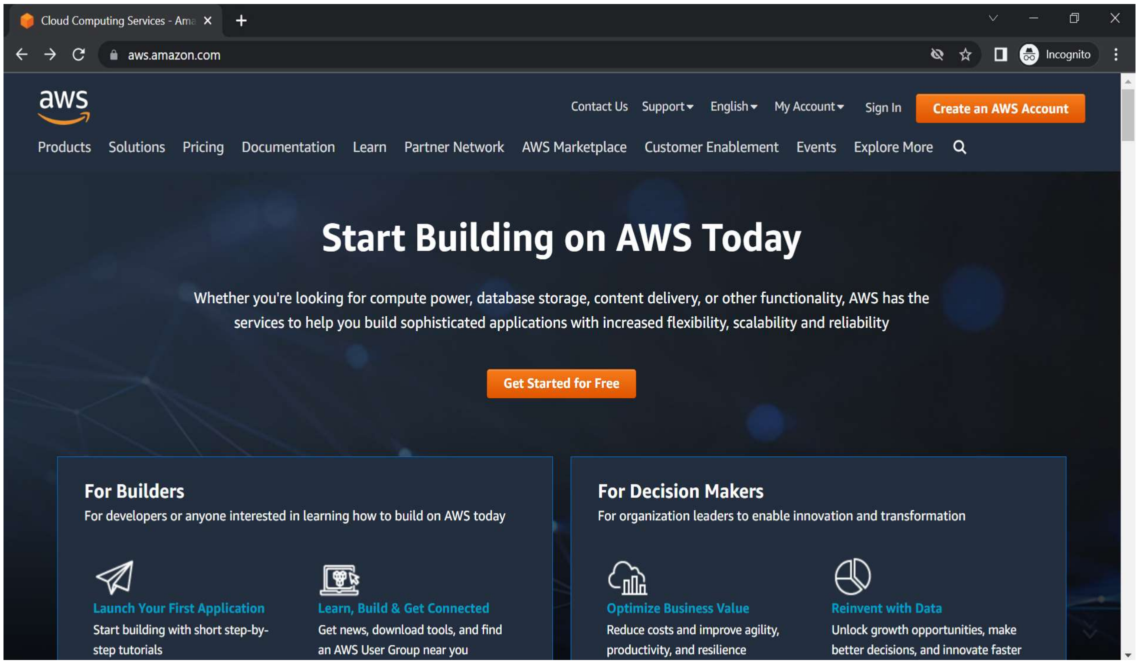
AWS Account Setup

URL - https://aws.amazon.com/



Start Building on AWS Today

Whether you're looking for compute power, database storage, content delivery, or other functionality, AWS has the services to help you build sophisticated applications with increased flexibility, scalability and reliability

For Builders

For developers or anyone interested in learning how to build on AWS today

Start building with short step-by-step tutorials

Get news, download tools, and find an AWS User Group near you

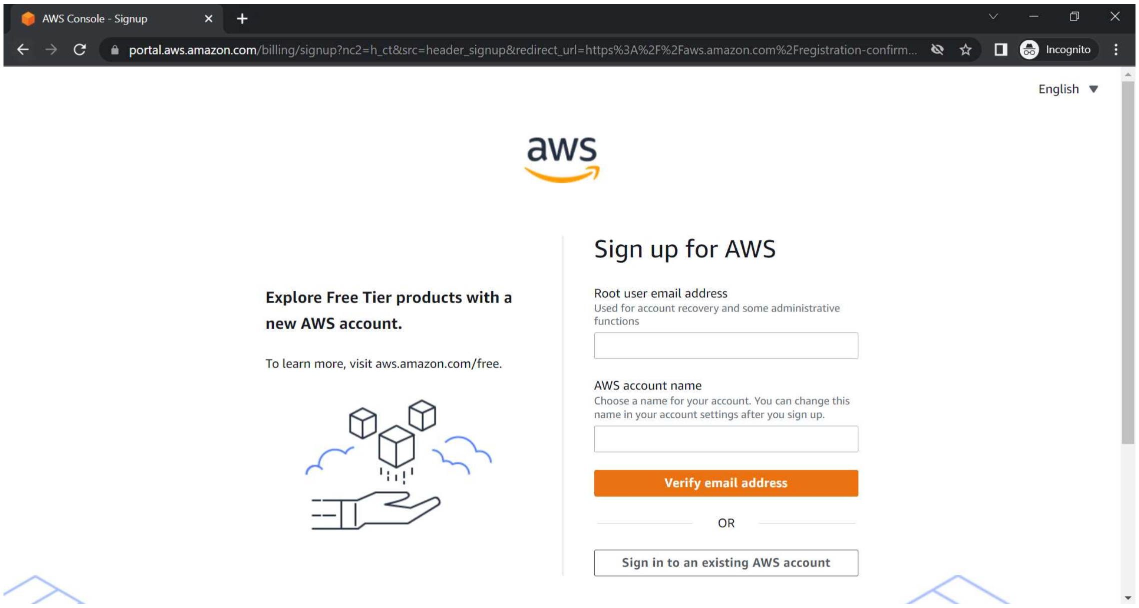
For Decision Makers

For organization leaders to enable innovation and transformation

Reduce costs and improve agility, productivity, and resilience

Unlock growth opportunities, make better decisions, and innovate faster

1) Click on Create an AWS Account button


AWS Console - Signup
portal.aws.amazon.com/billing/signup?nc2=h_ct&src=header_signup&redirect_url=https%3A%2F%2Faws.amazon.com%2Fregistration-confirm...
Incognito
English
aws
Verify email address
OR
Sign in to an existing AWS account

Explore Free Tier products with a new AWS account.

To learn more, visit aws.amazon.com/free.

Sign up for AWS

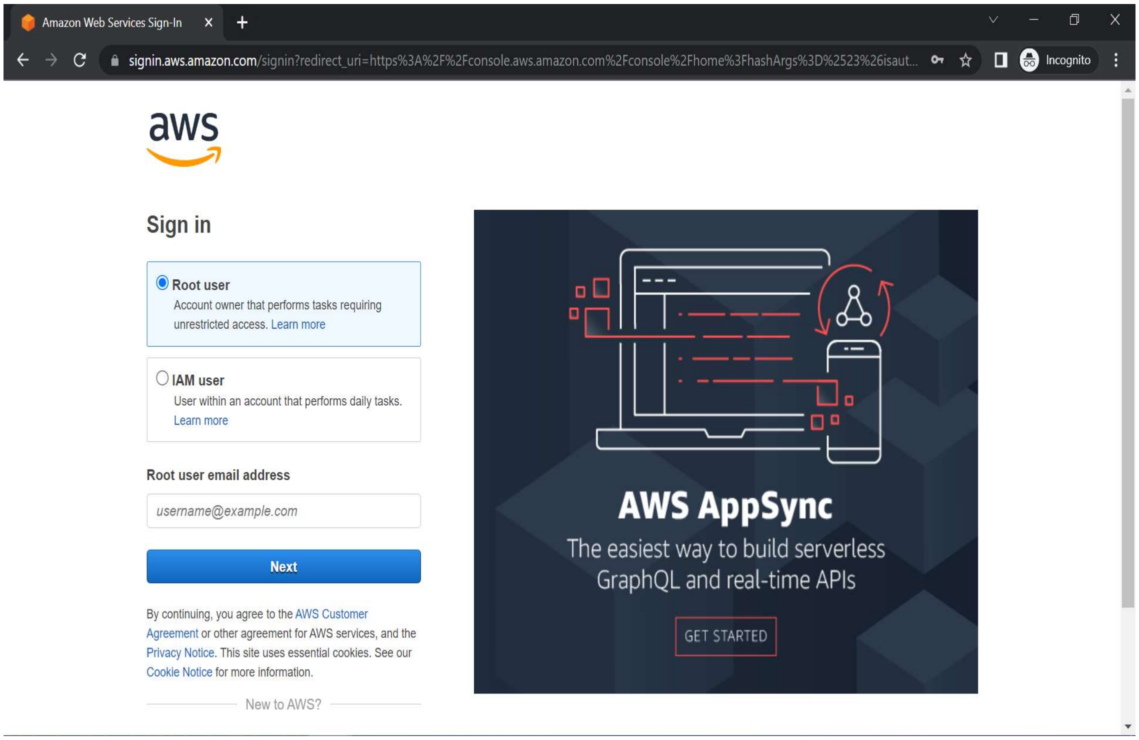
Root user email address

Used for account recovery and some administrative functions

AWS account name

Choose a name for your account. You can change this name in your account settings after you sign up.

2) Sign up new account with you email ID or If you have already an account in AWS, Click on Sign in to an existing AWS account



Amazon Web Services Sign-In

signin.aws.amazon.com/signin?redirect_uri=https%3A%2F%2Fconsole.aws.amazon.com%2Fconsole%2Fhome%3FhashArgs%3D%2523%26isaut...

Sign in

● Root user
Account owner that performs tasks requiring unrestricted access. Learn more

○ IAM user
User within an account that performs daily tasks. Learn more

Root user email address

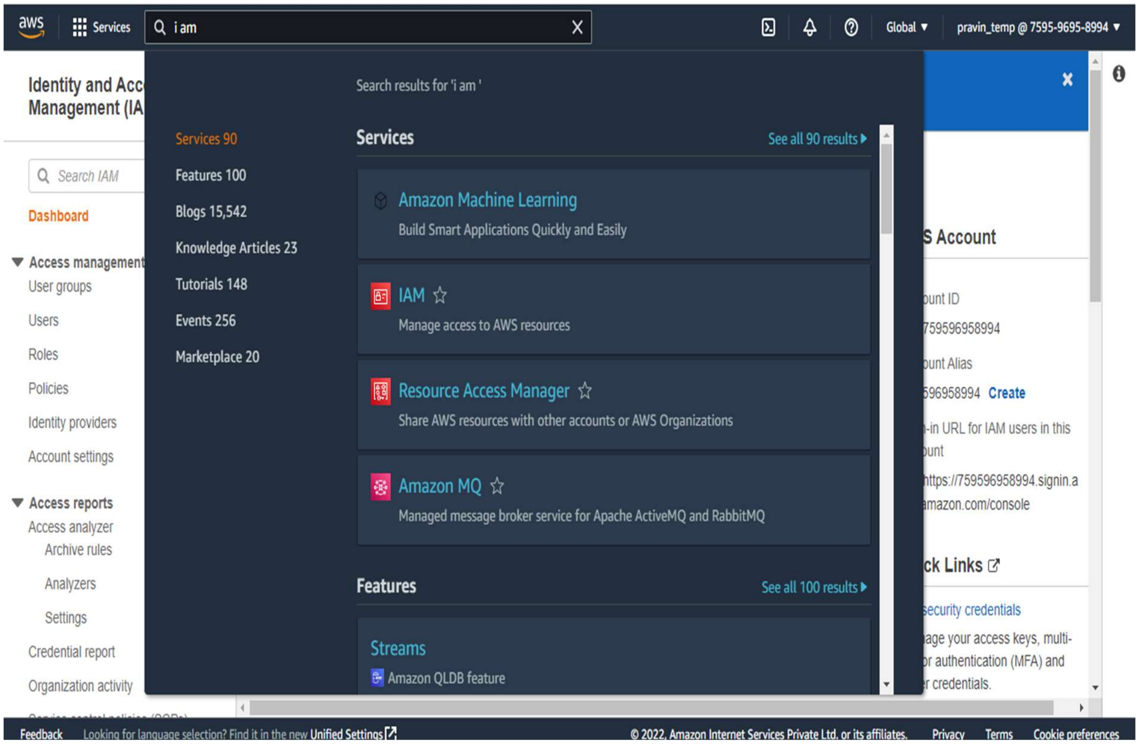
username@example.com

By continuing, you agree to the AWS Customer Agreement or other agreement for AWS services, and the Privacy Notice. This site uses essential cookies. See our Cookie Notice for more information.

New to AWS?

AWS AppSync
The easiest way to build serverless GraphQL and real-time APIs
GET STARTED

3) After Login



Q i am

Identity and Acc
Management (IA

Search IAM

Dashboard

▼ Access management
User groups
Users
Roles
Policies
Identity providers
Account settings

▼ Access reports
Access analyzer
Archive rules
Analyzers
Settings
Credential report
Organization activity

Search results for 'i am '

Services 90
Features 100
Blogs 15,542
Knowledge Articles 23
Tutorials 148
Events 256
Marketplace 20

Services — See all 90 results ▸

Amazon Machine Learning
Build Smart Applications Quickly and Easily

IAM ☆
Manage access to AWS resources

Resource Access Manager ☆
Share AWS resources with other accounts or AWS Organizations

Amazon MQ ☆
Managed message broker service for Apache ActiveMQ and RabbitMQ

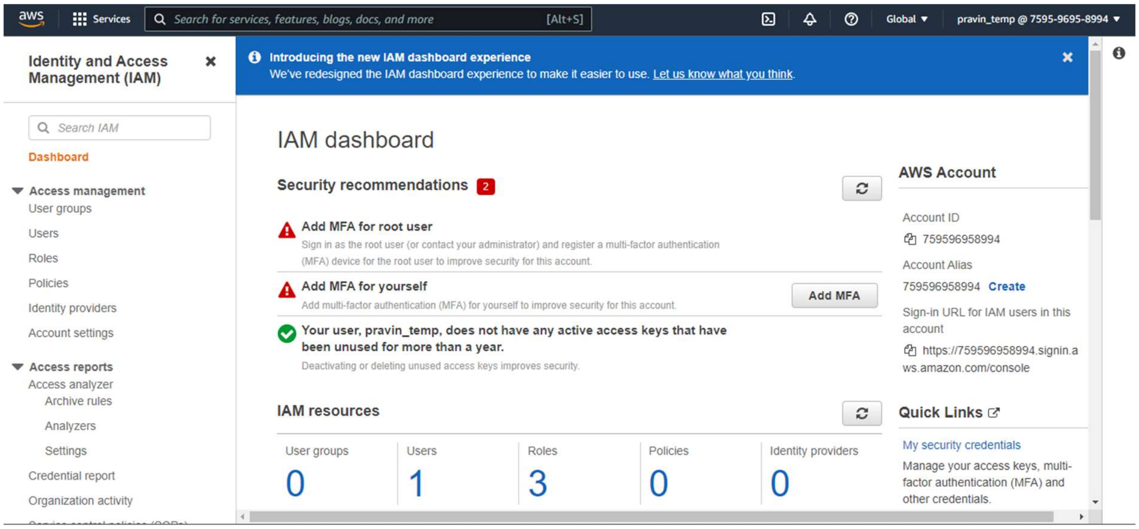
Features — See all 100 results ▸

Streams
Amazon QLDB feature

S Account
ount ID
759596958994
ount Alias
596958994 Create
-in URL for IAM users in this ount
https://759596958994.signin.a mazon.com/console
ck Links
security credentials
age your access keys, multi- r authentication (MFA) and r credentials.

Feedback Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

4) In search bar type iam and Click on IAM manage access to AWS resources



Introducing the new IAM dashboard experience
We've redesigned the IAM dashboard experience to make it easier to use. Let us know what you think.

IAM dashboard

Security recommendations 2

Add MFA for root user
Sign in as the root user (or contact your administrator) and register a multi-factor authentication (MFA) device for the root user to improve security for this account.

Add MFA for yourself
Add multi-factor authentication (MFA) for yourself to improve security for this account.

Your user, pravin_temp, does not have any active access keys that have been unused for more than a year.
Deactivating or deleting unused access keys improves security.

IAM resources

User groups	Users	Roles	Policies	Identity providers
0	1	3	0	0

AWS Account

Account ID
759596958994

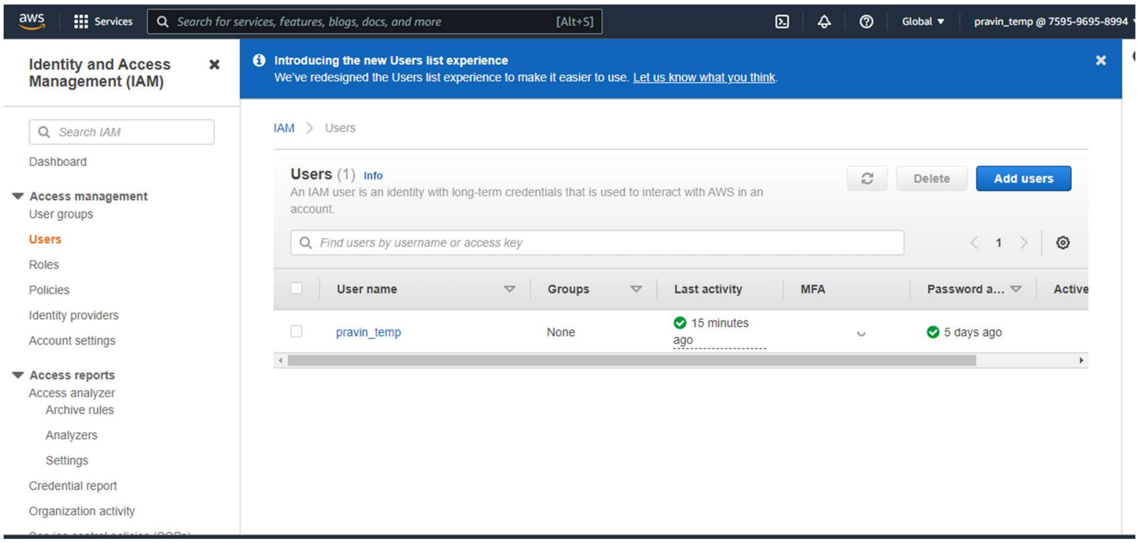
Account Alias
759596958994 Create

Sign-in URL for IAM users in this account
https://759596958994.signin.aws.amazon.com/console

Quick Links

My security credentials
Manage your access keys, multi-factor authentication (MFA) and other credentials.

5) Click on left menu Access management -> Users

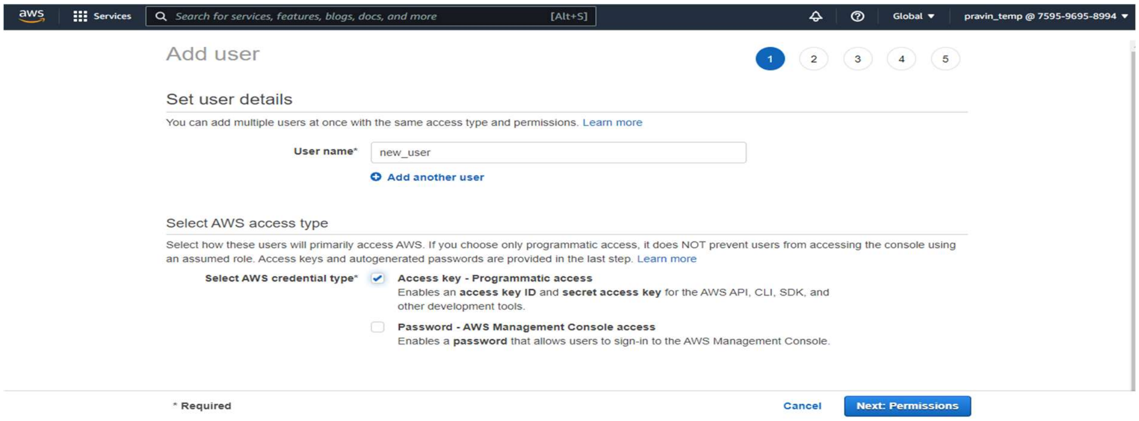


Introducing the new Users list experience
We've redesigned the Users list experience to make it easier to use. Let us know what you think.

Users (1) Info
An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

□	User name	Groups	Last activity	MFA	Password a...	Active
□	pravin_temp	None	15 minutes ago		5 days ago	

6) Click on Add users Button



aws
Services
Search for services, features, blogs, docs, and more [Alt+S]
Global
pravin_temp @ 7595-9695-8994
1
2
3
4
5
Cancel
Next: Permissions

Add user

Set user details

You can add multiple users at once with the same access type and permissions. Learn more

User name* new_user

Add another user

Select AWS access type

Select how these users will primarily access AWS. If you choose only programmatic access, it does NOT prevent users from accessing the console using an assumed role. Access keys and autogenerated passwords are provided in the last step. Learn more

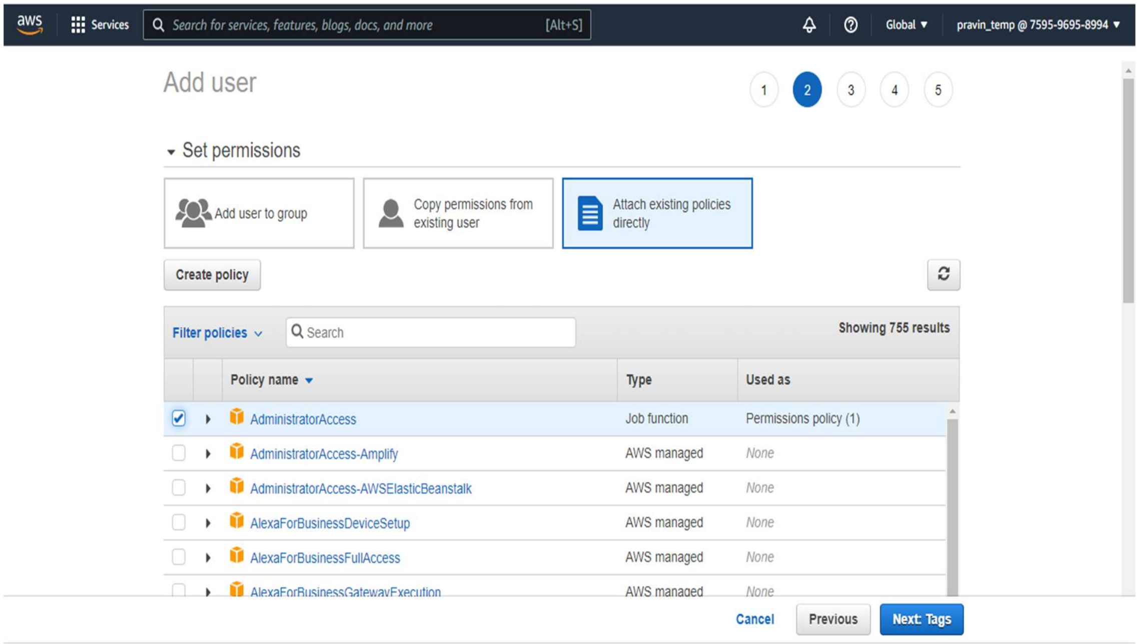
Select AWS credential type*

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

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

* Required

7) Fill **User name**, checked the box **Access Key – Programatic access** and Click on **Next: Permissions**


aws
Services
Search for services, features, blogs, docs, and more
[Alt+S]
Global ▾
pravin_temp @ 7595-9695-8994 ▾
1
2
3
4
5
Add user to group
Copy permissions from existing user
Attach existing policies directly
Create policy
Filter policies ⌄
Search
Cancel
Previous
Next: Tags

Add user

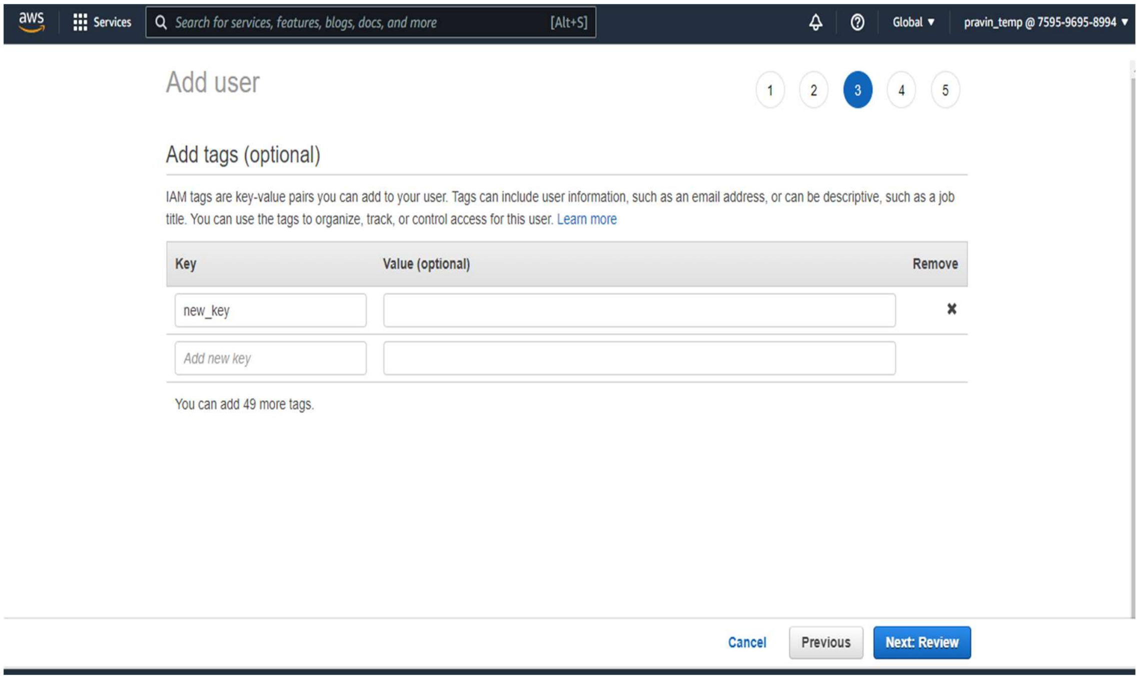
▾ Set permissions

Showing 755 results

		Policy name ▾	Type	Used as
☑	▸	AdministratorAccess	Job function	Permissions policy (1)
□	▸	AdministratorAccess-Amplify	AWS managed	None
□	▸	AdministratorAccess-AWSElasticBeanstalk	AWS managed	None
□	▸	AlexaForBusinessDeviceSetup	AWS managed	None
□	▸	AlexaForBusinessFullAccess	AWS managed	None
□	▸	AlexaForBusinessGatewayExecution	AWS managed	None

8) Select **Attach existing policies directly,** In Filter policies select **AdministratorAccess** And Click on **Next: Tags**

Note: You can also select multiple **Filter policies** according to requirement and Understanding of AWS Filter policies.


aws
Services
Search for services, features, blogs, docs, and more
[Alt+S]
Global ▾
pravin_temp @ 7595-9695-8994 ▾
1
2
3
4
5
Cancel
Previous
Next: Review

Add user

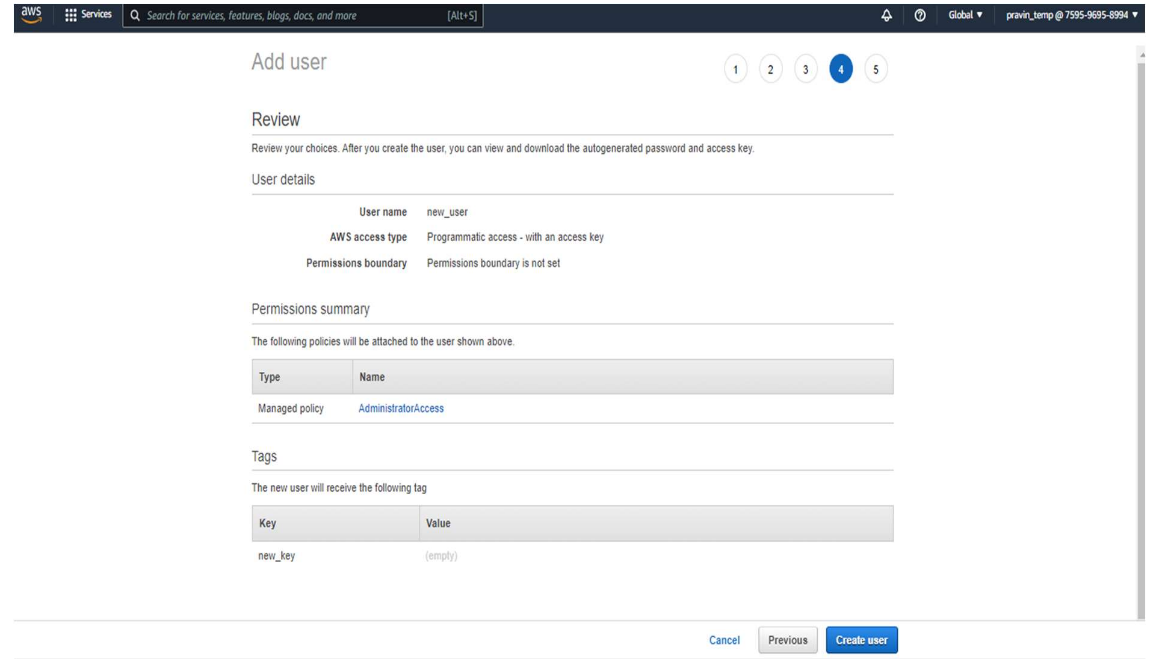
Add tags (optional)

IAM tags are key-value pairs you can add to your user. Tags can include user information, such as an email address, or can be descriptive, such as a job title. You can use the tags to organize, track, or control access for this user. Learn more

Key	Value (optional)	Remove
new_key		✖
Add new key		

You can add 49 more tags.

9) Fill the **Key** Value and click on **Next: Review**



aws
Services
Search for services, features, blogs, docs, and more
[Alt+S]
Global ▾
pravin_temp @ 7595-9695-8994 ▾
1
2
3
4
5
Cancel
Previous
Create user

Add user

Review

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

User details

User name	new_user
AWS access type	Programmatic access - with an access key
Permissions boundary	Permissions boundary is not set

Permissions summary

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

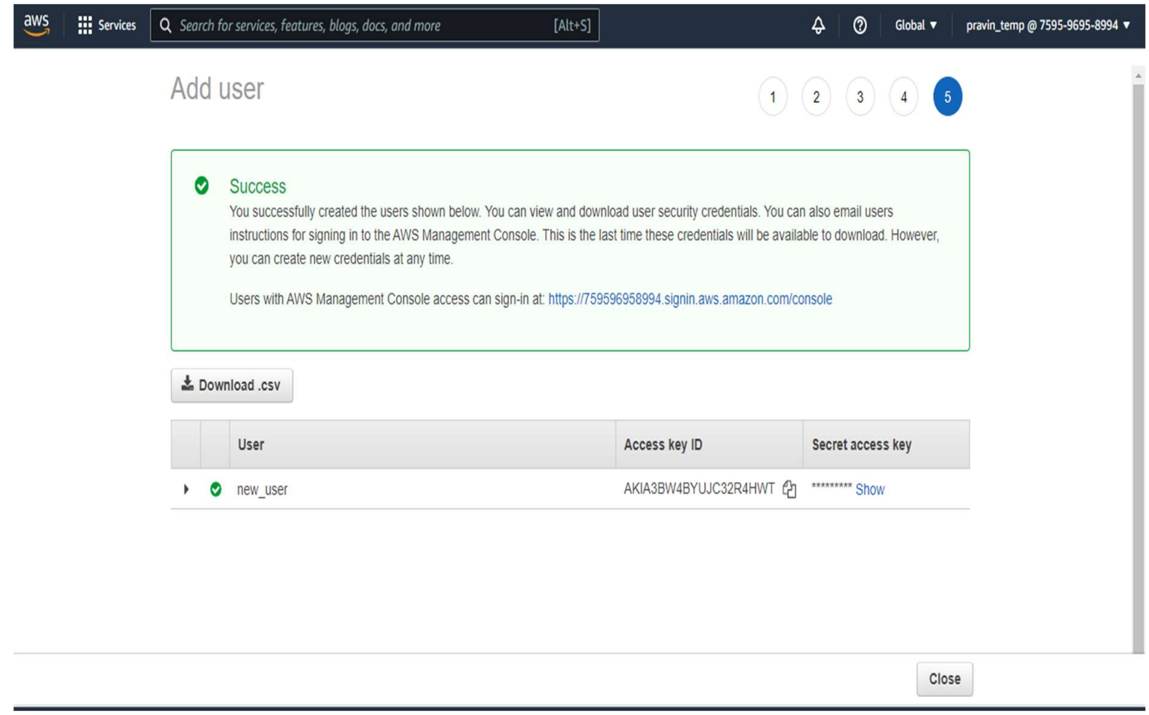
Type	Name
Managed policy	AdministratorAccess

Tags

The new user will receive the following tag

Key	Value
new_key	(empty)

10) Click on **Create user**



aws
Services
Search for services, features, blogs, docs, and more
[Alt+S]
Global ▾
pravin_temp @ 7595-9695-8994 ▾
1
2
3
4
5
Download .csv
Show
Close

Add user

Success

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

Users with AWS Management Console access can sign-in at: https://759596958994.signin.aws.amazon.com/console

User	Access key ID	Secret access key
new_user	AKIA3BW4BYUJC32R4HWT	******** Show

After User created successfully. Click on **Download.csv.** It downloads the .csv file that has **Access key ID** and **Secret access key** Values. These values required in setup the AWS CLI.

Install or update the AWS CLI for Windows

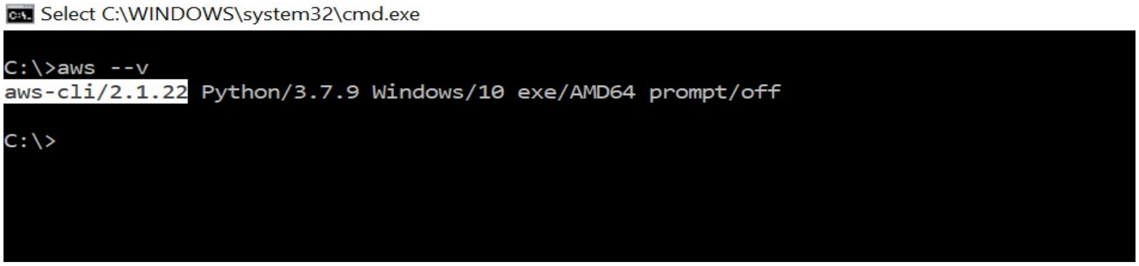
https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html

Follow the above url or install AWS CLI or Directly download and install from below url

https://awscli.amazonaws.com/AWSCLIV2.msi

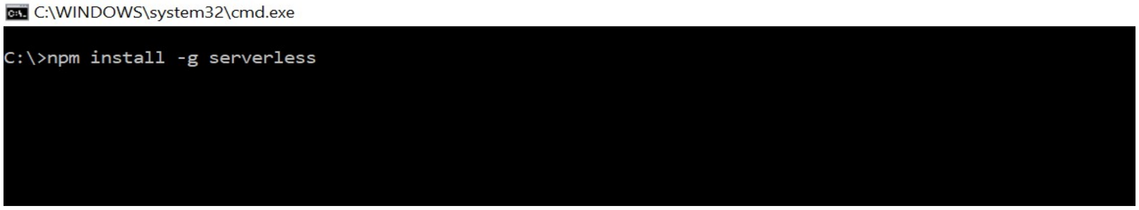
1) After Installation of AWS CLI, check AWS CLI installed or not

Type -> aws --v



```
C:\>aws --v
aws-cli/2.1.22 Python/3.7.9 Windows/10 exe/AMD64 prompt/off

C:\>
```

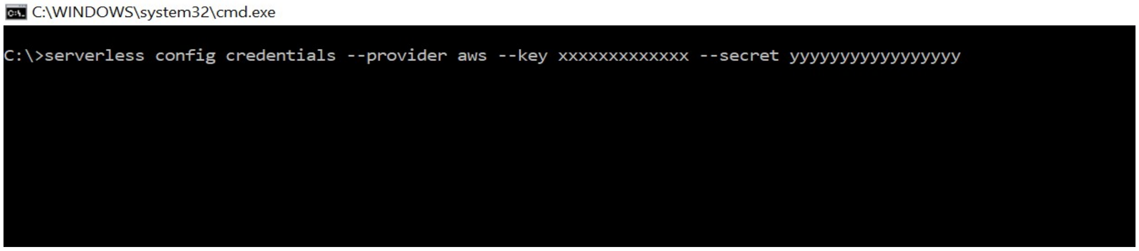
2) Install Serverless Framework



```
C:\>npm install -g serverless
```

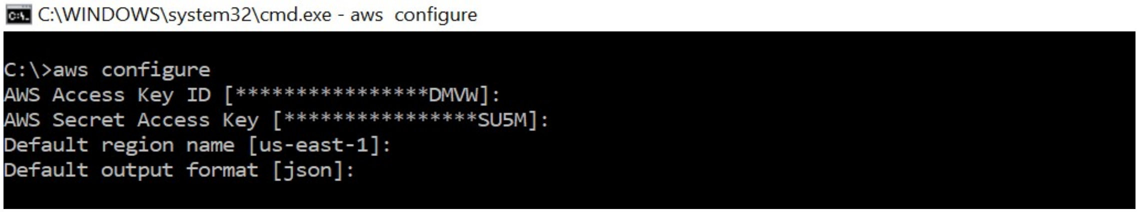
3) After installation of Serverless Framework configure the serverless configuration

Xxxxxxxx is a **Access key ID** and yyyyyyyyy is a **Secret access key.** This Access key and secret key get from .csv file. You have already downloaded. Follow the below command



```
C:\>serverless config credentials --provider aws --key xxxxxxxxxxxxx --secret yyyyyyyyyyyyyyyyy
```

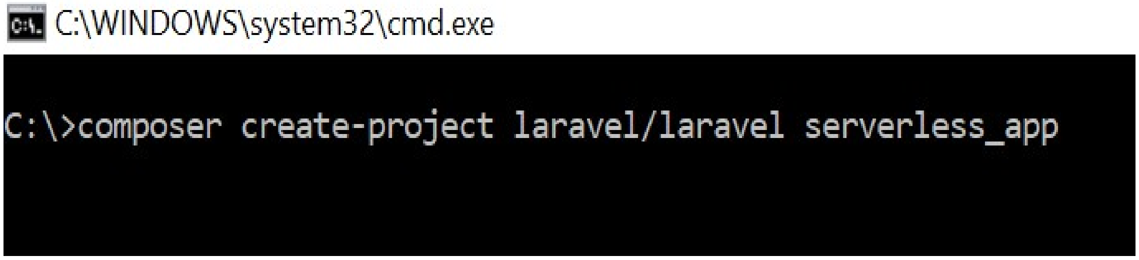
4) Check Serverless Framework configure or not



```
C:\>aws configure
AWS Access Key ID [****************DMVW]:
AWS Secret Access Key [****************SU5M]:
Default region name [us-east-1]:
Default output format [json]:
```

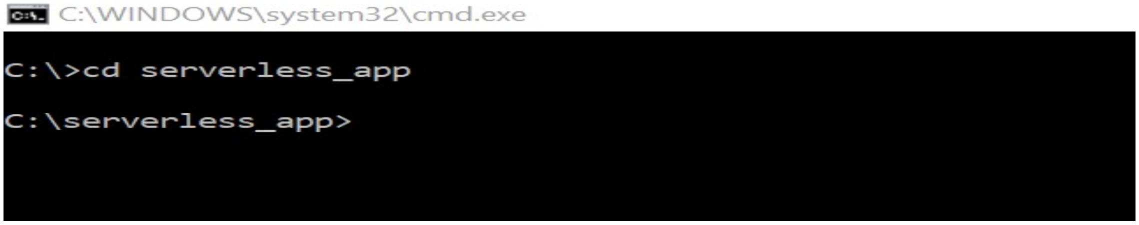
Serverless Laravel Project Setup

1) Create Laravel project in your local system, use below command in cmd and hit Enter button



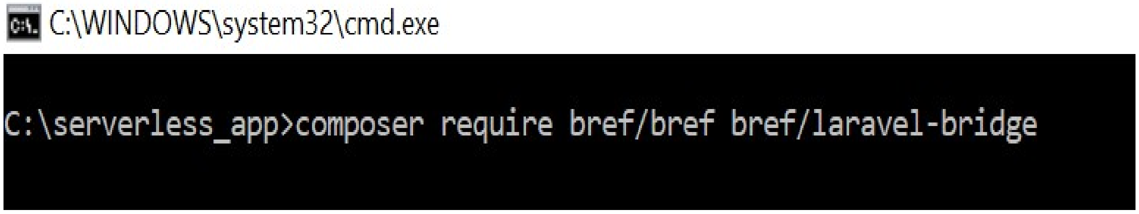
```
C:\>composer create-project laravel/laravel serverless_app
```

2) After completing the installation process, then use below command in cmd



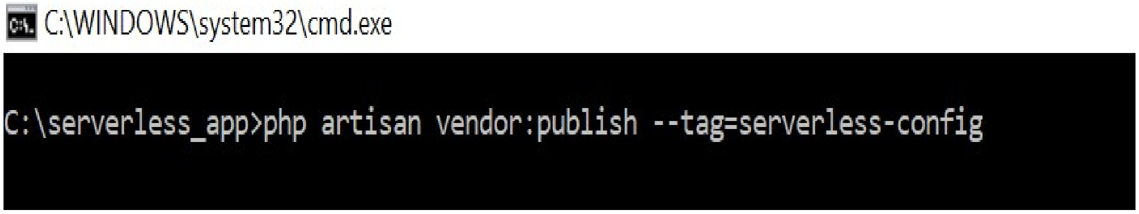
```
C:\>cd serverless_app

C:\serverless_app>
```

3) Install Laravel Bref + Laravel Bridge, use below command in cmd



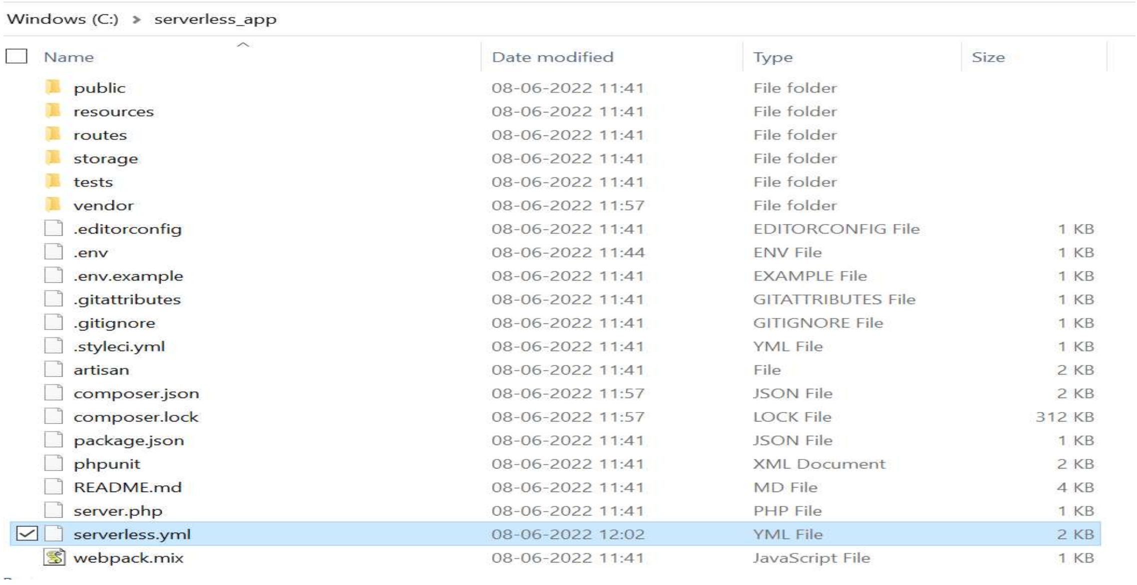
```
C:\serverless_app>composer require bref/bref bref/laravel-bridge
```

4) Then let's create a serverless.yml configuration file:



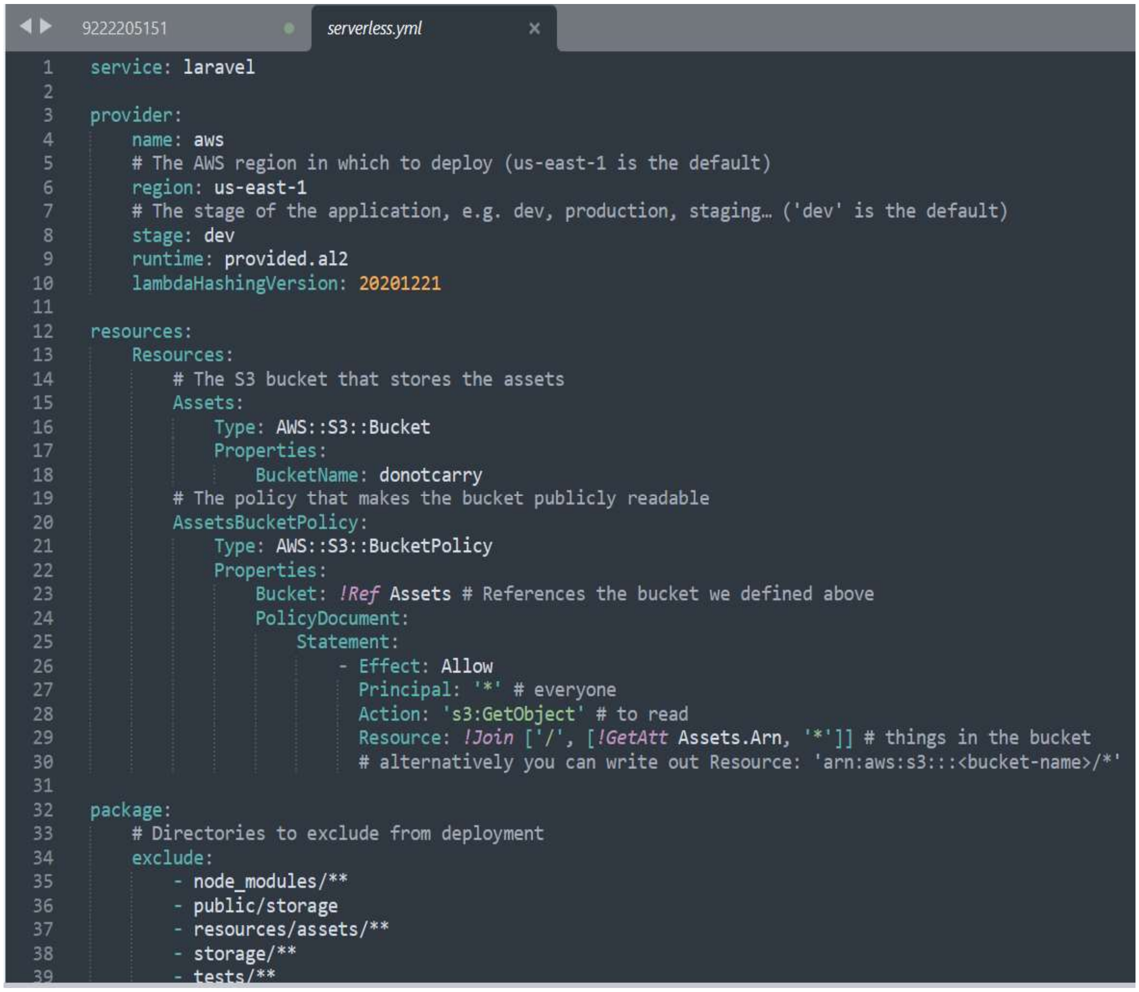
```
C:\serverless_app>php artisan vendor:publish --tag=serverless-config
```

Above command generate the serverless.yml file

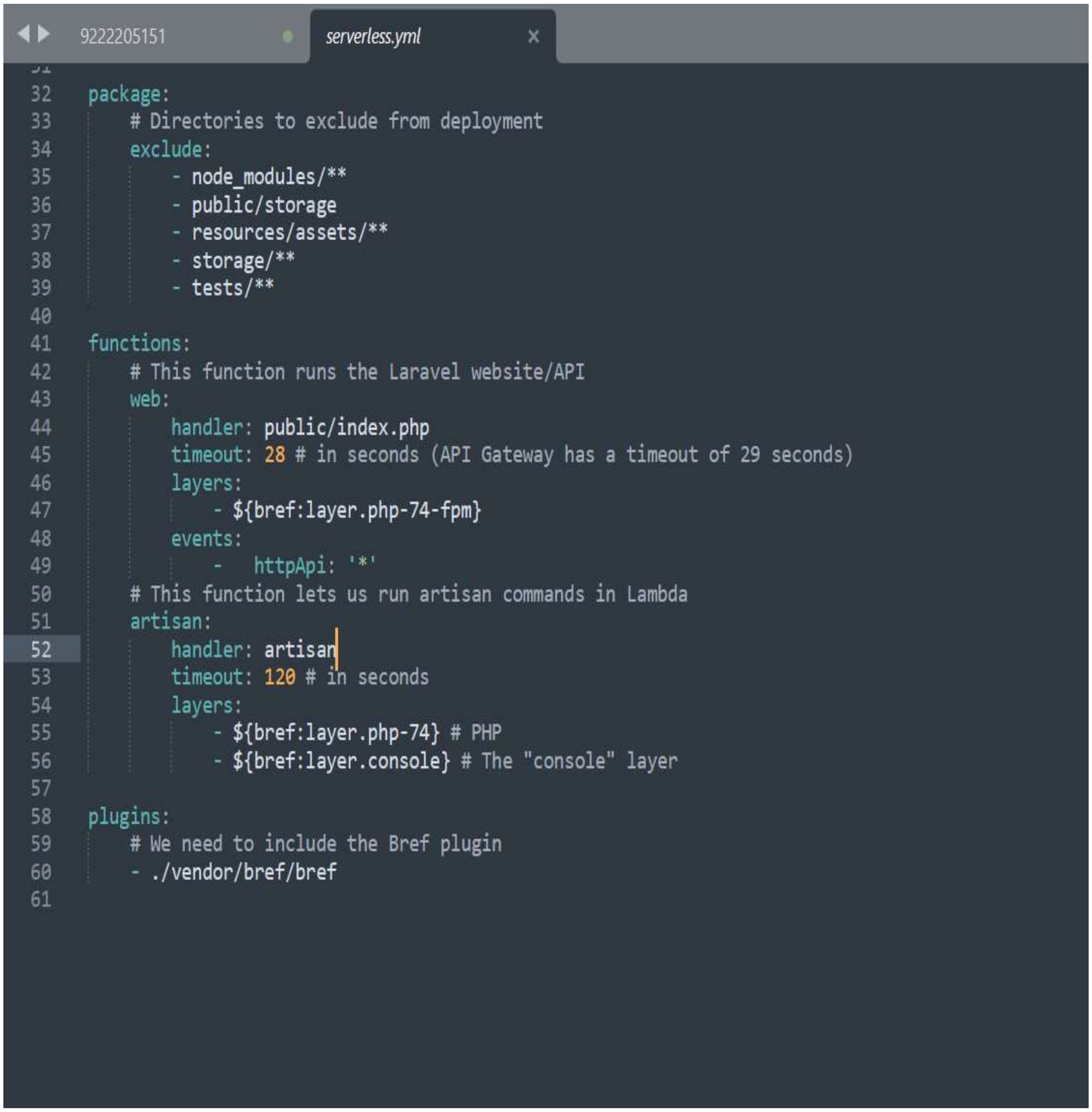


Name	Date modified	Type	Size
public	08-06-2022 11:41	File folder	
resources	08-06-2022 11:41	File folder	
routes	08-06-2022 11:41	File folder	
storage	08-06-2022 11:41	File folder	
tests	08-06-2022 11:41	File folder	
vendor	08-06-2022 11:57	File folder	
.editorconfig	08-06-2022 11:41	EDITORCONFIG File	1 KB
.env	08-06-2022 11:44	ENV File	1 KB
.env.example	08-06-2022 11:41	EXAMPLE File	1 KB
.gitattributes	08-06-2022 11:41	GITATTRIBUTES File	1 KB
.gitignore	08-06-2022 11:41	GITIGNORE File	1 KB
.styleci.yml	08-06-2022 11:41	YML File	1 KB
artisan	08-06-2022 11:41	File	2 KB
composer.json	08-06-2022 11:57	JSON File	2 KB
composer.lock	08-06-2022 11:57	LOCK File	312 KB
package.json	08-06-2022 11:41	JSON File	1 KB
phpunit	08-06-2022 11:41	XML Document	2 KB
README.md	08-06-2022 11:41	MD File	4 KB
server.php	08-06-2022 11:41	PHP File	1 KB
☑ serverless.yml	08-06-2022 12:02	YML File	2 KB
webpack.mix	08-06-2022 11:41	JavaScript File	1 KB

Write the code inside serverless.yml file. According to the understanding of .yml and AWS



```yaml
service: laravel

provider:
    name: aws
    # The AWS region in which to deploy (us-east-1 is the default)
    region: us-east-1
    # The stage of the application, e.g. dev, production, staging… ('dev' is the default)
    stage: dev
    runtime: provided.al2
    lambdaHashingVersion: 20201221

resources:
    Resources:
        # The S3 bucket that stores the assets
        Assets:
            Type: AWS::S3::Bucket
            Properties:
                BucketName: donotcarry
        # The policy that makes the bucket publicly readable
        AssetsBucketPolicy:
            Type: AWS::S3::BucketPolicy
            Properties:
                Bucket: !Ref Assets # References the bucket we defined above
                PolicyDocument:
                    Statement:
                        - Effect: Allow
                          Principal: '*' # everyone
                          Action: 's3:GetObject' # to read
                          Resource: !Join ['/', [!GetAtt Assets.Arn, '*']] # things in the bucket
                          # alternatively you can write out Resource: 'arn:aws:s3:::<bucket-name>/*'

package:
    # Directories to exclude from deployment
    exclude:
        - node_modules/**
        - public/storage
        - resources/assets/**
        - storage/**
        - tests/**
```



```yaml
package:
    # Directories to exclude from deployment
    exclude:
        - node_modules/**
        - public/storage
        - resources/assets/**
        - storage/**
        - tests/**

functions:
    # This function runs the Laravel website/API
    web:
        handler: public/index.php
        timeout: 28 # in seconds (API Gateway has a timeout of 29 seconds)
        layers:
            - ${bref:layer.php-74-fpm}
        events:
            -   httpApi: '*'
    # This function lets us run artisan commands in Lambda
    artisan:
        handler: artisan
        timeout: 120 # in seconds
        layers:
            - ${bref:layer.php-74} # PHP
            - ${bref:layer.console} # The "console" layer

plugins:
    # We need to include the Bref plugin
    - ./vendor/bref/bref
```

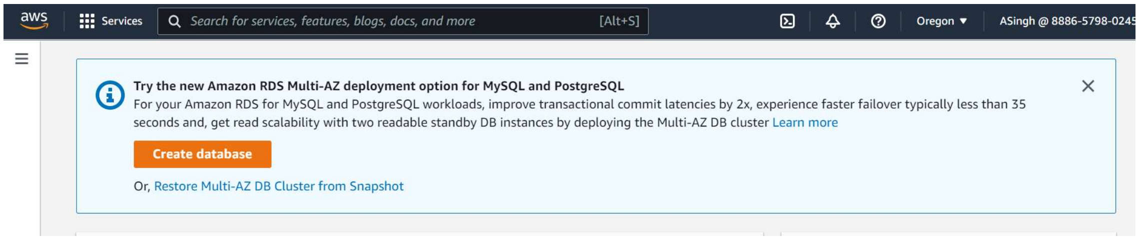
5) Let’s change .env configuration file

- ➔ LOG_CHANNEL=stack **To** LOG_CHANNEL=stderr
- ➔ SESSION_DRIVER=file **To** SESSION_DRIVER=cookie
- ➔ Add new variable - VIEW_COMPILED_PATH=/tmp/storage/framework/views

RDS Database Setup

Follow url for creating the RDS
https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_Tutorials.WebServerDB.CreateDBInstance.html

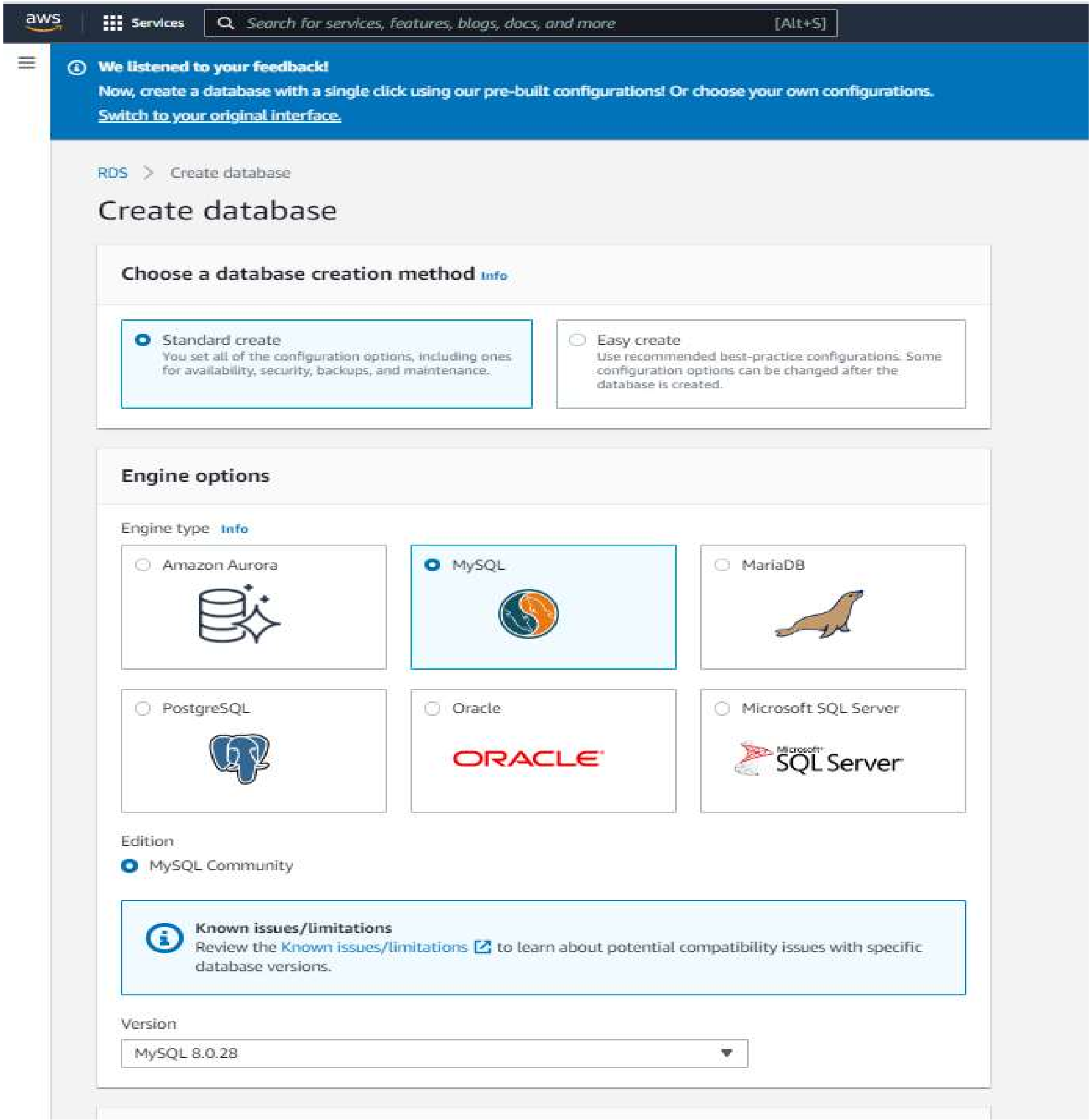
1) Login to the AWS Account and Click on https://console.aws.amazon.com/rds/



Try the new Amazon RDS Multi-AZ deployment option for MySQL and PostgreSQL
For your Amazon RDS for MySQL and PostgreSQL workloads, improve transactional commit latencies by 2x, experience faster failover typically less than 35 seconds and, get read scalability with two readable standby DB instances by deploying the Multi-AZ DB cluster Learn more

Or, Restore Multi-AZ DB Cluster from Snapshot

2) Click on create database



We listened to your feedback!
Now, create a database with a single click using our pre-built configurations! Or choose your own configurations.
Switch to your original interface.

Create database

Choose a database creation method Info

● Standard create
You set all of the configuration options, including ones for availability, security, backups, and maintenance.

○ Easy create
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

Engine options

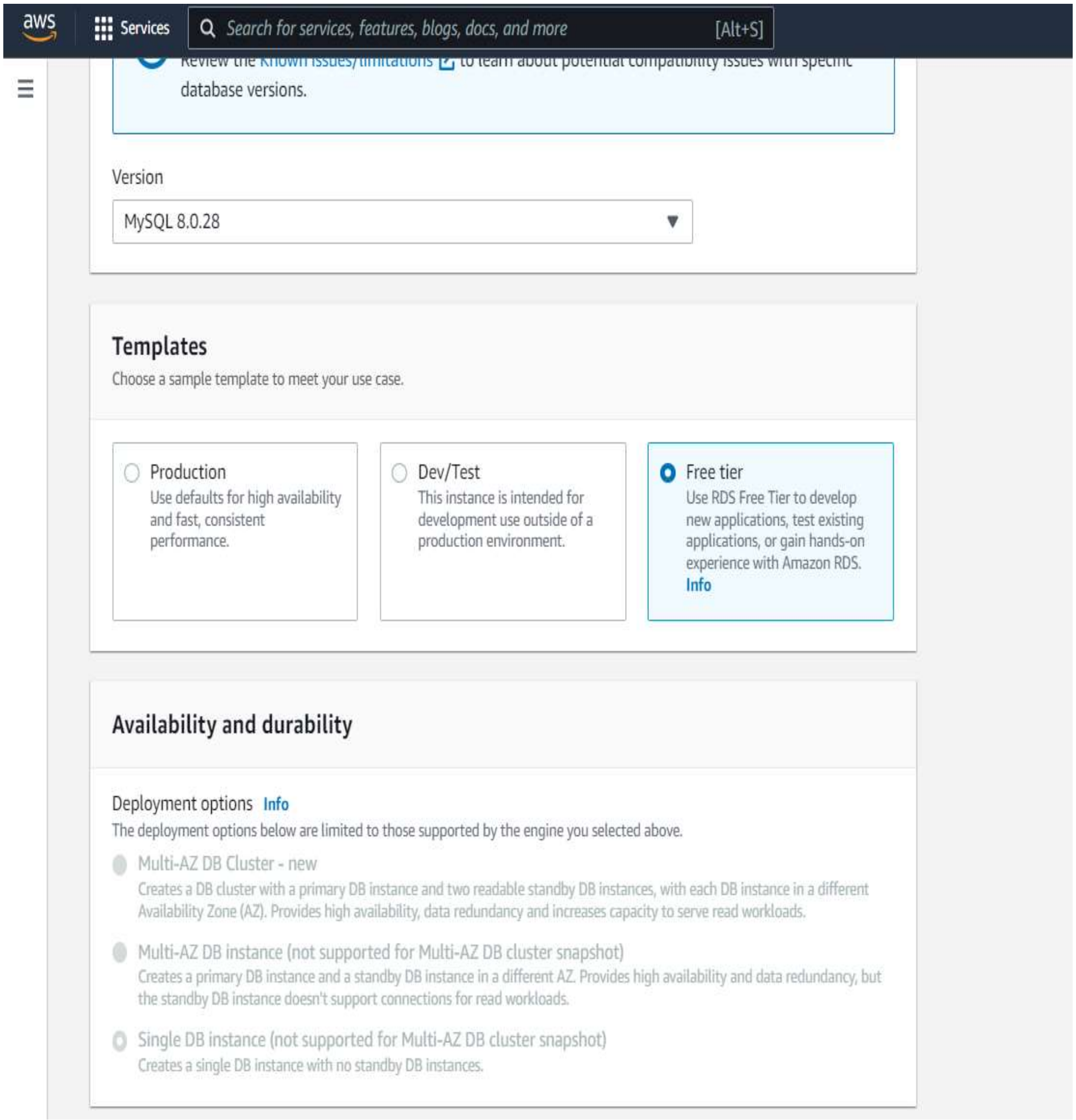
Engine type Info

○ Amazon Aurora
● MySQL
○ MariaDB
○ PostgreSQL
○ Oracle
○ Microsoft SQL Server

Edition
● MySQL Community

Known issues/limitations
Review the Known issues/limitations to learn about potential compatibility issues with specific database versions.

Version
MySQL 8.0.28 ▾



Review the known issues/limitations to learn about potential compatibility issues with specific database versions.

Version

MySQL 8.0.28 ▼

Templates

Choose a sample template to meet your use case.

○ Production
Use defaults for high availability and fast, consistent performance.

○ Dev/Test
This instance is intended for development use outside of a production environment.

● Free tier
Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS. Info

Availability and durability

Deployment options Info

The deployment options below are limited to those supported by the engine you selected above.

○ Multi-AZ DB Cluster - new
Creates a DB cluster with a primary DB instance and two readable standby DB instances, with each DB instance in a different Availability Zone (AZ). Provides high availability, data redundancy and increases capacity to serve read workloads.

○ Multi-AZ DB instance (not supported for Multi-AZ DB cluster snapshot)
Creates a primary DB instance and a standby DB instance in a different AZ. Provides high availability and data redundancy, but the standby DB instance doesn't support connections for read workloads.

● Single DB instance (not supported for Multi-AZ DB cluster snapshot)
Creates a single DB instance with no standby DB instances.

3) In Setting Section

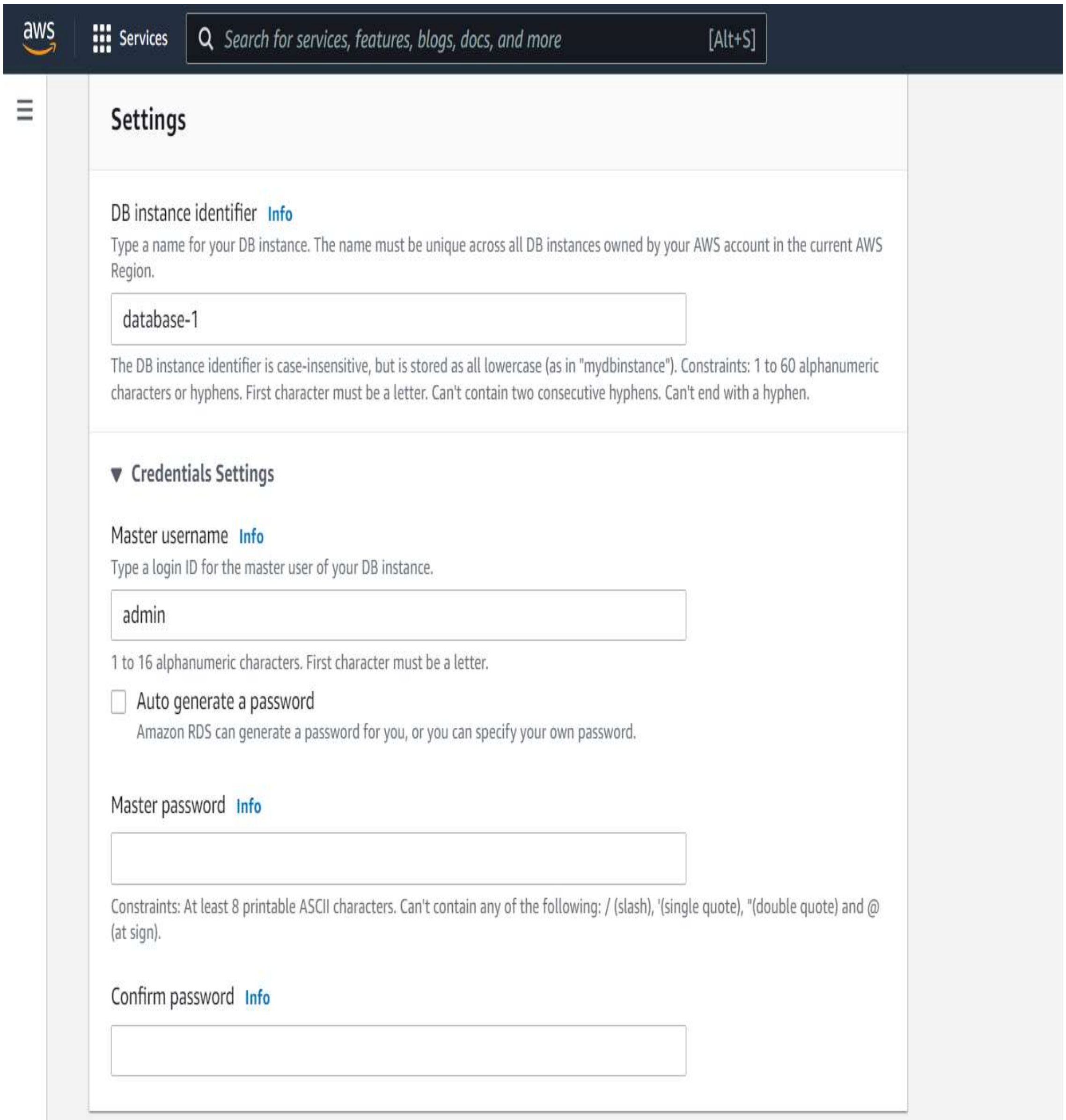
DB instance identifier – database-1 **or** any thing you want

Master username – admin **or** anything you want

Auto generate a password – Clear the check box.

Master password – Choose a password.

Confirm password – Retype the password.



Settings

DB instance identifier Info

Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

database-1

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

▼ Credentials Settings

Master username Info

Type a login ID for the master user of your DB instance.

admin

1 to 16 alphanumeric characters. First character must be a letter.

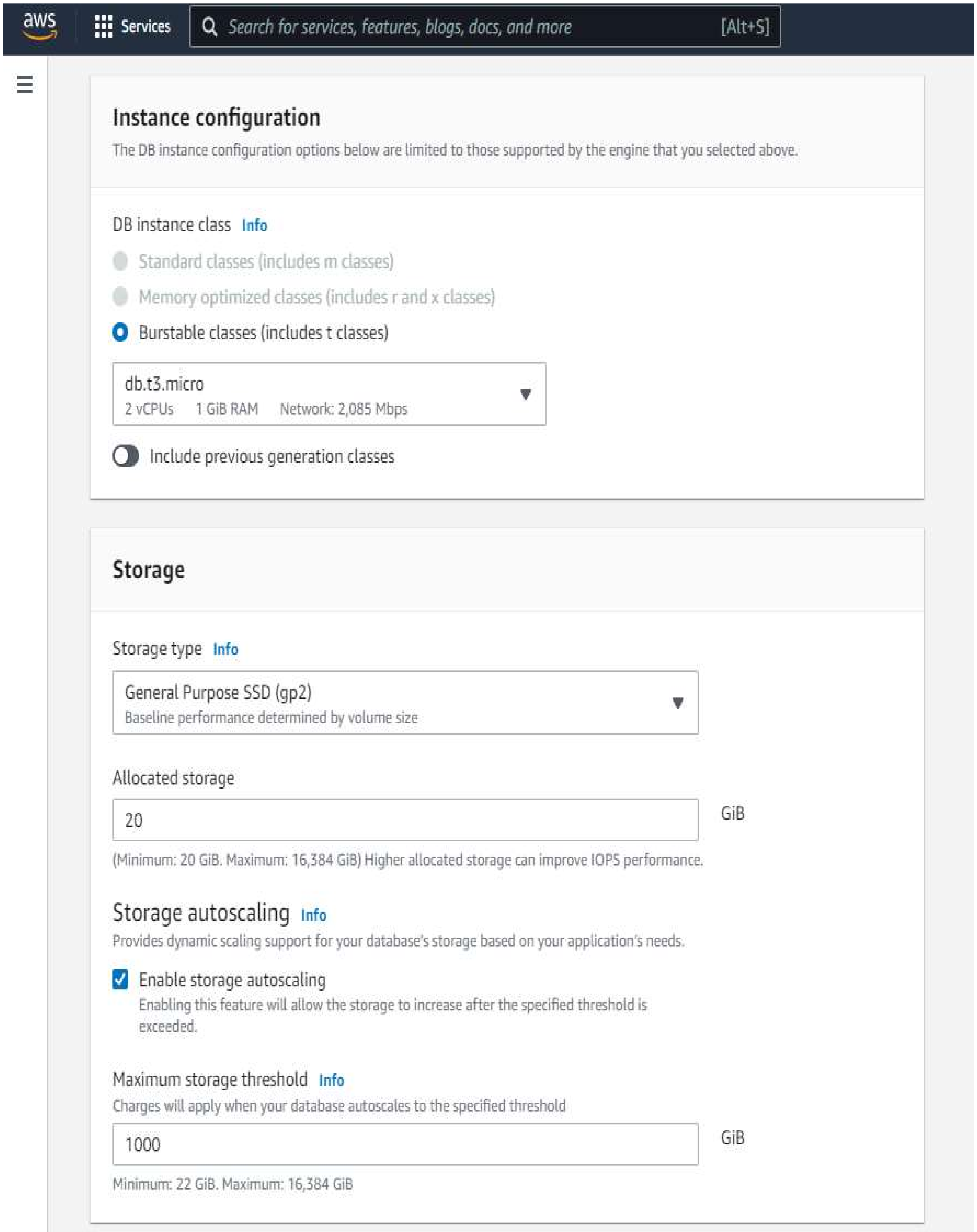
□ Auto generate a password

Amazon RDS can generate a password for you, or you can specify your own password.

Master password Info

Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), '(single quote), "(double quote) and @ (at sign).

Confirm password Info



Instance configuration

The DB instance configuration options below are limited to those supported by the engine that you selected above.

DB instance class Info

○ Standard classes (includes m classes)

○ Memory optimized classes (includes r and x classes)

● Burstable classes (includes t classes)

db.t3.micro
2 vCPUs 1 GiB RAM Network: 2,085 Mbps ▼

○ Include previous generation classes

Storage

Storage type Info

General Purpose SSD (gp2)
Baseline performance determined by volume size ▼

Allocated storage

20 GiB

(Minimum: 20 GiB. Maximum: 16,384 GiB) Higher allocated storage can improve IOPS performance.

Storage autoscaling Info

Provides dynamic scaling support for your database's storage based on your application's needs.

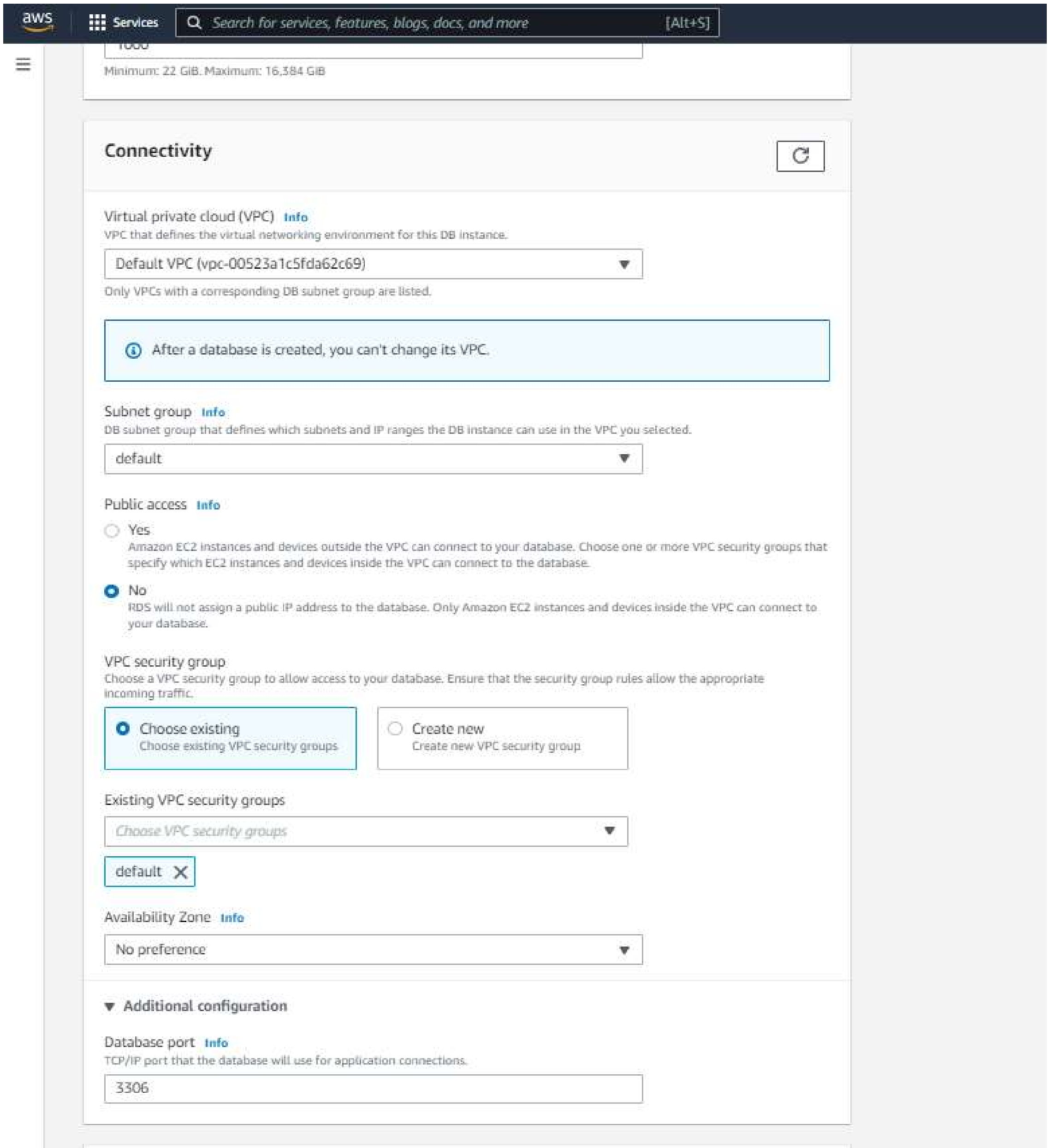
☑ Enable storage autoscaling
Enabling this feature will allow the storage to increase after the specified threshold is exceeded.

Maximum storage threshold Info

Charges will apply when your database autoscales to the specified threshold

1000 GiB

Minimum: 22 GiB. Maximum: 16,384 GiB



1000

Minimum: 22 GiB. Maximum: 16,384 GiB

Connectivity

Virtual private cloud (VPC) Info

VPC that defines the virtual networking environment for this DB instance.

Default VPC (vpc-00523a1c5fda62c69) ▼

Only VPCs with a corresponding DB subnet group are listed.

After a database is created, you can't change its VPC.

Subnet group Info

DB subnet group that defines which subnets and IP ranges the DB instance can use in the VPC you selected.

default ▼

Public access Info

○ Yes

Amazon EC2 instances and devices outside the VPC can connect to your database. Choose one or more VPC security groups that specify which EC2 instances and devices inside the VPC can connect to the database.

● No

RDS will not assign a public IP address to the database. Only Amazon EC2 instances and devices inside the VPC can connect to your database.

VPC security group

Choose a VPC security group to allow access to your database. Ensure that the security group rules allow the appropriate incoming traffic.

● Choose existing
Choose existing VPC security groups

○ Create new
Create new VPC security group

Existing VPC security groups

Choose VPC security groups ▼

default ✕

Availability Zone Info

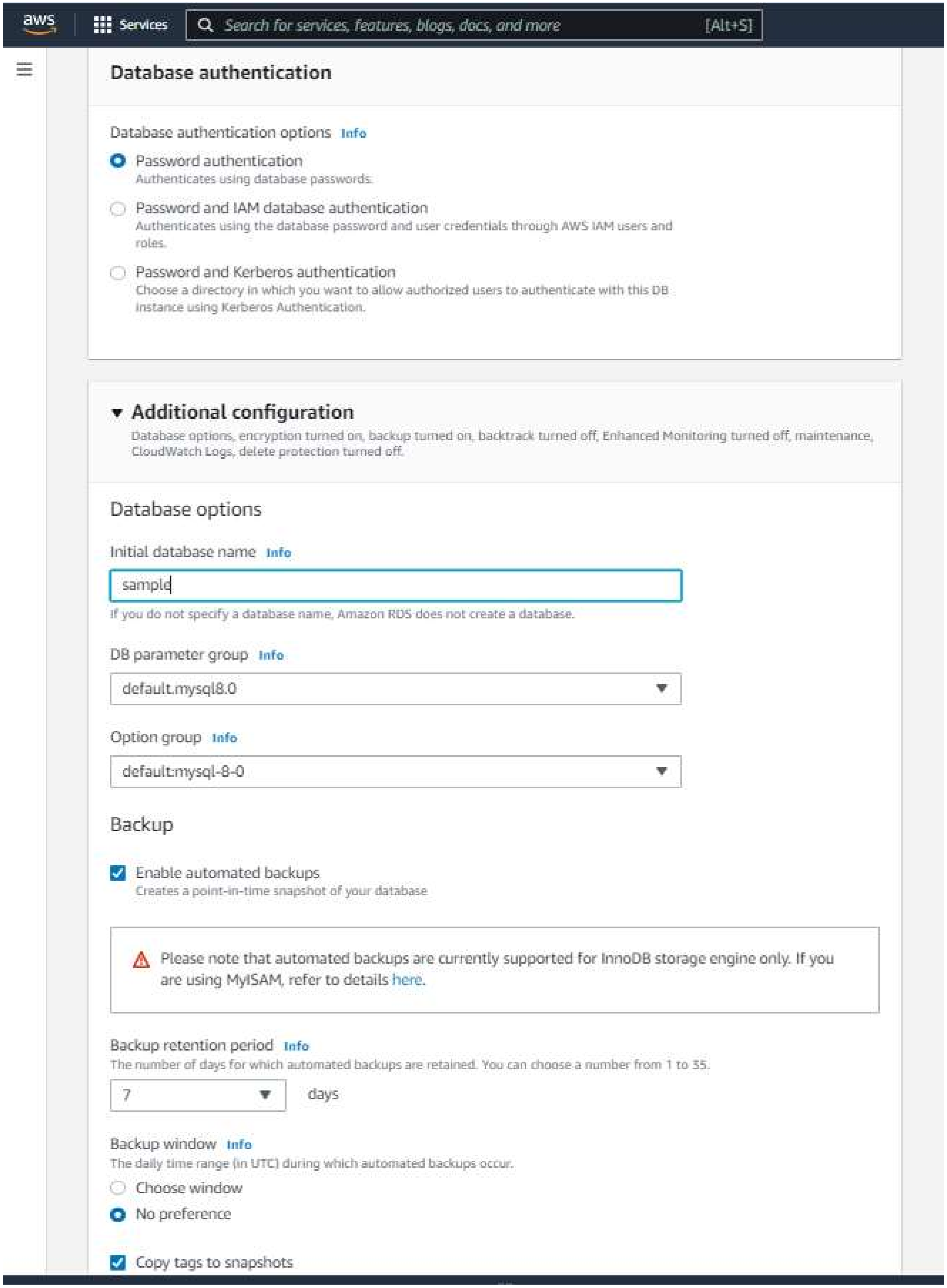
No preference ▼

▼ Additional configuration

Database port Info

TCP/IP port that the database will use for application connections.

3306



Database authentication

Database authentication options Info

● Password authentication
Authenticates using database passwords.

○ Password and IAM database authentication
Authenticates using the database password and user credentials through AWS IAM users and roles.

○ Password and Kerberos authentication
Choose a directory in which you want to allow authorized users to authenticate with this DB instance using Kerberos Authentication.

▼ Additional configuration
Database options, encryption turned on, backup turned on, backtrack turned off, Enhanced Monitoring turned off, maintenance, CloudWatch Logs, delete protection turned off.

Database options

Initial database name Info

sample

If you do not specify a database name, Amazon RDS does not create a database.

DB parameter group Info

default.mysql8.0 ▼

Option group Info

default:mysql-8-0 ▼

Backup

☑ Enable automated backups
Creates a point-in-time snapshot of your database

⚠ Please note that automated backups are currently supported for InnoDB storage engine only. If you are using MyISAM, refer to details here.

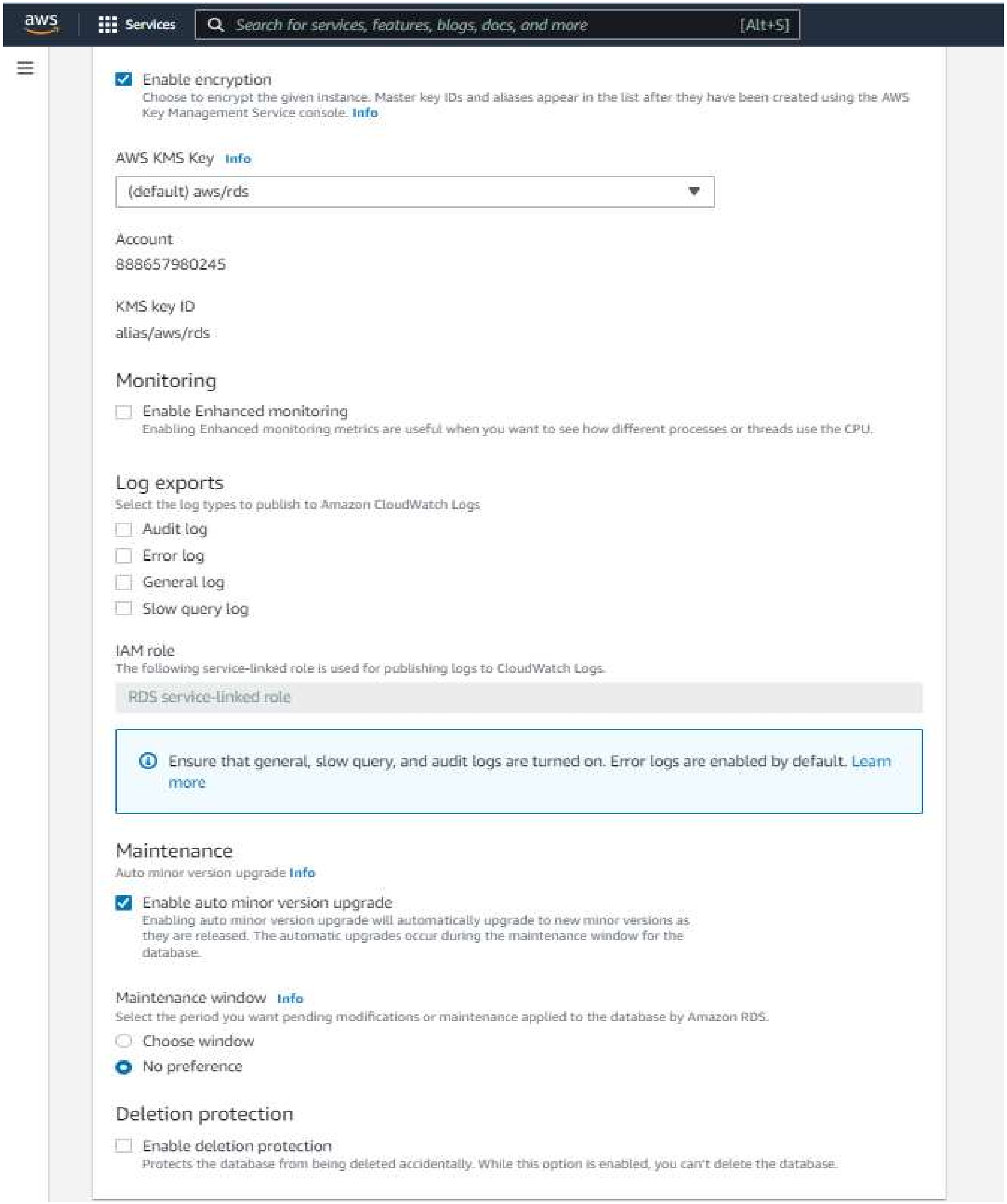
Backup retention period Info
The number of days for which automated backups are retained. You can choose a number from 1 to 35.

7 ▼ days

Backup window Info
The daily time range (in UTC) during which automated backups occur.

○ Choose window
● No preference

☑ Copy tags to snapshots



☑ Enable encryption
Choose to encrypt the given instance. Master key IDs and aliases appear in the list after they have been created using the AWS Key Management Service console. Info

AWS KMS Key Info
(default) aws/rds ▼

Account
888657980245

KMS key ID
alias/aws/rds

Monitoring

□ Enable Enhanced monitoring
Enabling Enhanced monitoring metrics are useful when you want to see how different processes or threads use the CPU.

Log exports
Select the log types to publish to Amazon CloudWatch Logs

□ Audit log
□ Error log
□ General log
□ Slow query log

IAM role
The following service-linked role is used for publishing logs to CloudWatch Logs.
RDS service-linked role

Ensure that general, slow query, and audit logs are turned on. Error logs are enabled by default. Learn more

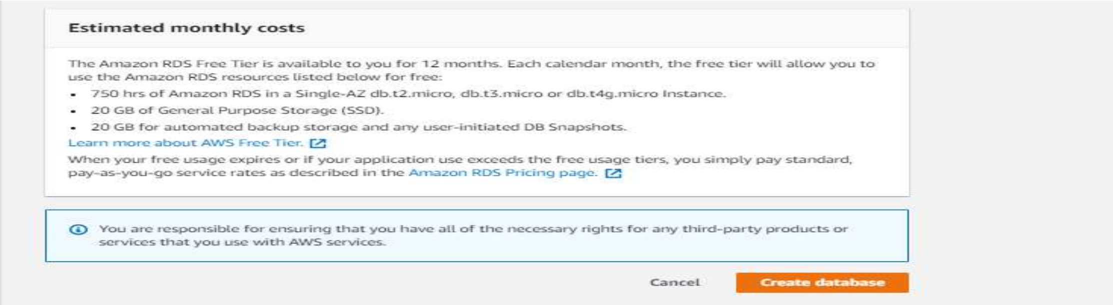
Maintenance
Auto minor version upgrade Info

☑ Enable auto minor version upgrade
Enabling auto minor version upgrade will automatically upgrade to new minor versions as they are released. The automatic upgrades occur during the maintenance window for the database.

Maintenance window Info
Select the period you want pending modifications or maintenance applied to the database by Amazon RDS.
○ Choose window
● No preference

Deletion protection

□ Enable deletion protection
Protects the database from being deleted accidentally. While this option is enabled, you can't delete the database.



Estimated monthly costs

The Amazon RDS Free Tier is available to you for 12 months. Each calendar month, the free tier will allow you to use the Amazon RDS resources listed below for free:

- 750 hrs of Amazon RDS in a Single-AZ db.t2.micro, db.t3.micro or db.t4g.micro Instance.
- 20 GB of General Purpose Storage (SSD).
- 20 GB for automated backup storage and any user-initiated DB Snapshots.

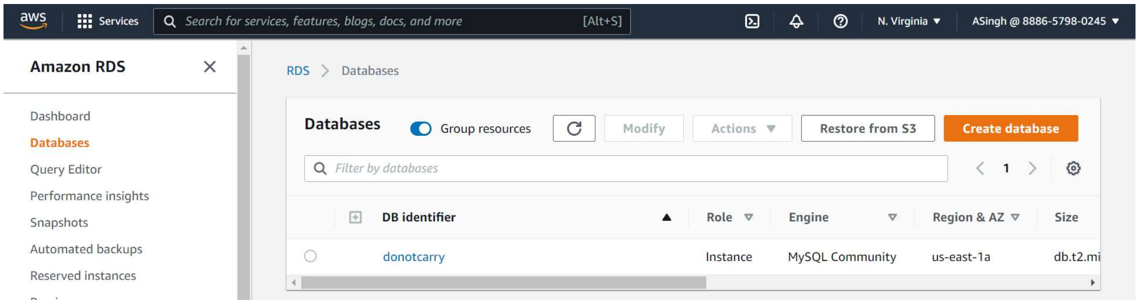
Learn more about AWS Free Tier.

When your free usage expires or if your application use exceeds the free usage tiers, you simply pay standard, pay-as-you-go service rates as described in the Amazon RDS Pricing page.

You are responsible for ensuring that you have all of the necessary rights for any third-party products or services that you use with AWS services.

4) Click on Create database

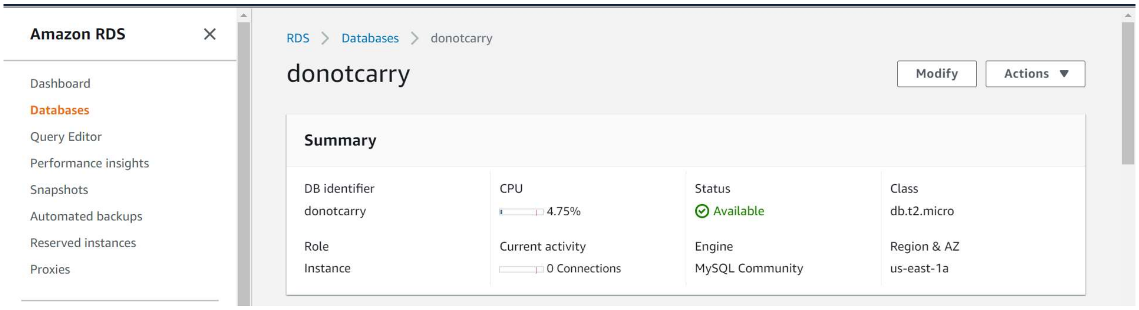
5) Created database show below



Databases

DB identifier	Role	Engine	Region & AZ	Size
donotcarry	Instance	MySQL Community	us-east-1a	db.t2.mi

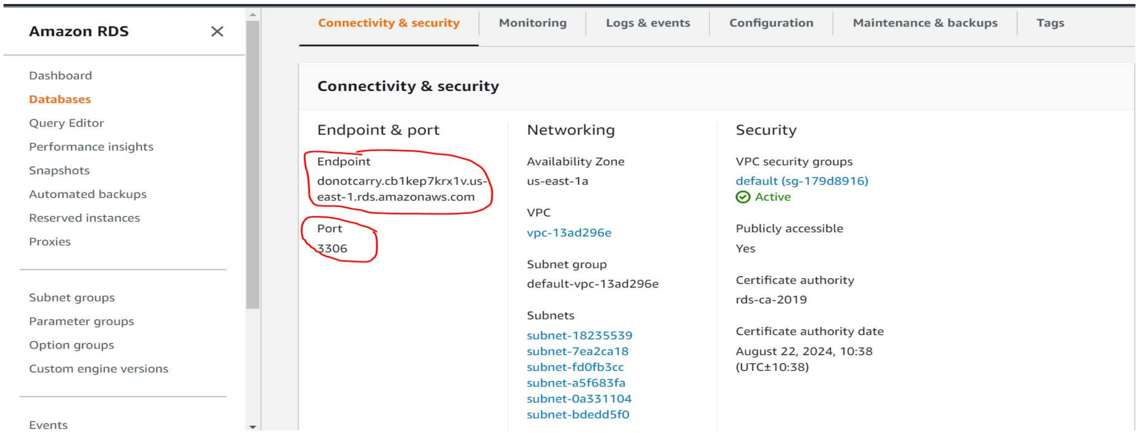
6) Click on donotcarry database



donotcarry

Summary

DB identifier	CPU	Status	Class
donotcarry	4.75%	Available	db.t2.micro
Role	**Current activity**	**Engine**	**Region & AZ**
Instance	0 Connections	MySQL Community	us-east-1a



Connectivity & security

Endpoint & port

Endpoint
donotcarry.cb1kep7krx1v.us-east-1.rds.amazonaws.com

Port
3306

Networking

Availability Zone
us-east-1a

VPC
vpc-13ad296e

Subnet group
default-vpc-13ad296e

Subnets
subnet-18235539
subnet-7ea2ca18
subnet-fd0fb3cc
subnet-a5f683fa
subnet-0a331104
subnet-bdedd5f0

Security

VPC security groups
default (sg-179d8916)
Active

Publicly accessible
Yes

Certificate authority
rds-ca-2019

Certificate authority date
August 22, 2024, 10:38 (UTC±10:38)

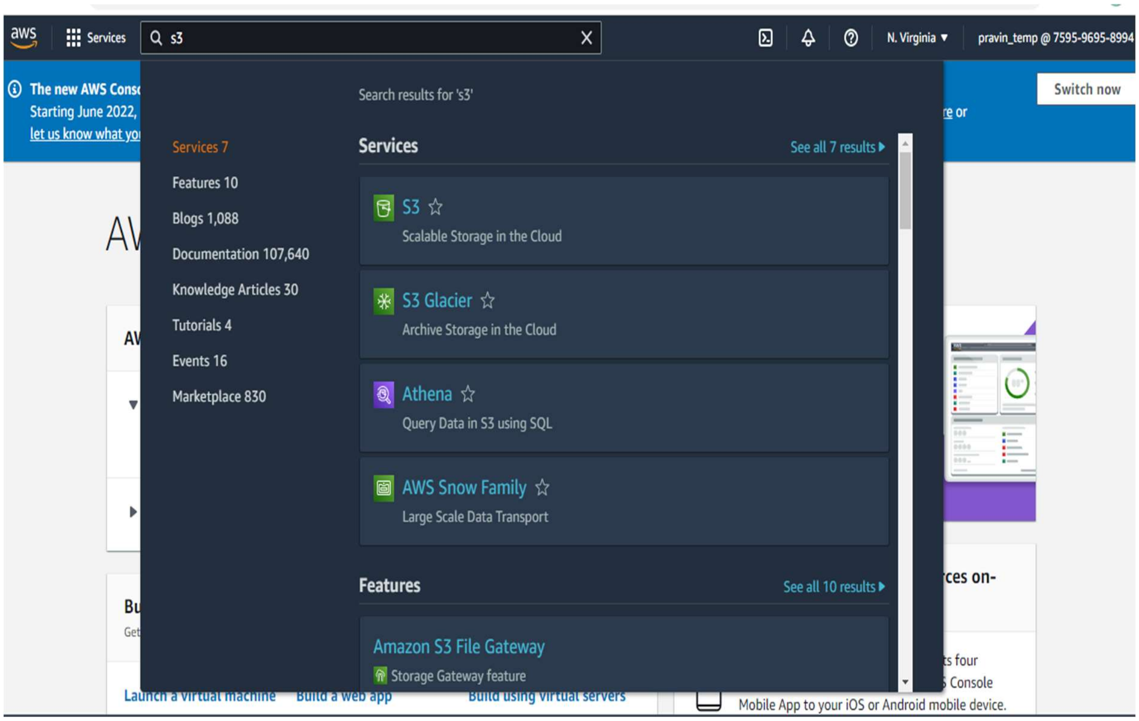
7) Configure database in .env file

```
DB_CONNECTION=mysql

DB_HOST=donotcarry.cb1kep7krx1v.us-east-1.rds.amazonaws.com

DB_PORT=3306

DB_DATABASE=donotcarry

DB_USERNAME=

DB_PASSWORD=
```

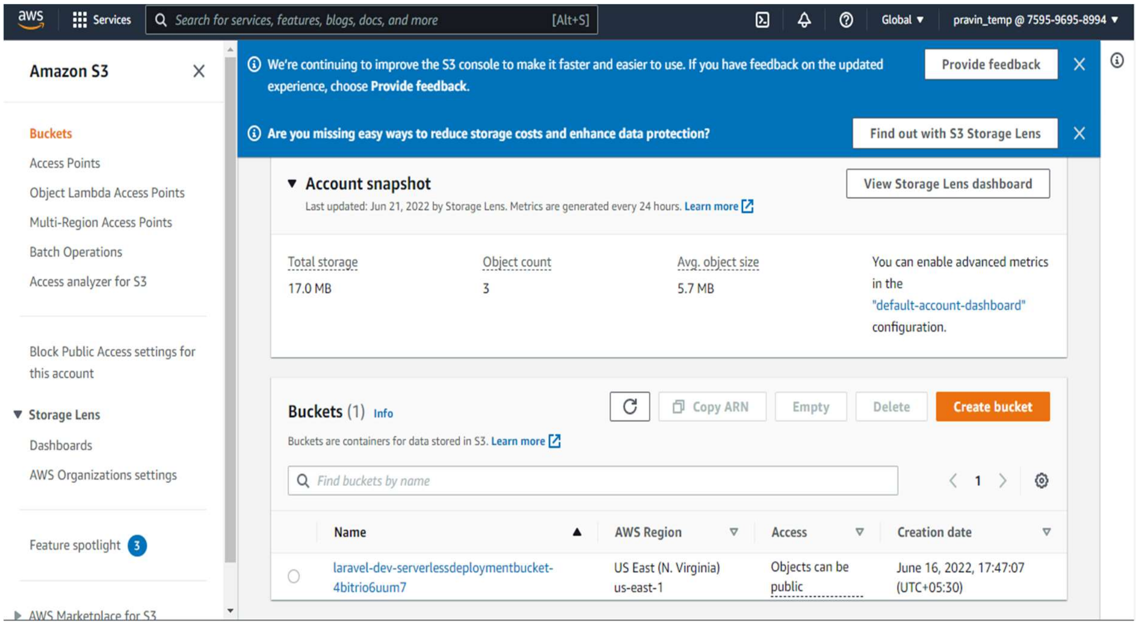
S3 Setup

1) After Login AWS Account

In Search bar type s3



2) Click On S3 Scalable Storage in the Cloud


aws
Services
Search for services, features, blogs, docs, and more
[Alt+S]
Global ▾
pravin_temp @ 7595-9695-8994 ▾
Amazon S3
Buckets
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
Access analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
AWS Organizations settings
Feature spotlight 3
AWS Marketplace for S3
Provide feedback
Find out with S3 Storage Lens
View Storage Lens dashboard
Copy ARN
Empty
Delete
Create bucket
Find buckets by name

We're continuing to improve the S3 console to make it faster and easier to use. If you have feedback on the updated experience, choose **Provide feedback**.

Are you missing easy ways to reduce storage costs and enhance data protection?

Account snapshot

Last updated: Jun 21, 2022 by Storage Lens. Metrics are generated every 24 hours. Learn more

Total storage	Object count	Avg. object size	You can enable advanced metrics in the "default-account-dashboard" configuration.
17.0 MB	3	5.7 MB	

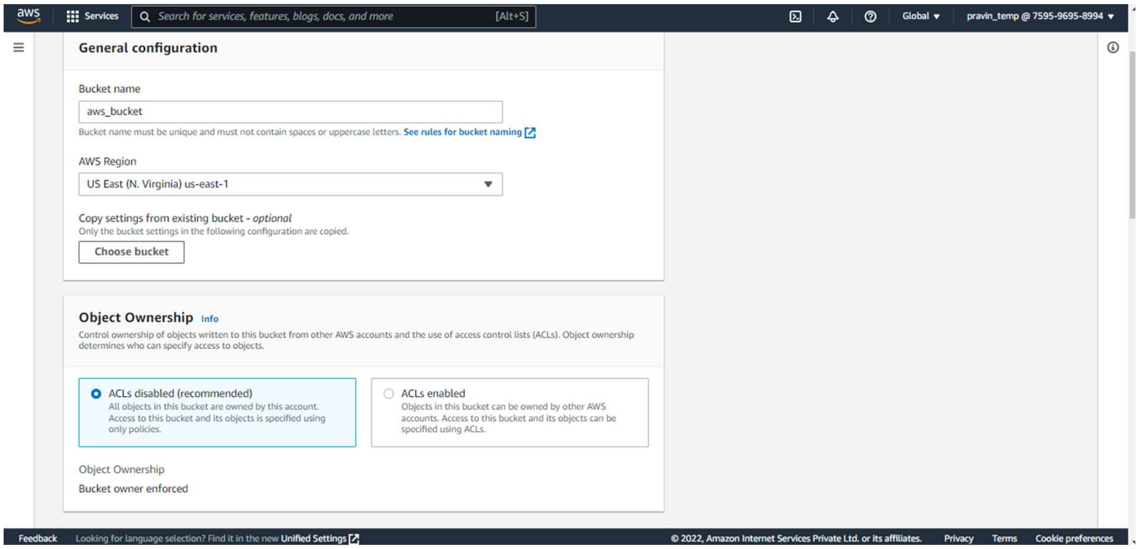
Buckets (1) Info

Buckets are containers for data stored in S3. Learn more

Name	AWS Region	Access	Creation date
laravel-dev-serverlessdeploymentbucket-4bitrio6uum7	US East (N. Virginia) us-east-1	Objects can be public	June 16, 2022, 17:47:07 (UTC+05:30)

3) Click on Create bucket Button

4) Fill the Bucket name


aws
Services
Search for services, features, blogs, docs, and more [Alt+S]
Global ▾
pravin_temp @ 7595-9695-8994 ▾
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Internet Services Private Ltd. or its affiliates.
Privacy
Terms
Cookie preferences

General configuration

Bucket name

aws_bucket

Bucket name must be unique and must not contain spaces or uppercase letters. See rules for bucket naming

AWS Region

US East (N. Virginia) us-east-1

Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

Choose bucket

Object Ownership Info

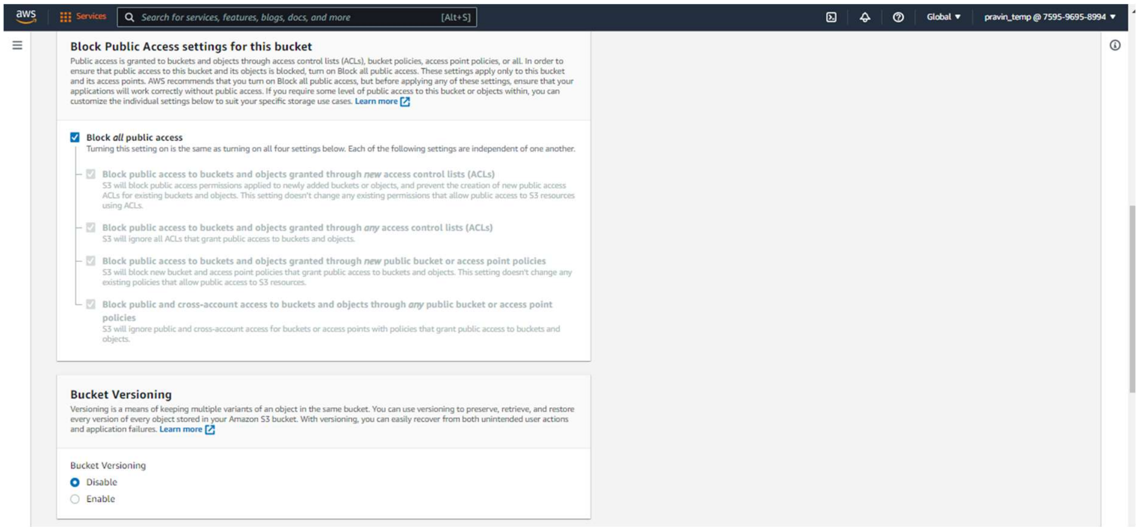
Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

● ACLs disabled (recommended)
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

○ ACLs enabled
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

Object Ownership

Bucket owner enforced


aws
Services
Search for services, features, blogs, docs, and more [Alt+S]
Global ▾
pravin_temp @ 7595-9695-8994 ▾

Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more

☑ **Block *all* public access**
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

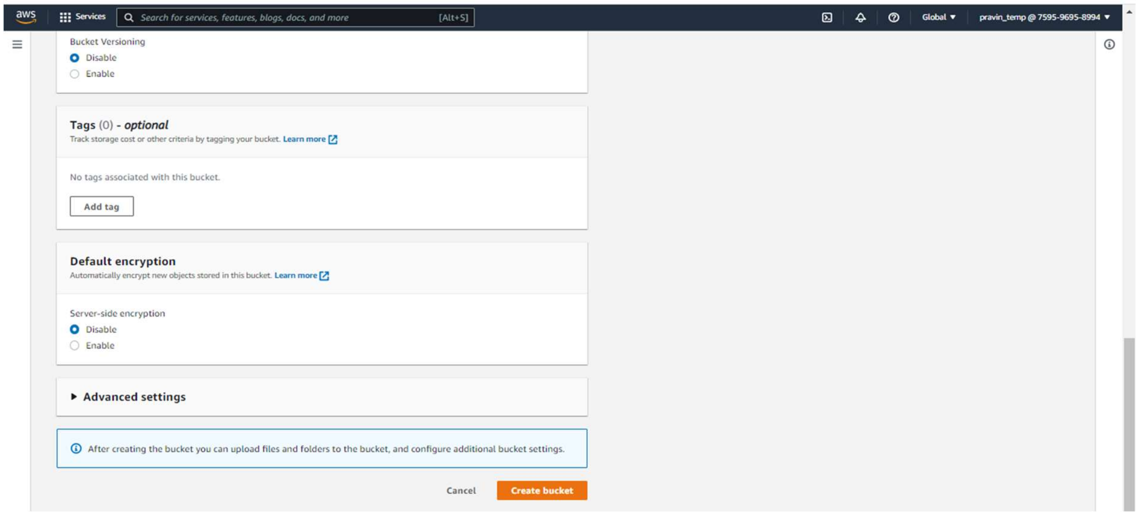
- ☑ **Block public access to buckets and objects granted through *new* access control lists (ACLs)**
  S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- ☑ **Block public access to buckets and objects granted through *any* access control lists (ACLs)**
  S3 will ignore all ACLs that grant public access to buckets and objects.
- ☑ **Block public access to buckets and objects granted through *new* public bucket or access point policies**
  S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- ☑ **Block public and cross-account access to buckets and objects through *any* public bucket or access point policies**
  S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. Learn more

Bucket Versioning

● Disable
○ Enable


aws
Services
Search for services, features, blogs, docs, and more [Alt+S]
Global ▾
pravin_temp @ 7595-9695-8994 ▾

Bucket Versioning

● Disable
○ Enable

Tags (0) - ***optional***

Track storage cost or other criteria by tagging your bucket. Learn more

No tags associated with this bucket.

Add tag

Default encryption

Automatically encrypt new objects stored in this bucket. Learn more

Server-side encryption

● Disable
○ Enable

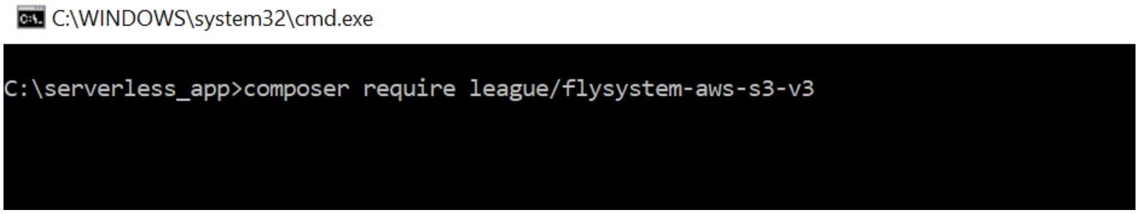
▸ **Advanced settings**

ⓘ After creating the bucket you can upload files and folders to the bucket, and configure additional bucket settings.

Cancel Create bucket

5) Click on Create bucket button

6) After Creating Bucket then install the Laravel AWS filsystem



C:\WINDOWS\system32\cmd.exe

```
C:\serverless_app>composer require league/flysystem-aws-s3-v3
```

7) configure to .env file

AWS_ACCESS_KEY_ID= get From .csv file

AWS_SECRET_ACCESS_KEY= get From .csv file

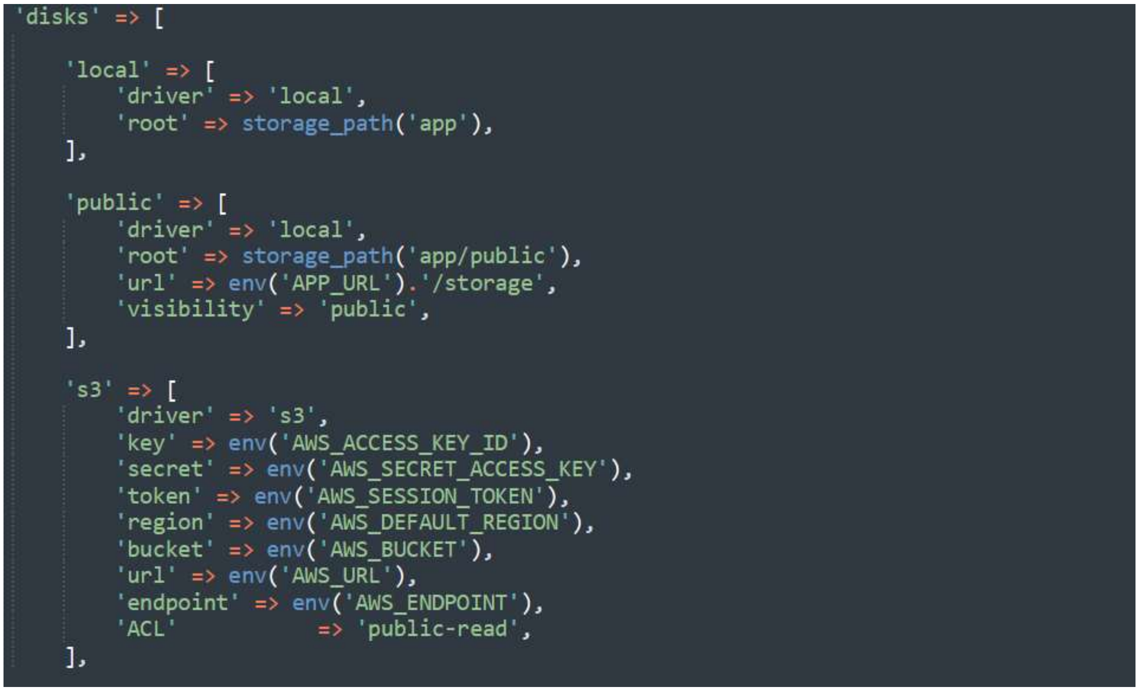
AWS_DEFAULT_REGION= bucket created region (us-east-1)

AWS_BUCKET=bucket name(donotcarry)

AWS_URL=https://donotcarry.s3.us-east-1.amazonaws.com

1) Bucket name with s3 -> https://donotcarry.s3
2) Region -> us-east-1

8) Configure code in Config folder-> filesystems.php



```
'disks' => [

    'local' => [
        'driver' => 'local',
        'root' => storage_path('app'),
    ],

    'public' => [
        'driver' => 'local',
        'root' => storage_path('app/public'),
        'url' => env('APP_URL').'/storage',
        'visibility' => 'public',
    ],

    's3' => [
        'driver' => 's3',
        'key' => env('AWS_ACCESS_KEY_ID'),
        'secret' => env('AWS_SECRET_ACCESS_KEY'),
        'token' => env('AWS_SESSION_TOKEN'),
        'region' => env('AWS_DEFAULT_REGION'),
        'bucket' => env('AWS_BUCKET'),
        'url' => env('AWS_URL'),
        'endpoint' => env('AWS_ENDPOINT'),
        'ACL'           => 'public-read',
    ],
```

Follow the Documentation
https://docs.aws.amazon.com/AmazonS3/latest/userguide/UsingBucket.html