s)

One or more of your environments is configured with a deprecated or retired platform. [Learn more](#)

All environments

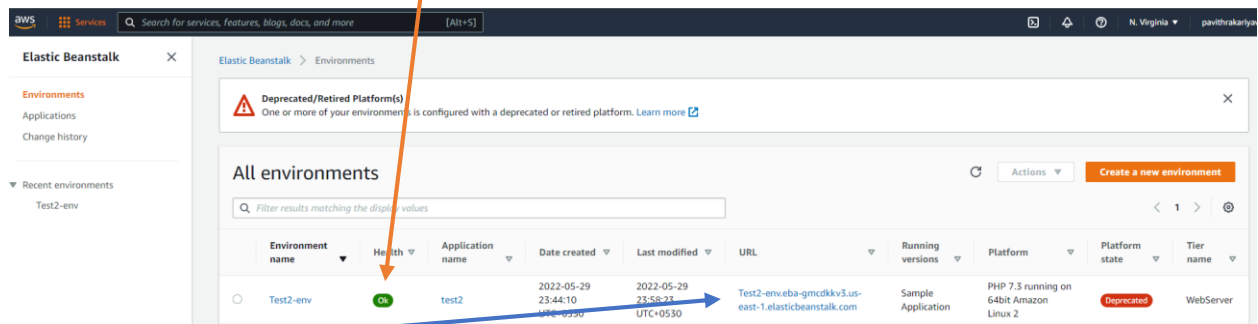
Filter results matching the display values

Actions

Create a new environment

Environment name	Health	Application name	Date created	Last modified	URL	Running versions	Platform	Platform state	Tier name
Test2-env	Pending	test2	2022-05-29 23:44:10 UTC+0530	2022-05-29 23:46:05 UTC+0530	Test2-env.eba-gmcdkvv3.us-east-1.elasticbeanstalk.com		PHP 7.3 running on 64bit Amazon Linux 2	Depreciated	WebServer

Once finished (health become ok – green color)

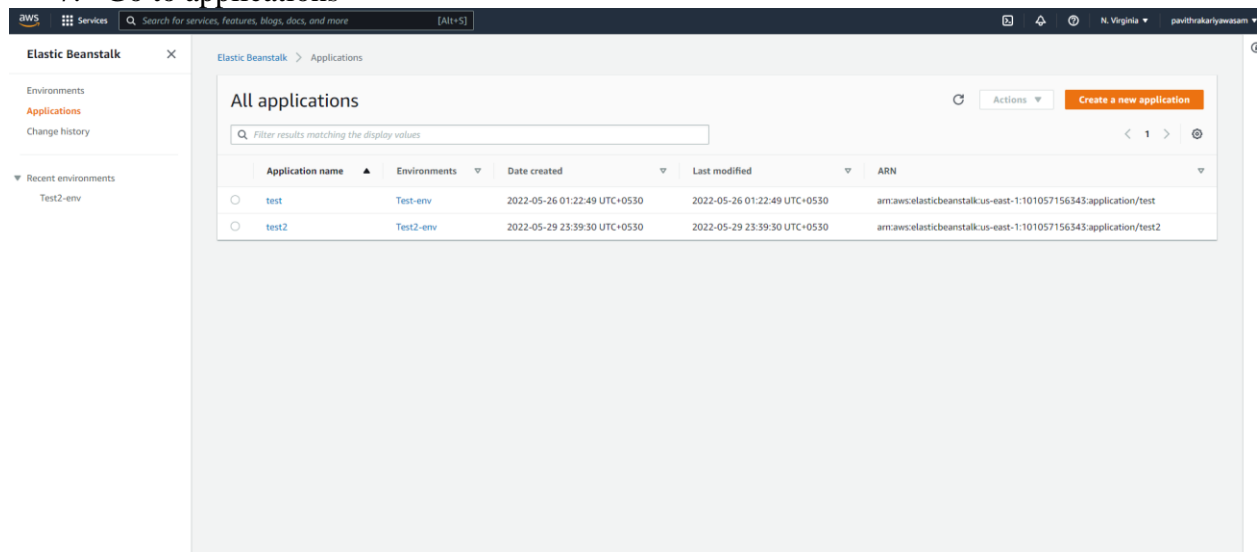


Click on the url of the environment, you should view web application like this



Go back

7. Go to applications



Click on name of the application you created

Go to application versions

Elastic Beanstalk ×

Environments

Applications

Change history

▼ test2

Application versions

Saved configurations

▼ Recent environments

Test2-env

Elastic Beanstalk ×

Environments

Applications

Change history

▼ test2

Application versions

Saved configurations

▼ Recent environments

Test2-env

Elastic Beanstalk > Applications > test2 > Application versions

Application versions

Actions ▼ Settings Upload ↻ Refresh

<input type="checkbox"/>	Version label	Description	Date created	Source	Deployed to
<input type="checkbox"/>	Sample Application		2022-05-29T23:44:07+05:30	Sample Application	Test2-env

8. Upload

Upload the source_codes.zip file here

Select latest version -> actions -> deploy

Elastic Beanstalk > Applications > test2 > Application versions

Info
Created application version Sample Application-1...

Application versions Actions ▾ Settings Upload Refresh

<input type="checkbox"/>	Version label	Description	Date created	Source	Deployed to
<input checked="" type="checkbox"/>	Sample Application-1		2022-05-30T00:19:43+05:30	2022149fPw-source_codes.zip	-
<input type="checkbox"/>	Sample Application		2022-05-29T23:44:07+05:30	Sample Application	Test2-env

Deploy Application Version

✕

Version label:

Sample Application-1

Environment:

Test2-env ▾

Environment URL:

Test2-env.eba-gmcdkvv3.us-east-1.elasticbeanstalk.com

Cancel Deploy

Wait few minutes.

Under deployed to section your environment name will appear

Elastic Beanstalk > Applications > test2 > Application versions

Info
The deployment to Test2-env started successfully. See the events page.

Application versions Actions ▾ Settings Upload Refresh

<input type="checkbox"/>	Version label	Description	Date created	Source	Deployed to
<input type="checkbox"/>	Sample Application-1		2022-05-30T00:19:43+05:30	2022149fPw-source_codes.zip	Test2-env
<input type="checkbox"/>	Sample Application		2022-05-29T23:44:07+05:30	Sample Application	-

You can also check with this

aws Services Search for services, features, blogs, docs, and more [Alt+S]

N. Virginia pavitrakalyan

Elastic Beanstalk

- Environments
- Applications
- Change history

test2

- Application versions
- Saved configurations

Test2-env

- Go to environment
- Configuration
- Logs
- Health
- Monitoring
- Alarms
- Managed updates
- Events
- Tags

Recent environments

- Test2-env

Deprecated platform

This environment uses a deprecated platform branch. We recommended that you upgrade to a supported platform branch. A deprecated branch may have a scheduled retirement date. It still receives ongoing maintenance updates. [Learn more](#)

Test2-env

Test2-env.aba-gmcdkkv3.us-east-1.elasticbeanstalk.com [in-guuh8m@bw5]

Application name: test2

Refresh Actions

Health

Ok

Causes

Running version

Sample Application-1

Upload and deploy

Platform

PHP 7.3 running on 64bit Amazon Linux 2/3.3.14

Deprecated

Change

Recent events

Show all

Time	Type	Details
2022-05-30 00:22:54 UTC+0530	INFO	Environment update completed successfully.
2022-05-30 00:22:54 UTC+0530	INFO	New application version was deployed to running EC2 instances.
2022-05-30 00:22:14 UTC+0530	INFO	Instance deployment completed successfully.

Always health should be green color

Elastic Beanstalk

Environments

Applications

Change history

▼ test2

Application versions

Saved configurations

▼ Test2-env

Go to environment

Configuration

Logs

Health

Monitoring

Alarms

Managed updates

Events

Tags

▼ Recent environments

Test2-env

Go to environment

Ride sharing

[Register As a Driver](#)
[Register As a Passenger](#)

Username

Password

Submit

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

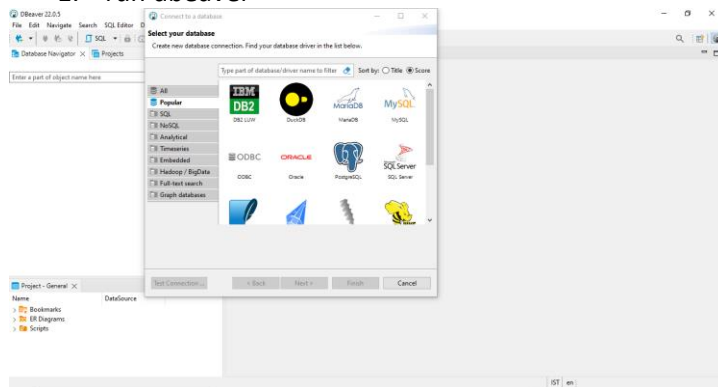
RDS

Get dbeaver (you can use any database tool)

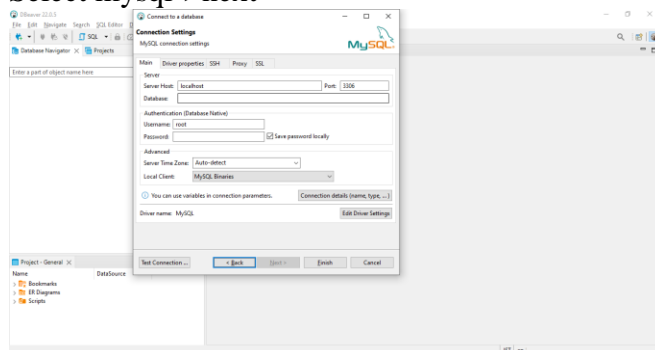
<https://dbeaver.io/download/>

install dbeaver

1. run dbeaver

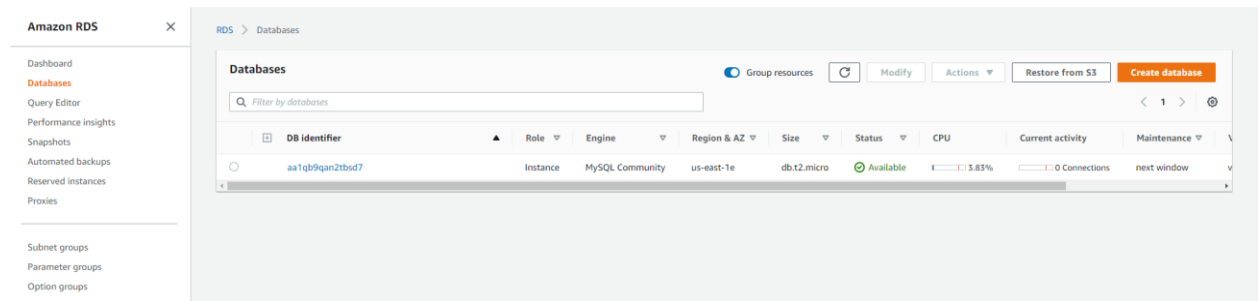


Select mysql->next

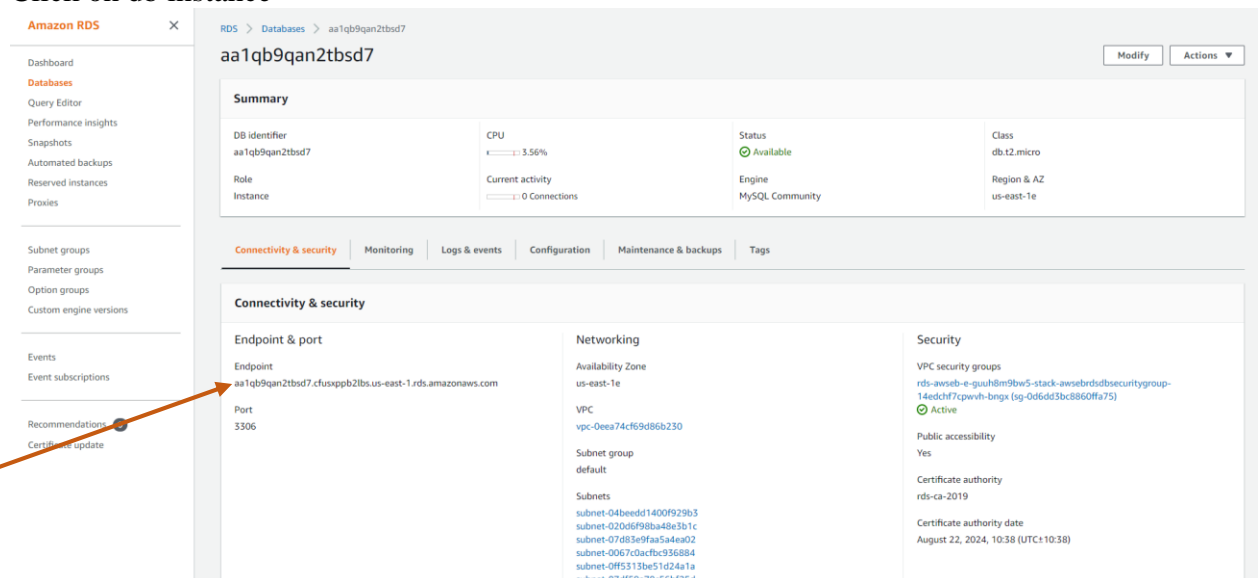


2. go to RDS dashboard

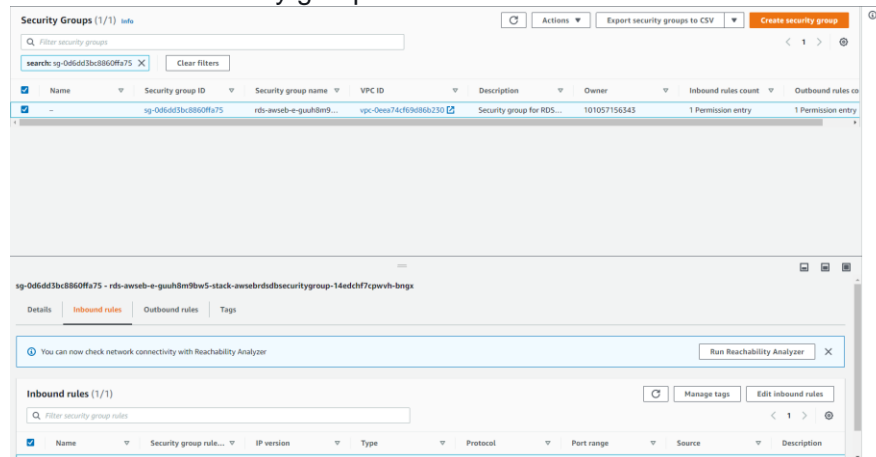
<https://us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#databases:>



Click on db instance



Click on VPC security groups



Edit inbound rules -> add rule
Type – all traffic

Source – Anywhere-IPV4

The screenshot shows the 'Edit inbound rules' page in the AWS IAM console. The breadcrumb trail is: EC2 > Security Groups > sg-0d6dd3bc860ffa75 - rds-awseb-e-guuh8m9bw5-stack-awsebrdssecuritygroup-14edch7cpevh-bngx > Edit inbound rules. The page title is 'Edit inbound rules' with an 'Info' link. Below the title is a note: 'Inbound rules control the incoming traffic that's allowed to reach the instance.' The main section is 'Inbound rules' with an 'Info' link. It contains a table with two rows. The first row has a 'Security group rule ID' of 'sgr-0b343124fe6f9d853', 'Type' of 'MySQL/Aurora', 'Protocol' of 'TCP', 'Port range' of '3306', 'Source' of 'Custom', and a 'Description - optional' field. The second row has a 'Security group rule ID' of '-', 'Type' of 'All traffic', 'Protocol' of 'All', 'Port range' of 'All', 'Source' of 'Anywhere-I...', and a 'Description - optional' field. There are 'Add rule', 'Delete', 'Cancel', 'Preview changes', and 'Save rules' buttons.

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-0b343124fe6f9d853	MySQL/Aurora	TCP	3306	Custom	
-	All traffic	All	All	Anywhere-I...	

Save

Copy endpoint and go back to dbeaver

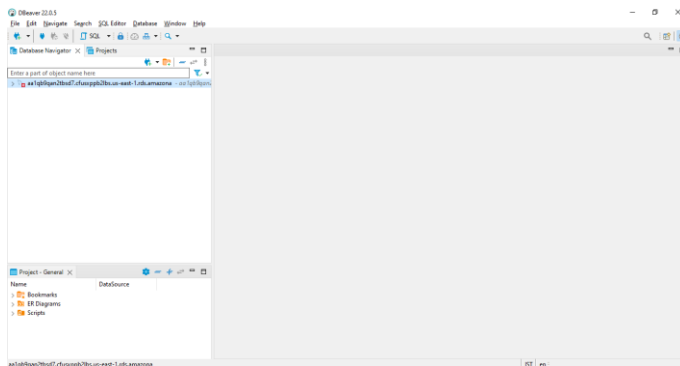
The screenshot shows the 'Connect to a database' dialog box in DBeaver. The title bar says 'Connect to a database'. The main section is 'Connection Settings' with the MySQL logo. There are tabs for 'Main', 'Driver properties', 'SSH', 'Proxy', and 'SSL'. The 'Main' tab is selected. It contains fields for 'Server Host' (qan2tbsd7.cfusxppb2lbs.us-east-1.rds.amazonaws.com), 'Port' (3306), 'Database', 'Username' (root), 'Password' (masked), and 'Save password locally' (checked). There are also 'Advanced' settings for 'Server Time Zone' (Auto-detect) and 'Local Client' (MySQL Binaries). At the bottom, there are buttons for 'Test Connection ...', '< Back', 'Next >', 'Finish', and 'Cancel'.

Server host – endpoint copied from rds

Username – assigned by you when creating RDS in beanstalk (step 5)

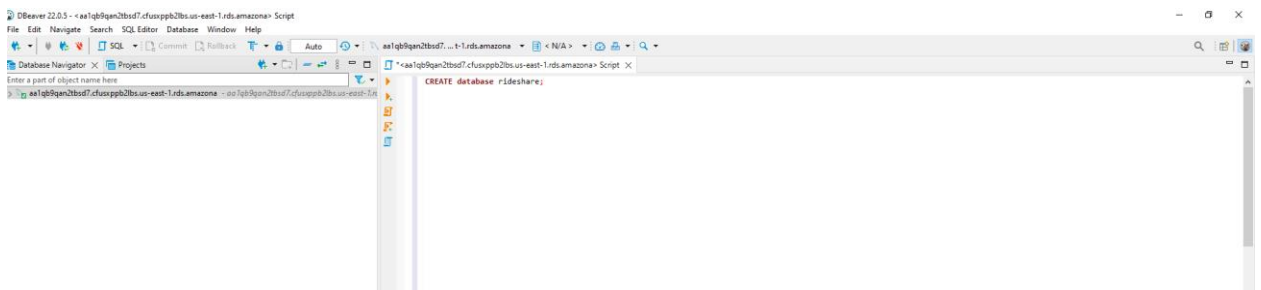
Password - assigned by you when creating RDS in beanstalk (step 5)

Finished

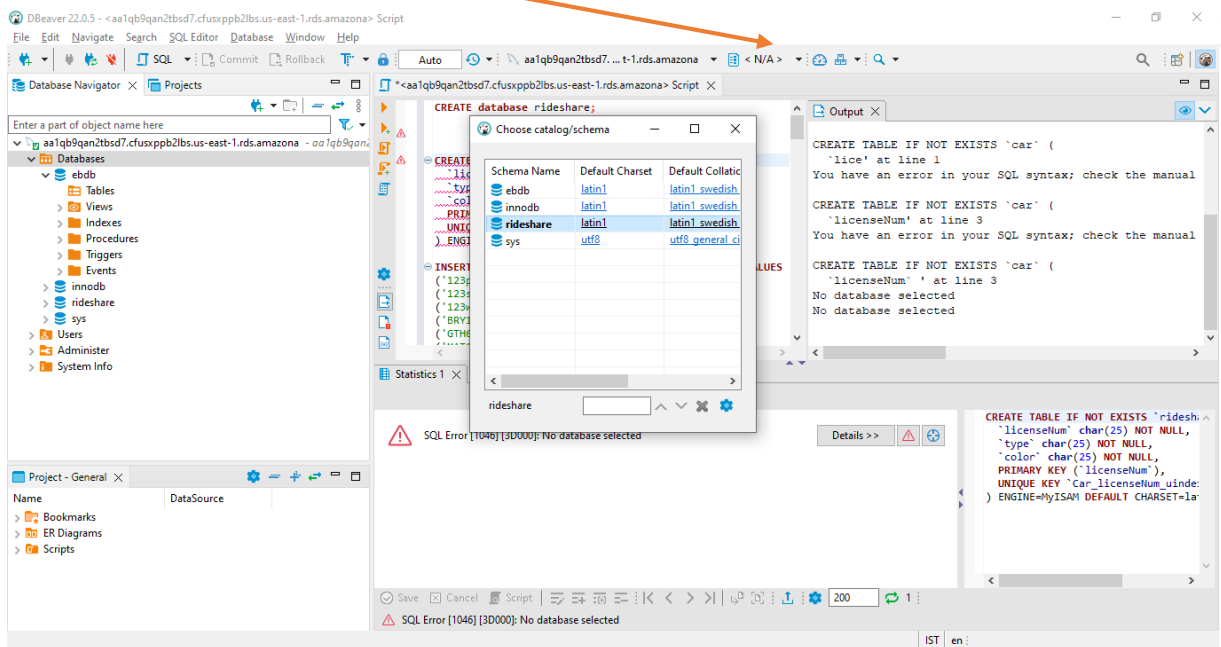


Right click on connection -> connect

- 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

The screenshot shows the Amazon CloudFront landing page in the AWS console. The page is titled "Amazon CloudFront" and "Securely deliver content with low latency and high transfer speeds". It features a "Get started with CloudFront" section with a "Create a CloudFront distribution" button. Below this, there are sections for "Benefits and features" (Reduce latency, Improve security, Cut costs, Customize delivery) and "Pricing (US)".

Get started with CloudFront

Enable accelerated, reliable and secure content delivery for Amazon S3 buckets, Application Load Balancers, Amazon API Gateway APIs, and more in 5 minutes or less.

[Create a CloudFront distribution](#)

Benefits and features

- Reduce latency**
The CloudFront network has 225+ points of presence (PoP) that are connected by fully redundant, parallel 100 GbE fiber delivering ultra-low latency performance and high availability to your end users. CloudFront automatically maps network conditions and intelligently routes your user's traffic when serving up cached or dynamic content.
- Improve security**
Use CloudFront for perimeter protection, traffic encryption, and access controls. AWS Shield Standard defends traffic transmitted through CloudFront from DDoS attacks at no additional charge. For application protection, you can integrate AWS WAF, managed rules, and managed third-party firewall options into CloudFront workloads.
- Cut costs**
Integrated with AWS, using CloudFront consolidates requests and removes data transfer out fees from your AWS origins. CloudFront offers customizable pricing options including simple pay-as-you-go pricing with no upfront fees and the CloudFront Security Savings Bundle that helps save up to an additional 30%.
- Customize delivery**
Serverless compute features enable you to securely run your own code at the AWS CDN edge. Customize your delivery to overcome the unique challenges your business faces, creating your own balance between cost, performance, and security.

Pricing (US)

Feature	Price
First 1 TB data transfer free each month	
10 TB/month	\$0.085 per GB
HTTP requests	\$0.0075 per 10,000

1. create distribution

select origin domain which url start with elasticbeanstalk....

The screenshot shows the "Create distribution" form in the AWS console. The form is titled "Create distribution" and has a "Origin" section. The "Origin domain" field is populated with "elasticbeanstalk-us-east-1-101057156343.s3.us-east-1.amazonaws.com". The "Origin path" field is empty. The "Name" field is populated with "elasticbeanstalk-us-east-1-101057156343.s3.us-east-1.amazonaws.com". The "S3 bucket access" section has two radio buttons: "Don't use OAI (bucket must allow public access)" (selected) and "Yes use OAI (bucket can restrict access to only CloudFront)". The "Add custom header" section has an "Add header" button. The "Enable Origin Shield" section has two radio buttons: "No" (selected) and "Yes".

Create distribution

Origin

Origin domain
Choose an AWS origin, or enter your origin's domain name.

Origin path - optional [Info](#)
Enter a URL path to append to the origin domain name for origin requests.

Name
Enter a name for this origin.

S3 bucket access [Info](#)
Use a CloudFront origin access identity (OAI) to access the S3 bucket.

☒ Don't use OAI (bucket must allow public access)
☐ Yes use OAI (bucket can restrict access to only CloudFront)

Add custom header - optional
CloudFront includes this header in all requests that it sends to your origin.

Enable Origin Shield [Info](#)
Origin Shield is an additional caching layer that can help reduce the load on your origin and help protect its availability.

☒ No
☐ Yes

[Additional settings](#)

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

Default cache behavior

Path pattern [Info](#)

Default (*)

Compress objects automatically [Info](#)

☐ No

☒ Yes

Viewer

Viewer protocol policy

☒ HTTP and HTTPS

☐ Redirect HTTP to HTTPS

☐ HTTPS only

Allowed HTTP methods

☐ GET, HEAD

☐ GET, HEAD, OPTIONS

☒ GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE

Cache HTTP methods

GET and HEAD methods are cached by default.

☐ OPTIONS

Restrict viewer access

If you restrict viewer access, viewers must use CloudFront signed URLs or signed cookies to access your content.

☒ No

☐ Yes

Cache key and origin requests

We recommend using a cache policy and origin request policy to control the cache key and origin requests.

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

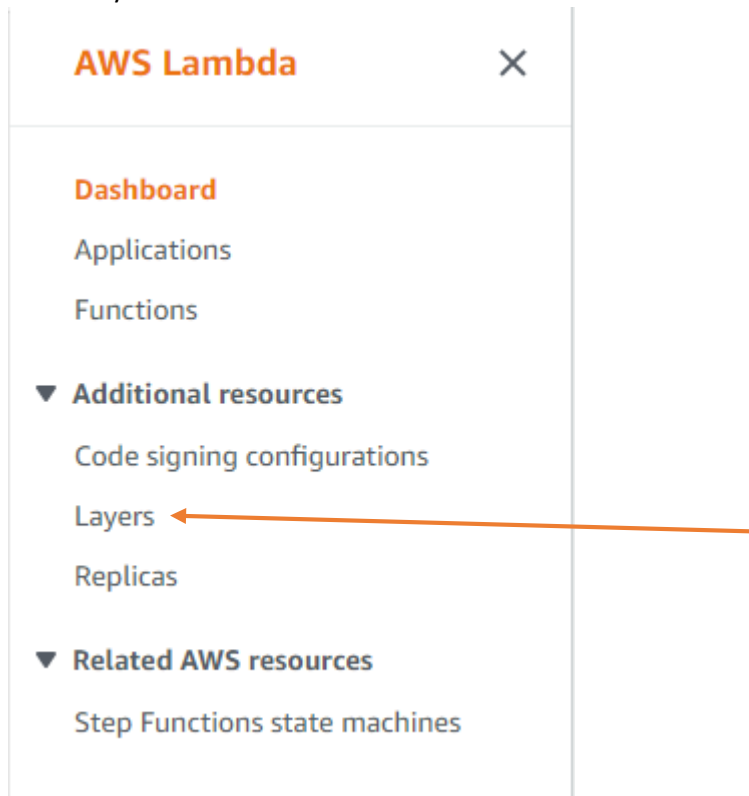
within folder your source code zip file will be available.

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

LAMBDA

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

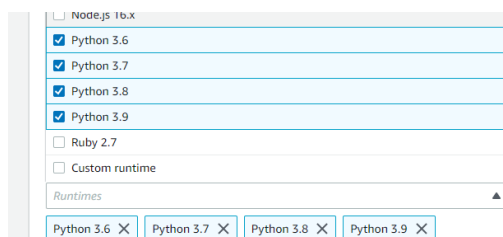
1. create layer



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 X

Python 3.7 X

Python 3.8 X

Python 3.9 X

License - *optional* [Info](#)

Cancel

Create

Create

Lambda > Layers > pymysql

ARN: arn:aws:lambda:us-east-1:101057156343:layer:pymysql:1

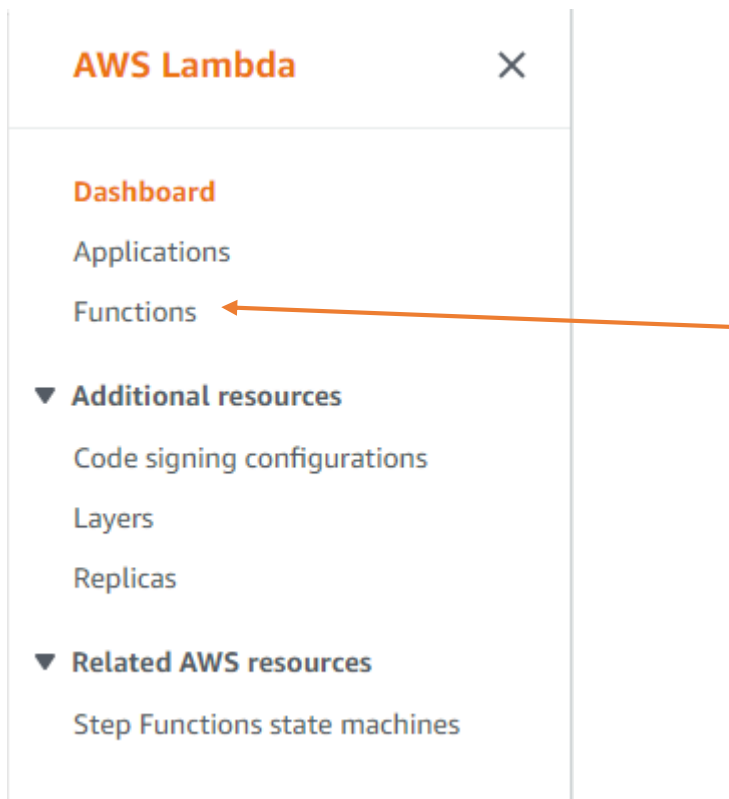
pymysql [Delete](#) [Download](#) [Create version](#)

Successfully created layer pymysql version 1.

Version details		
Version 1	Description -	Created 2 seconds ago
License -	Compatible runtimes python3.6, python3.7, python3.8, python3.9	Compatible architectures -

Versions Functions using this version

All versions		
Version	Version ARN	Description
1	arn:aws:lambda:us-east-1:101057156343:layer:pymysql:1	-



2. Create function

Function name - Any function name

Runtime – python 3.7

Create function [Info](#)

Choose one of the following options to create your function.

Author from scratch ☒
Start with a simple Hello World example.

Use a blueprint ☐
Build a Lambda application from sample code and configuration presets for common use cases.

Container image ☐
Select a container image to deploy for your function.

Browse serverless app repository ☐
Deploy a sample Lambda application from the AWS Serverless Application Repository.

Basic information

Function name
Enter a name that describes the purpose of your function.

Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime [Info](#)
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

Architecture [Info](#)
Choose the instruction set architecture you want for your function code.
☒ x86_64
☐ arm64

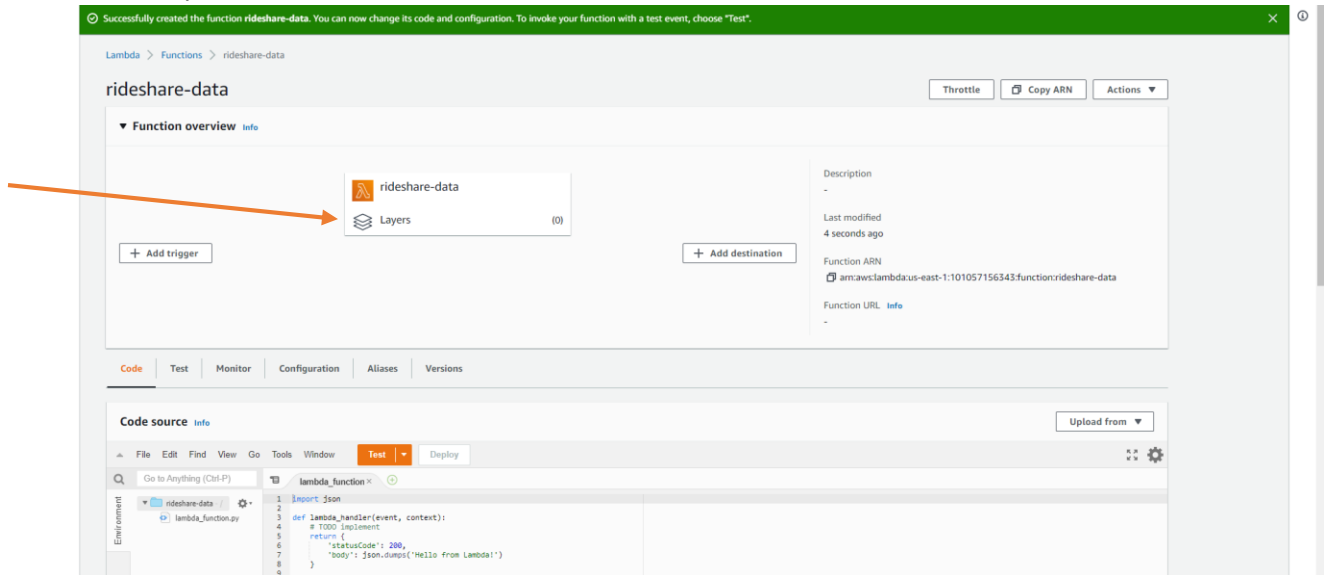
Permissions [Info](#)
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.
[Change default execution role](#)

[Advanced settings](#)

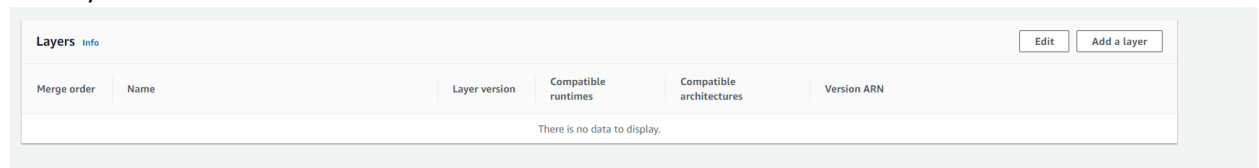
[Cancel](#) [Create function](#)

Create function

3. Attach layer



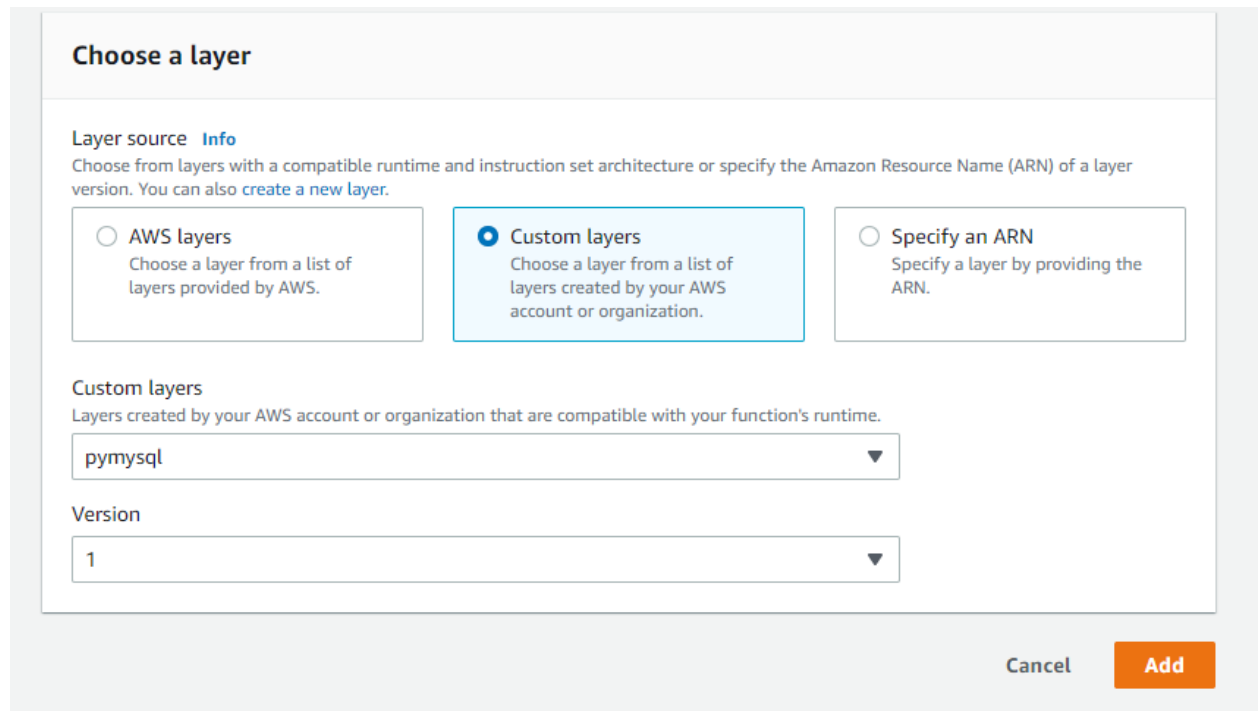
Add layer



Custom layers

Select the layer name you create previously

Select latest version

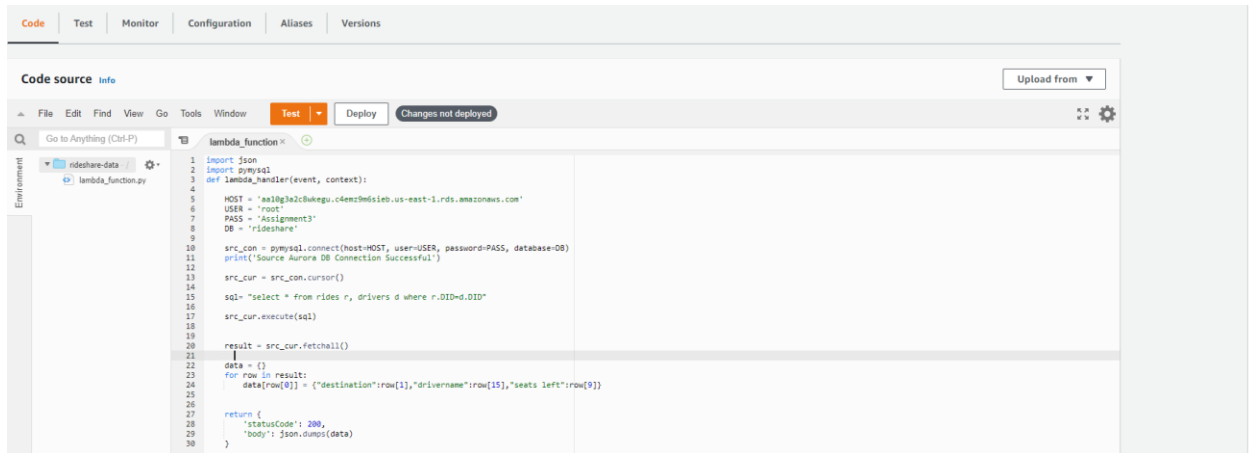


add

4. Add code

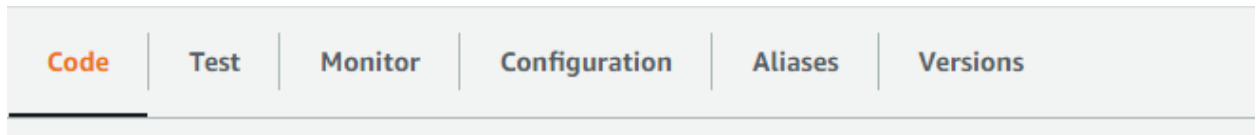
Delete existing code and paste following(lambda.py code)

Change database credentials (host, user, password, db)(RDS credentials)

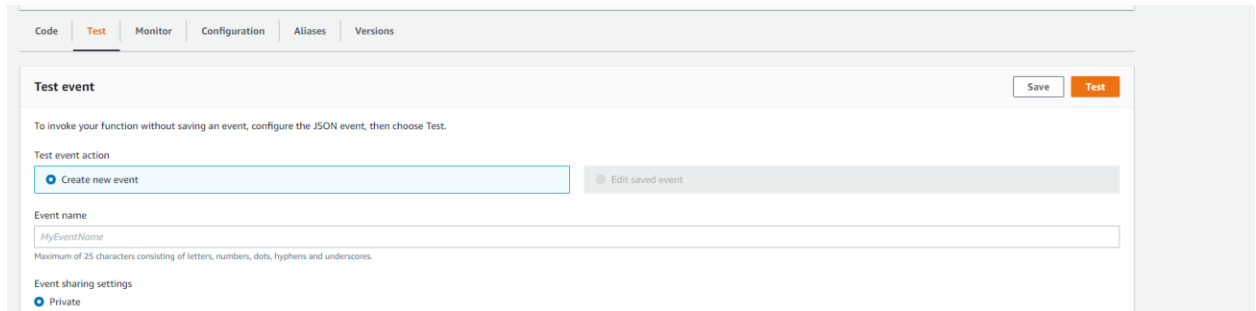


Deploy

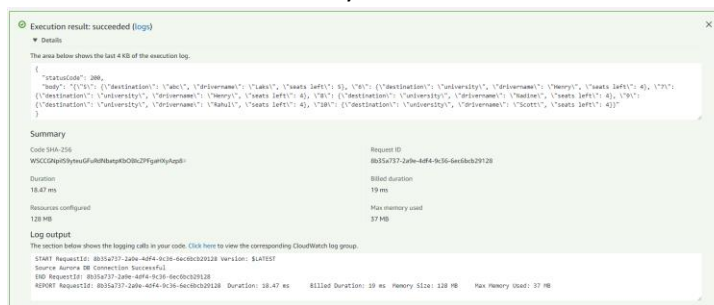
5. Go to test



test



Function should run correctly

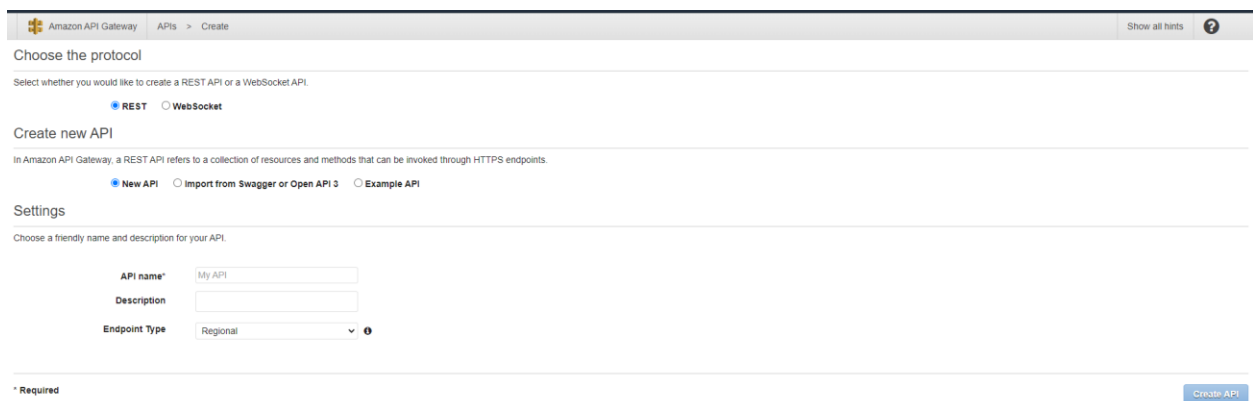
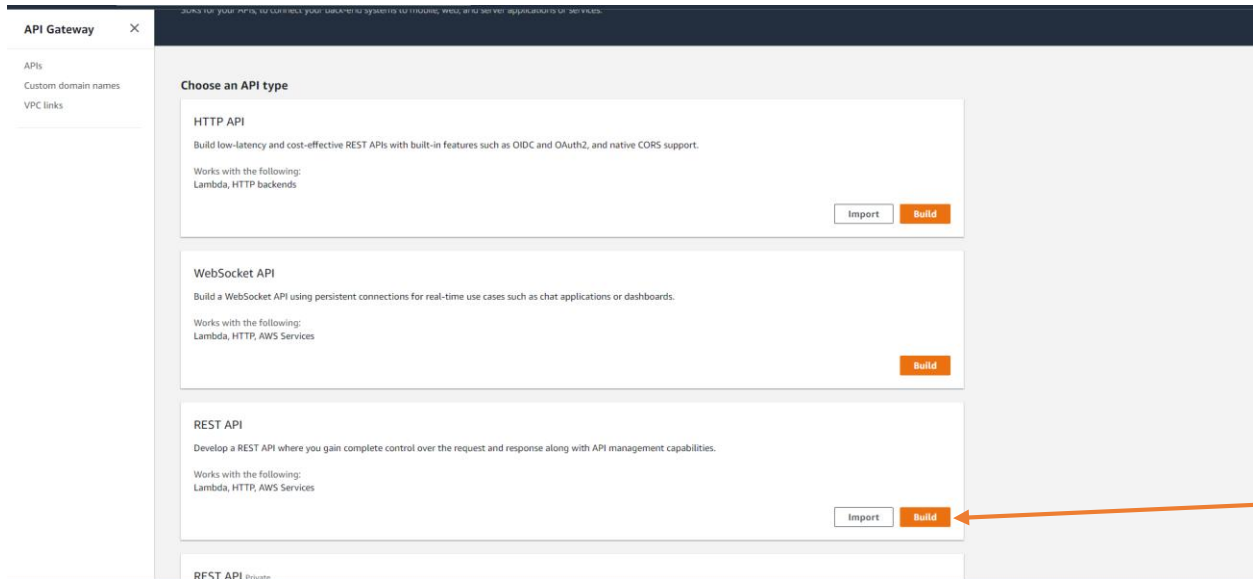


Similar output should come

API Gateway

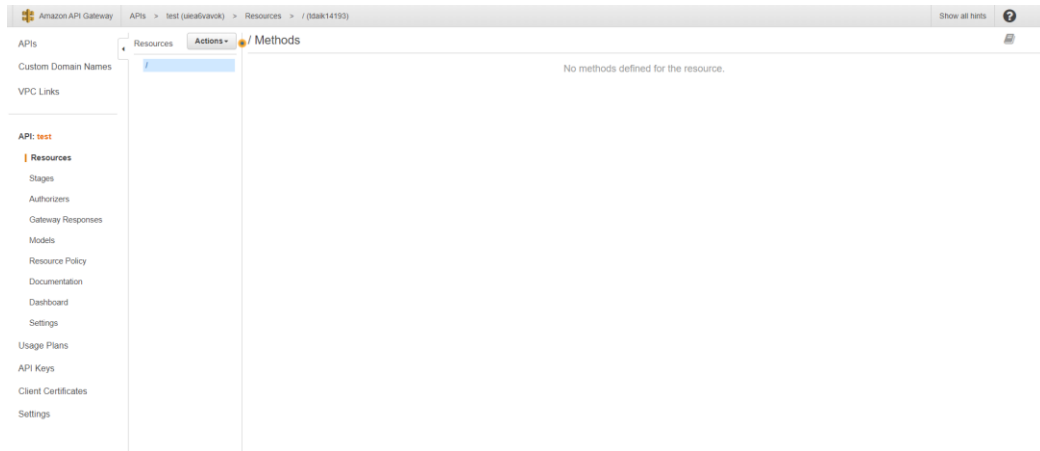
<https://us-east-1.console.aws.amazon.com/apigateway>

1. Build – REST API



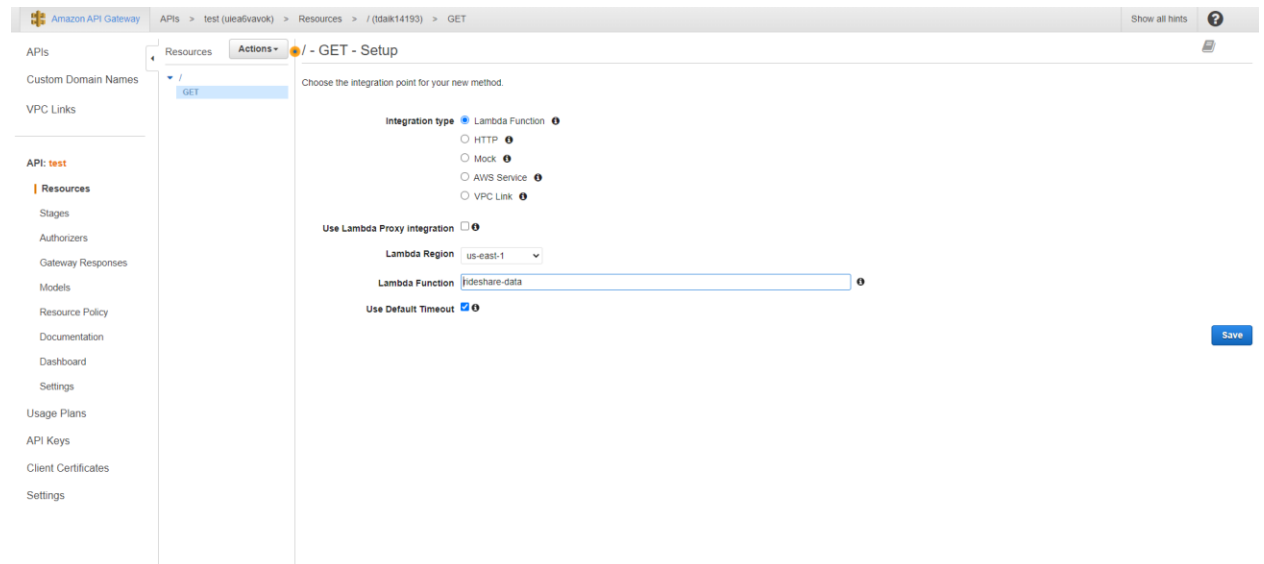
Give name to the API

Create



Actions -> Create Method
Select method type as GET

Type name of the lambda function you created in previously



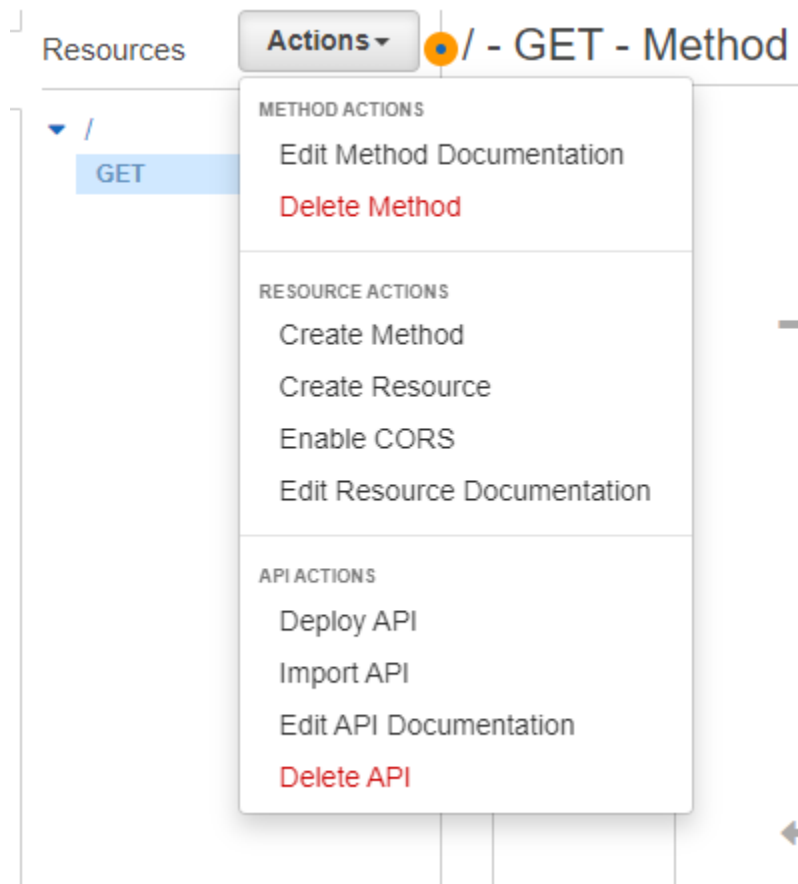
Save

Actions -> Enable CORS

The screenshot shows the AWS API Gateway console. The breadcrumb trail is: APIs > rides (7t1uvugno9) > Resources > / (z2zjsqxz52) > GET. The left sidebar shows the API 'rides' and its 'Resources' section. The 'Actions' dropdown menu is open, showing 'METHOD ACTIONS', 'RESOURCE ACTIONS', and 'API ACTIONS'. The 'Enable CORS' option is highlighted under 'RESOURCE ACTIONS'. An orange arrow points from this option to the 'Integration Request' box in the diagram. The diagram shows a 'Method Request' box with 'Auth: NONE' and 'ARN: arn:aws:execute-api:us-east-1:1314090525186:7t1uvugno9/*/*/GET/'. It connects to an 'Integration Request' box with 'Type: LAMBDA' and 'Region: us-east-1'. The 'Integration Request' box connects to an 'Integration Response' box with 'HTTP status pattern: -' and 'Output passthrough: Yes'. The 'Integration Response' box connects to a 'Method Response' box with 'HTTP Status: 200' and 'Models: application/json => Empty'. The 'Method Response' box connects back to the 'Method Request' box. A vertical box on the right is labeled 'Lambda rides-to-public'.

The screenshot shows the 'Enable CORS' configuration page in the AWS API Gateway console. The breadcrumb trail is: APIs > rides (7t1uvugno9) > Resources > / (z2zjsqxz52) > Enable CORS. The left sidebar shows the API 'rides' and its 'Resources' section. The 'Actions' dropdown menu is open, showing 'Enable CORS'. The 'Enable CORS' configuration page is displayed. It includes the following settings: 'Gateway Responses for rides API' with checkboxes for 'DEFAULT 4XX' and 'DEFAULT 5XX'; 'Methods' with checkboxes for 'GET' and 'OPTIONS'; 'Access-Control-Allow-Methods' set to 'GET, OPTIONS'; 'Access-Control-Allow-Headers' set to 'Content-Type,X-Amz-Date,Authorization'; and 'Access-Control-Allow-Origin*' set to '*'. There is an 'Advanced' section below. A blue button at the bottom right says 'Enable CORS and replace existing CORS headers'.

Action->Deploy API



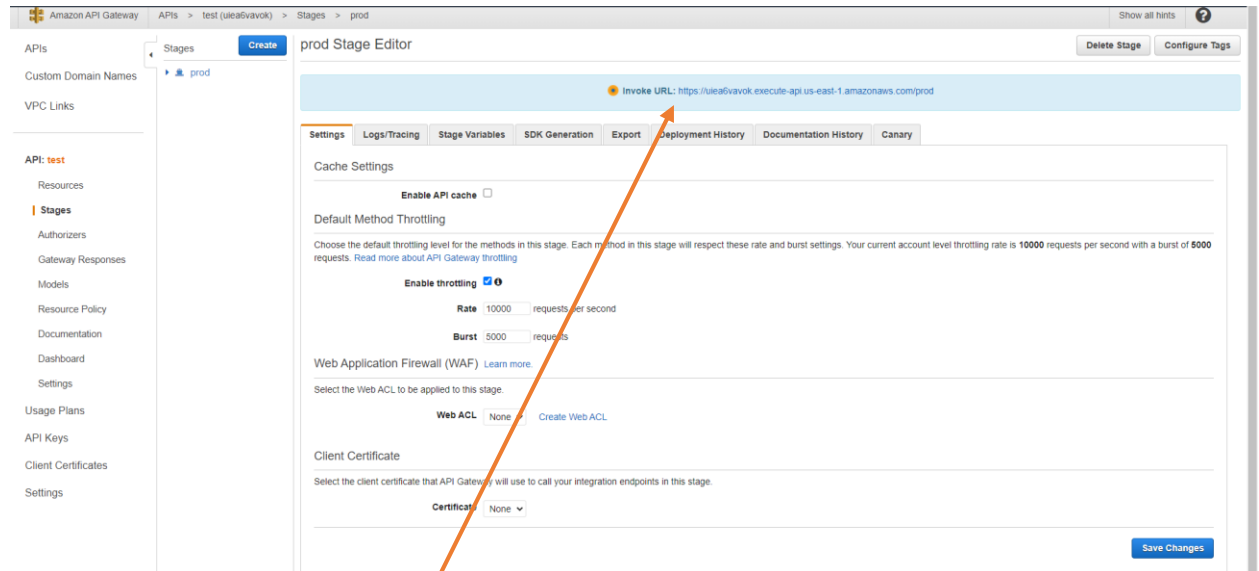
Development stage - select as new stage

Stage new – prod (any name stage name you prefer)

The 'Deploy API' dialog box is shown. It contains a message: 'Choose a stage where your API will be deployed. For example, a test version of your API could be deployed to a stage named beta.' Below this, there are four input fields: 'Deployment stage' (a dropdown menu with '[New Stage]' selected), 'Stage name*' (a text input field with 'prod' entered), 'Stage description' (a text input field), and 'Deployment description' (a text input field). At the bottom right, there are two buttons: 'Cancel' and 'Deploy'.

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

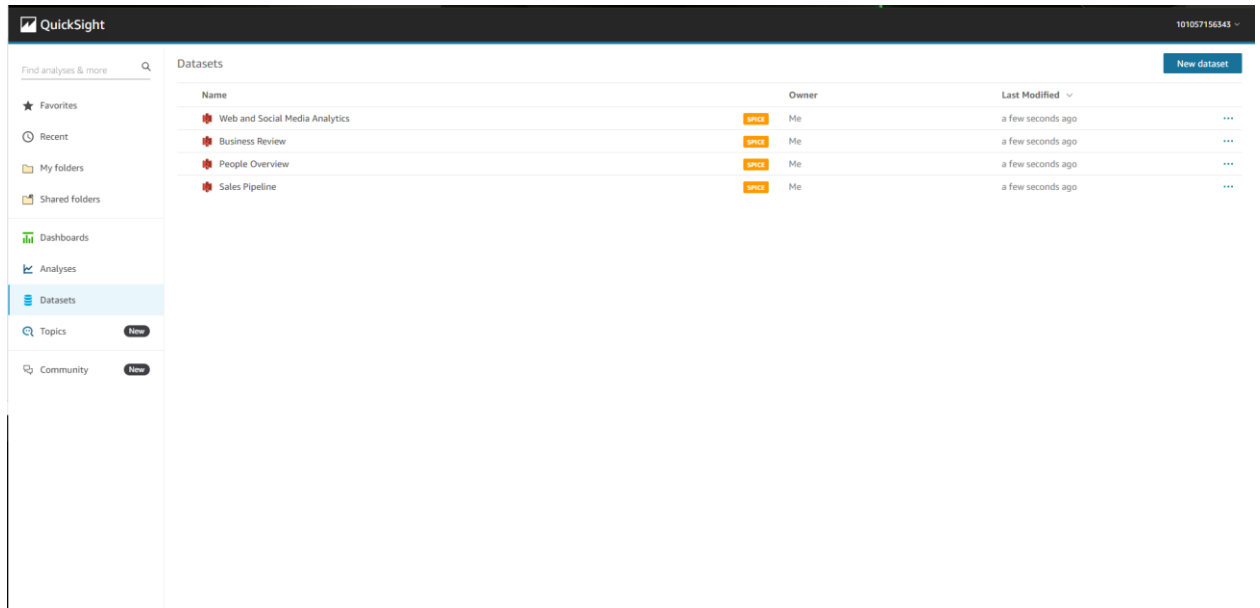
Zip all php files

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

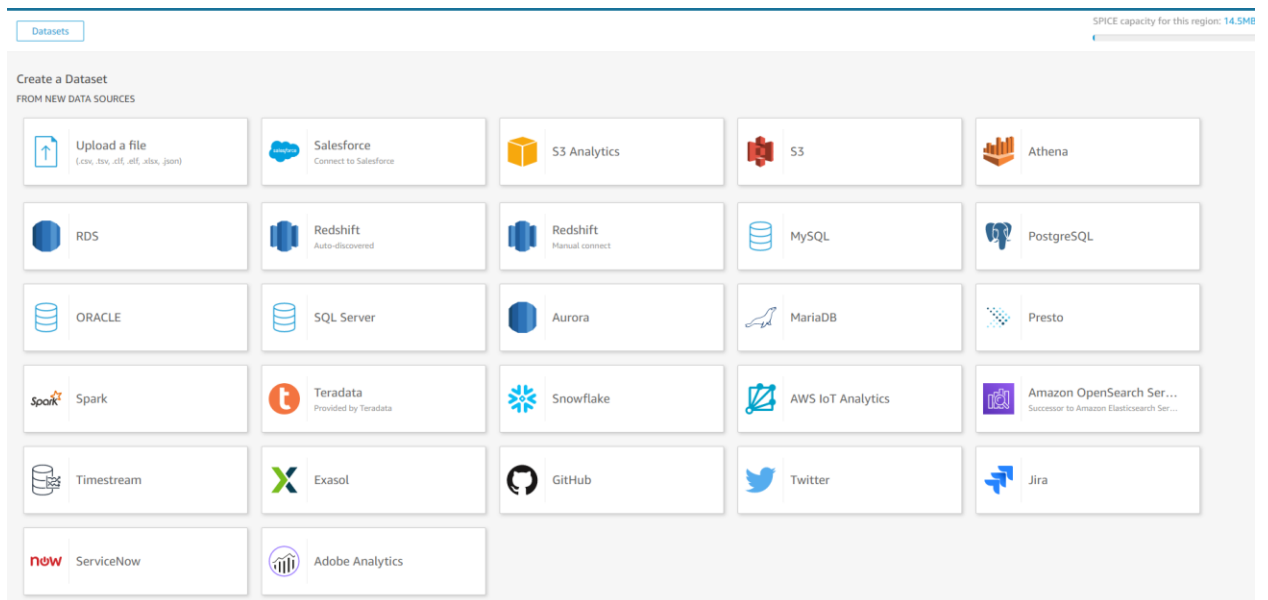
Quicksight

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

1. create dataset



New dataset



Select RDS

New RDS data source

Data source name

rideshare driver data

Instance ID

aa1qb9qan2tbsd7

Connection type

Public network

Database name

rideshare

Username

root

Password

.....

Validate connection

SSL is enabled

Create data source

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

Choose your table



rideshare driver data

Tables: contain the data you can visualize.

- ☐ car
- ☐ drivers
- ☐ locations
- ☐ participates
- ☐ passengers
- ☒ rides

Edit/Preview data

Use custom SQL

Select

Select table you want
Select

Finish dataset creation

Table:rides

Estimated table size:428 bytes

Data source:rideshare driver data

☐ Import to SPICE for quicker analytics

✓ 1009.5MB available

SPICE

☒ Directly query your data

Edit/Preview data


Augment with SageMaker


Visualize


Select direct query your data
Visualize


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


Visual types















































































































































































































































































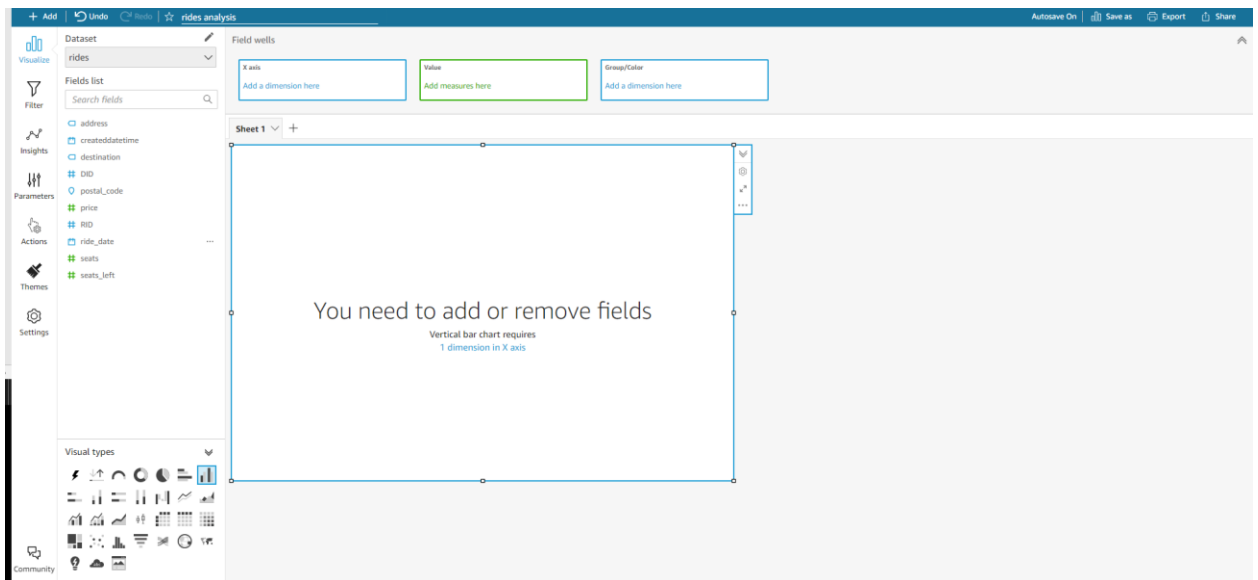




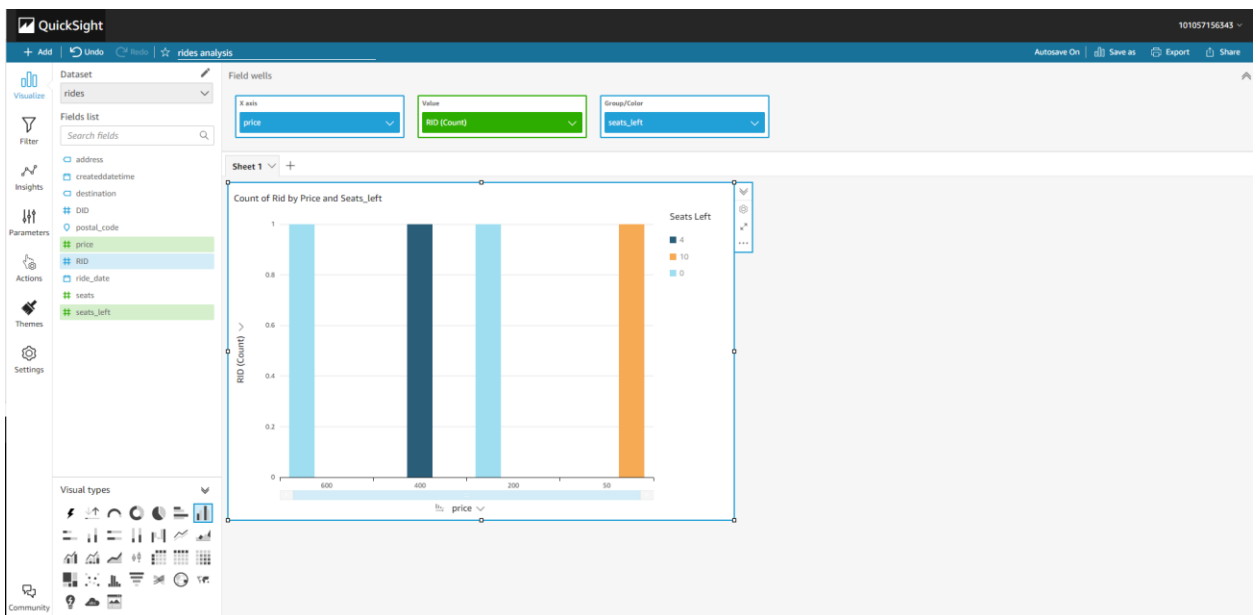








Select files for x axis, value (these changes according to visual selected)




Can get these type of charts

Share -> publish dashboard

Publish a dashboard ×

☒ Publish new dashboard as

☐ Replace an existing dashboard

Advanced publish options 

Publish dashboard

Give name you prefer
Publish dashboard