➤ *Definition of WordPress & Amazon RDS*

➤ *Configure MySQL Server using Amazon RDS Service*

➤ *Create an AWS EC2 Instance*

➤ *Configure the Instance with Apache Webserver*

➤ *Download php application name "WordPress"*

➤ *Provide Endpoint/Connection string of Amazon RDS to the WordPress application to store data there.*

Let's Start…!!👩‍💻👩‍💻

WordPress

- WordPress is a free and open-source content management service used to build and maintain websites. Its ease of use and unique blogging features have helped it to become the most popular blogging tool on the web.

- WordPress provides a web-based user interface for designing, publishing, and updating websites. Instead of writing HTML, you can simply choose one of many different website "templates" or "themes" that has a design you like.

- WordPress interface makes it easy for anyone without web development experience to create and publish a website. The built-in blogging tools provide a simple way to track individual posts, visitors, and user comments.

Amazon RDS

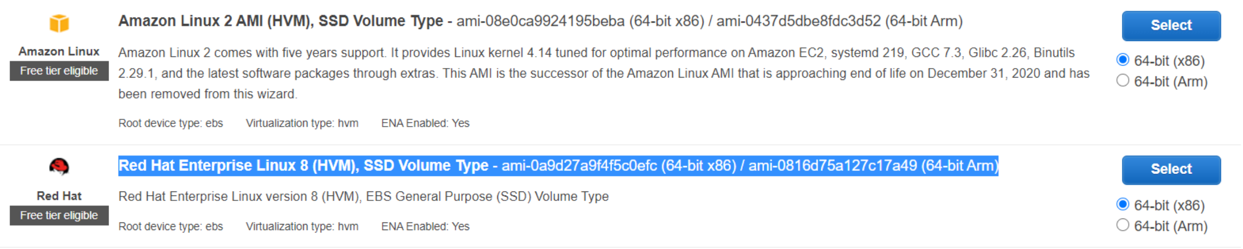
- Amazon Relational Database Service (Amazon RDS) is a web service that makes it easier to set up, operate, and scale a relational database in the AWS Cloud. It provides cost-efficient, resizable capacity for an industry-standard relational database and manages common database administration tasks.

- Amazon RDS manages backups, software patching, automatic failure detection, high availability, and recovery. It is a fully managed database service provided by amazon.

- With Amazon RDS we can use the database products you are already familiar with MySQL, MariaDB, PostgreSQL, Oracle, Microsoft SQL Server.

Step 1): Create an AWS EC2 Instance

First select an Amazon Machine Image (AMI) to launch EC2 instance for the installation of WordPress. I have used Red Hat Enterprise Linux 8(RHEL8); But you can use any image.



Amazon Linux
Free tier eligible

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-08e0ca9924195beba (64-bit x86) / ami-0437d5dbe8fdc3d52 (64-bit Arm)

Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux AMI that is approaching end of life on December 31, 2020 and has been removed from this wizard.

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

● 64-bit (x86)
○ 64-bit (Arm)

Red Hat
Free tier eligible

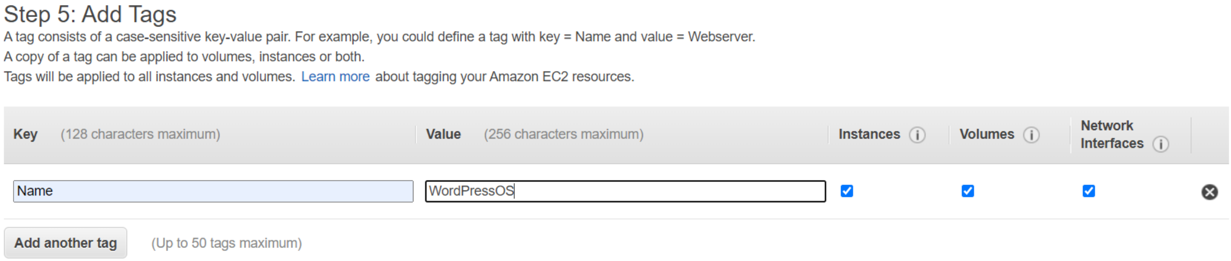
Red Hat Enterprise Linux 8 (HVM), SSD Volume Type - ami-0a9d27a9f4f5c0efc (64-bit x86) / ami-0816d75a127c17a49 (64-bit Arm)

Red Hat Enterprise Linux version 8 (HVM), EBS General Purpose (SSD) Volume Type

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

● 64-bit (x86)
○ 64-bit (Arm)

- Add any tag for your instance, i.e., WordPressOS



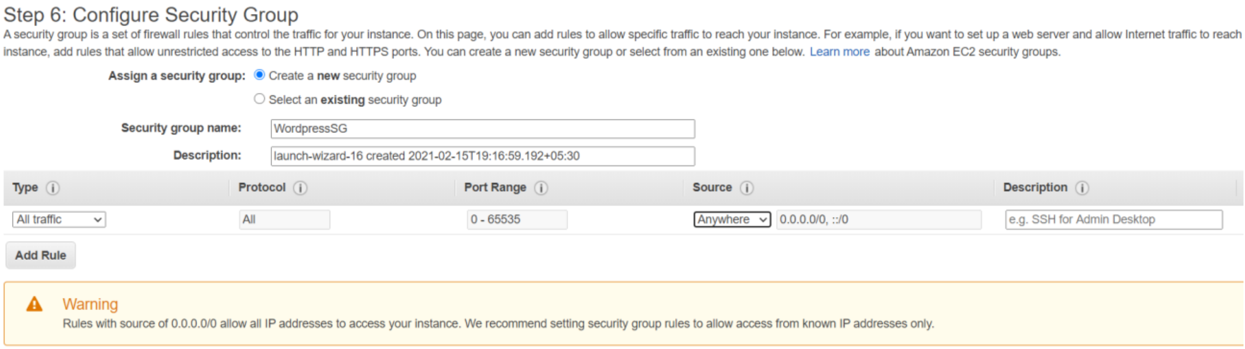
Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.
A copy of a tag can be applied to volumes, instances or both.
Tags will be applied to all instances and volumes. Learn more about tagging your Amazon EC2 resources.

Key (128 characters maximum)	Value (256 characters maximum)	Instances	Volumes	Network Interfaces	
Name	WordPressOS	☑	☑	☑	⊗

(Up to 50 tags maximum)

- Create a Security group for your EC2 instance, in your Security group allow all traffic in inbound rules. i.e., WordpressSG



Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

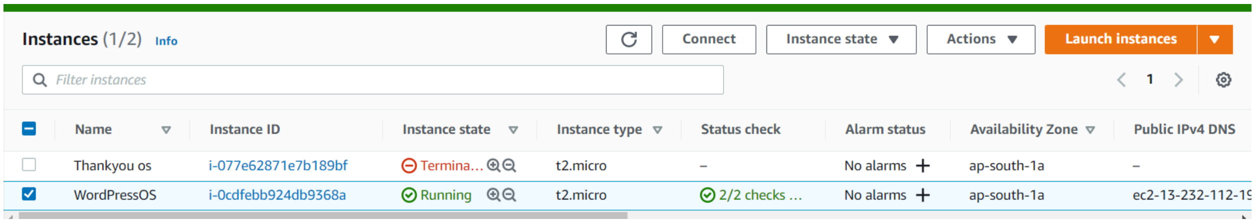
Assign a security group: ● Create a **new** security group
○ Select an **existing** security group

Security group name: WordpressSG
Description: launch-wizard-16 created 2021-02-15T19:16:59.192+05:30

Type	Protocol	Port Range	Source		Description
All traffic	All	0 - 65535	Anywhere	0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop

Warning
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

- Finally, Instance is launched.



Instances (1/2) Info

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
□	Thankyou os	i-077e62871e7b189bf	Termina...	t2.micro	–	No alarms	ap-south-1a	–
☑	WordPressOS	i-0cdfebb924db9368a	Running	t2.micro	2/2 checks ...	No alarms	ap-south-1a	ec2-13-232-112-19

Step 2): Install php, wget, httpd & MySQL.

To run WordPress, we have some requirements: -

- PHP version 7.4 or greater.
- MySQL version 5.6 or greater OR MariaDB version 10.1 or greater.

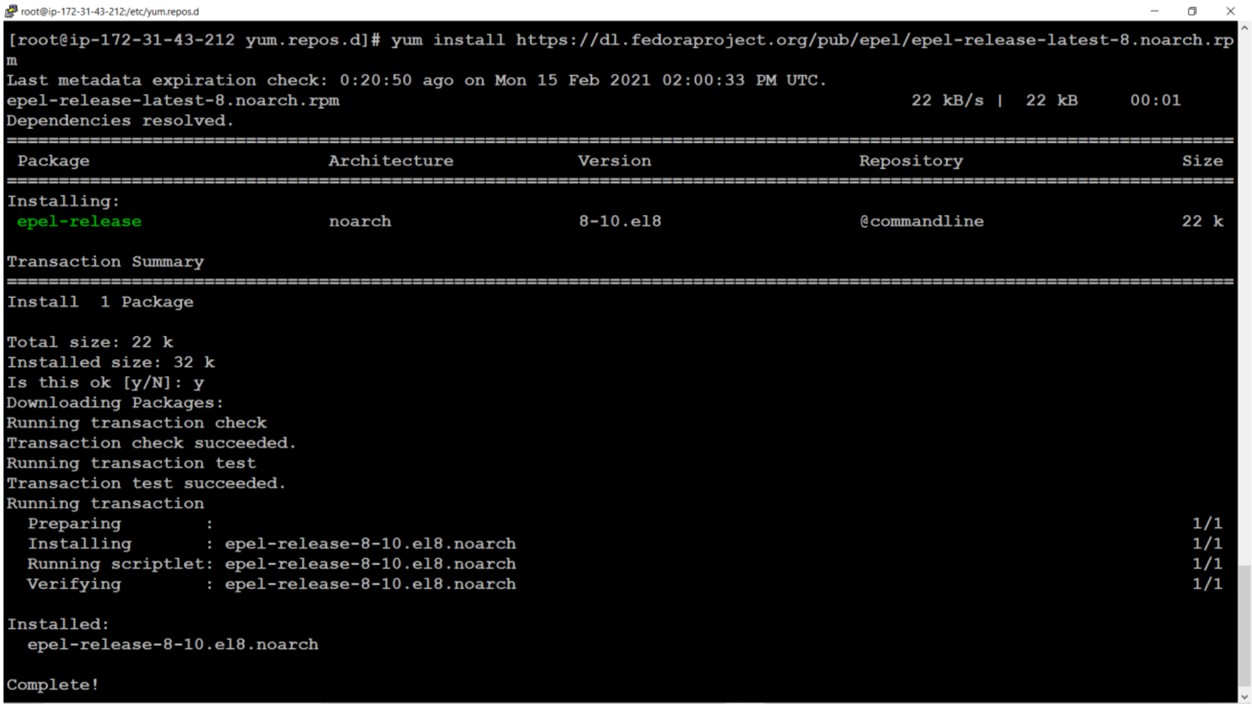
PHP 7.x packages are available in several different repositories. We will use the Remi repository which provides newer versions of various software packages including PHP. The Remi repository depends on the EPEL repository.

- First login inside instance via putty software and Install EPEL & Remi Repository.
- To install EPEL Repository

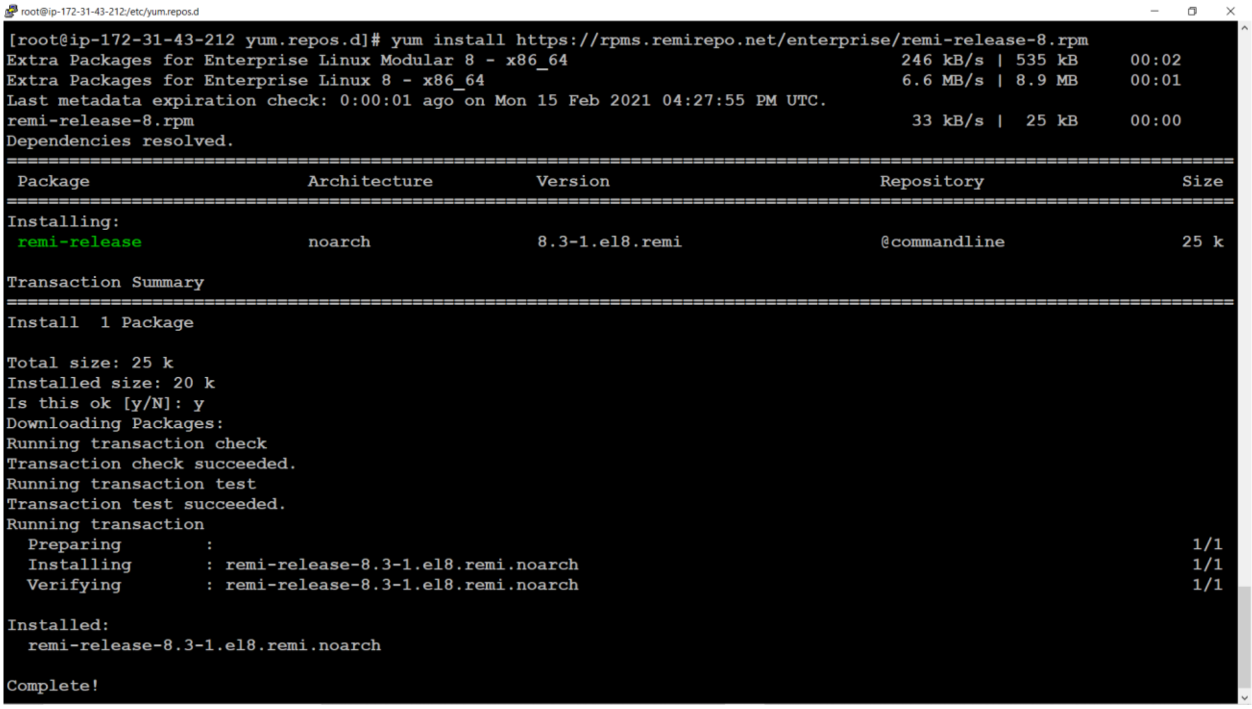
  Yum install https://dl.fedoraproject.org/pub/epel/epel-release- latest-8.noarch.rpm

- To install Remi Repository

  Yum install https://rpms.remirepo.net/enterprise/remi-release-8.rpm



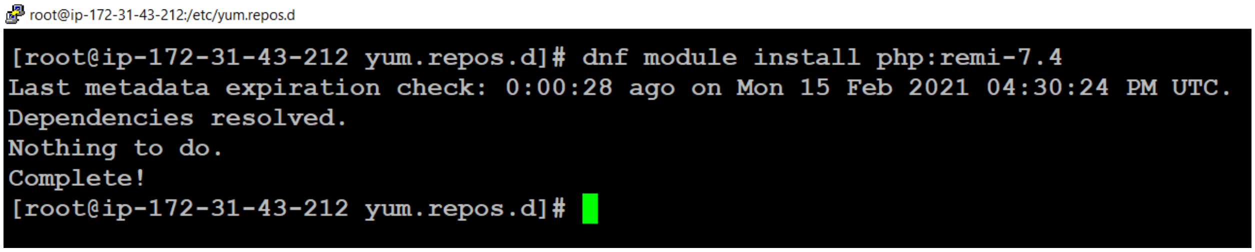
```
[root@ip-172-31-43-212 yum.repos.d]# yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rp
m
Last metadata expiration check: 0:20:50 ago on Mon 15 Feb 2021 02:00:33 PM UTC.
epel-release-latest-8.noarch.rpm                                                       22 kB/s |  22 kB     00:01
Dependencies resolved.
======================================================================================================================
 Package                       Architecture            Version                    Repository                     Size
======================================================================================================================
Installing:
 epel-release                  noarch                  8-10.el8                   @commandline                   22 k

Transaction Summary
======================================================================================================================
Install  1 Package

Total size: 22 k
Installed size: 32 k
Is this ok [y/N]: y
Downloading Packages:
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                                                              1/1
  Installing       : epel-release-8-10.el8.noarch                                                                 1/1
  Running scriptlet: epel-release-8-10.el8.noarch                                                                 1/1
  Verifying        : epel-release-8-10.el8.noarch                                                                 1/1

Installed:
  epel-release-8-10.el8.noarch

Complete!
```



```
[root@ip-172-31-43-212 yum.repos.d]# yum install https://rpms.remirepo.net/enterprise/remi-release-8.rpm
Extra Packages for Enterprise Linux Modular 8 - x86_64                                246 kB/s | 535 kB     00:02
Extra Packages for Enterprise Linux 8 - x86_64                                        6.6 MB/s | 8.9 MB     00:01
Last metadata expiration check: 0:00:01 ago on Mon 15 Feb 2021 04:27:55 PM UTC.
remi-release-8.rpm                                                                     33 kB/s |  25 kB     00:00
Dependencies resolved.
======================================================================================================================
 Package                     Architecture          Version                          Repository                   Size
======================================================================================================================
Installing:
 remi-release                noarch                8.3-1.el8.remi                   @commandline                 25 k

Transaction Summary
======================================================================================================================
Install  1 Package

Total size: 25 k
Installed size: 20 k
Is this ok [y/N]: y
Downloading Packages:
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                                                              1/1
  Installing       : remi-release-8.3-1.el8.remi.noarch                                                           1/1
  Verifying        : remi-release-8.3-1.el8.remi.noarch                                                           1/1

Installed:
  remi-release-8.3-1.el8.remi.noarch

Complete!
```

- Install php 7.4 & php-mysqlnd.

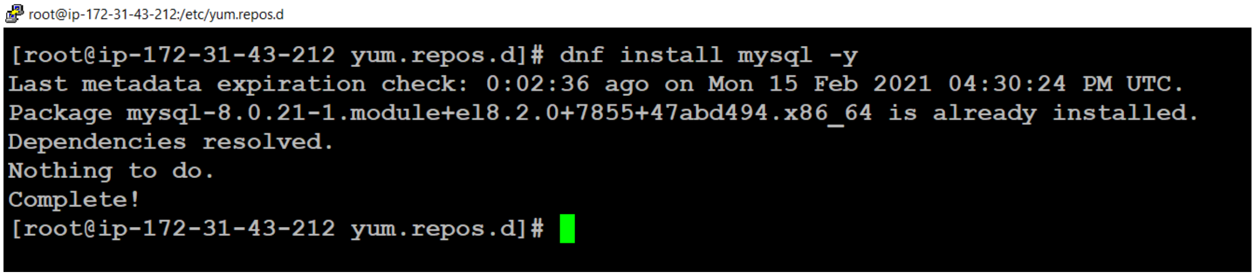
```
dnf module install php:remi-7.4
```



```
[root@ip-172-31-43-212 yum.repos.d]# dnf module install php:remi-7.4
Last metadata expiration check: 0:00:28 ago on Mon 15 Feb 2021 04:30:24 PM UTC.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-43-212 yum.repos.d]#
```

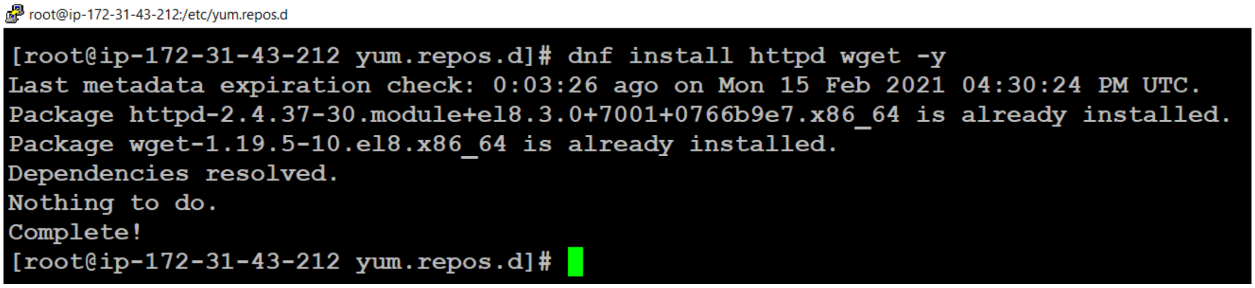
- Install mysql, wget & httpd.

dnf install mysql -y



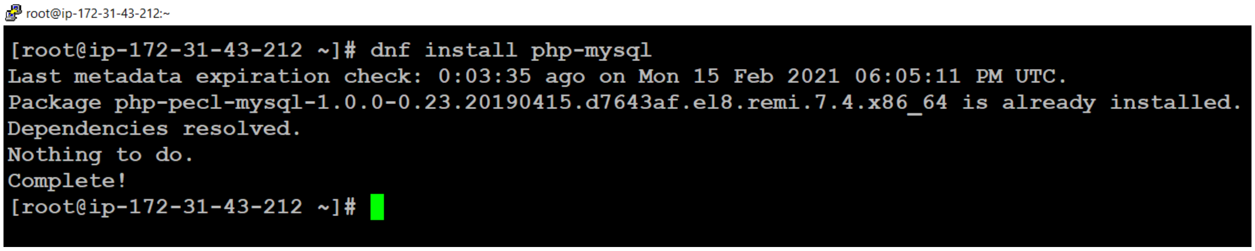
```
[root@ip-172-31-43-212 yum.repos.d]# dnf install mysql -y
Last metadata expiration check: 0:02:36 ago on Mon 15 Feb 2021 04:30:24 PM UTC.
Package mysql-8.0.21-1.module+el8.2.0+7855+47abd494.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-43-212 yum.repos.d]#
```

dnf install httpd wget -y



```
[root@ip-172-31-43-212 yum.repos.d]# dnf install httpd wget -y
Last metadata expiration check: 0:03:26 ago on Mon 15 Feb 2021 04:30:24 PM UTC.
Package httpd-2.4.37-30.module+el8.3.0+7001+0766b9e7.x86_64 is already installed.
Package wget-1.19.5-10.el8.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-43-212 yum.repos.d]#
```

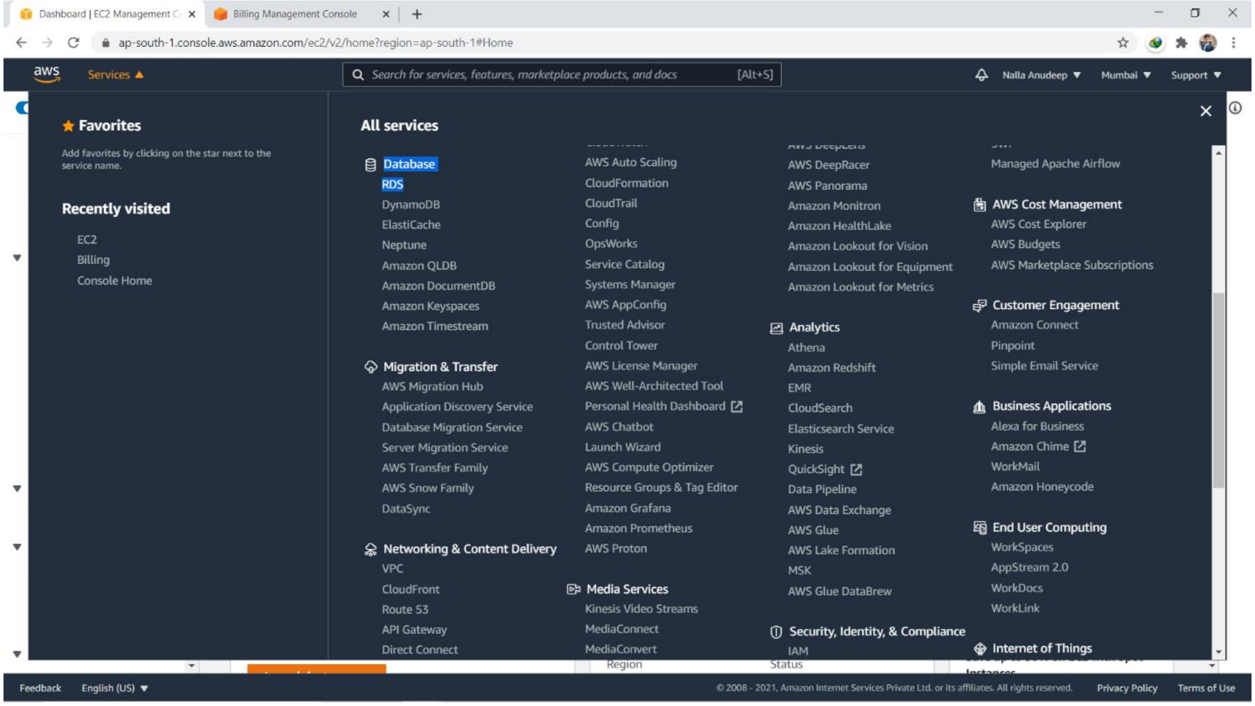
dnf install php-mysql



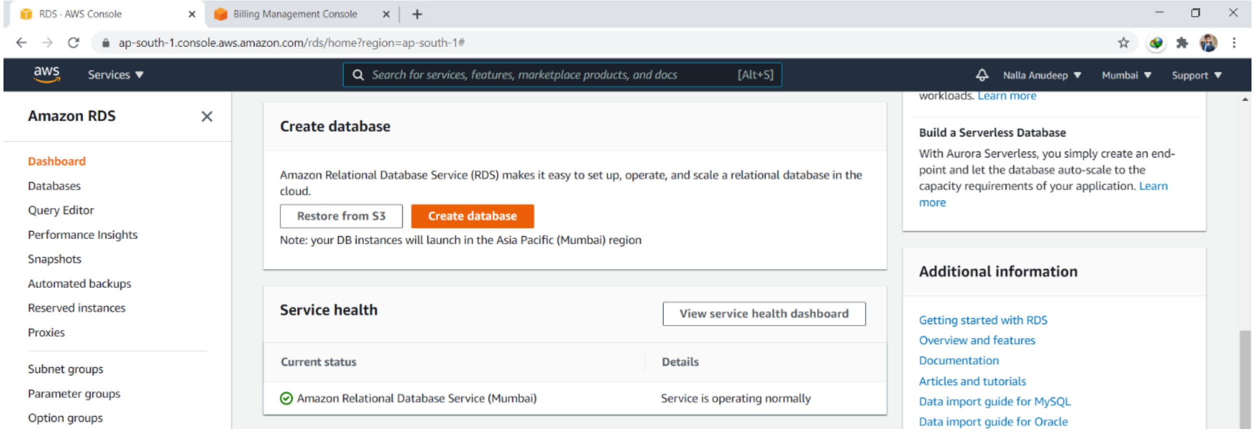
```
[root@ip-172-31-43-212 ~]# dnf install php-mysql
Last metadata expiration check: 0:03:35 ago on Mon 15 Feb 2021 06:05:11 PM UTC.
Package php-pecl-mysql-1.0.0-0.23.20190415.d7643af.el8.remi.7.4.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-43-212 ~]#
```

Step 3): Configure MySQL Server using Amazon RDS Service

- Go into Services section then select RDS under Database.



- Click on Create database



workloads. Learn more

Create database

Amazon Relational Database Service (RDS) makes it easy to set up, operate, and scale a relational database in the cloud.

Note: your DB instances will launch in the Asia Pacific (Mumbai) region

Build a Serverless Database

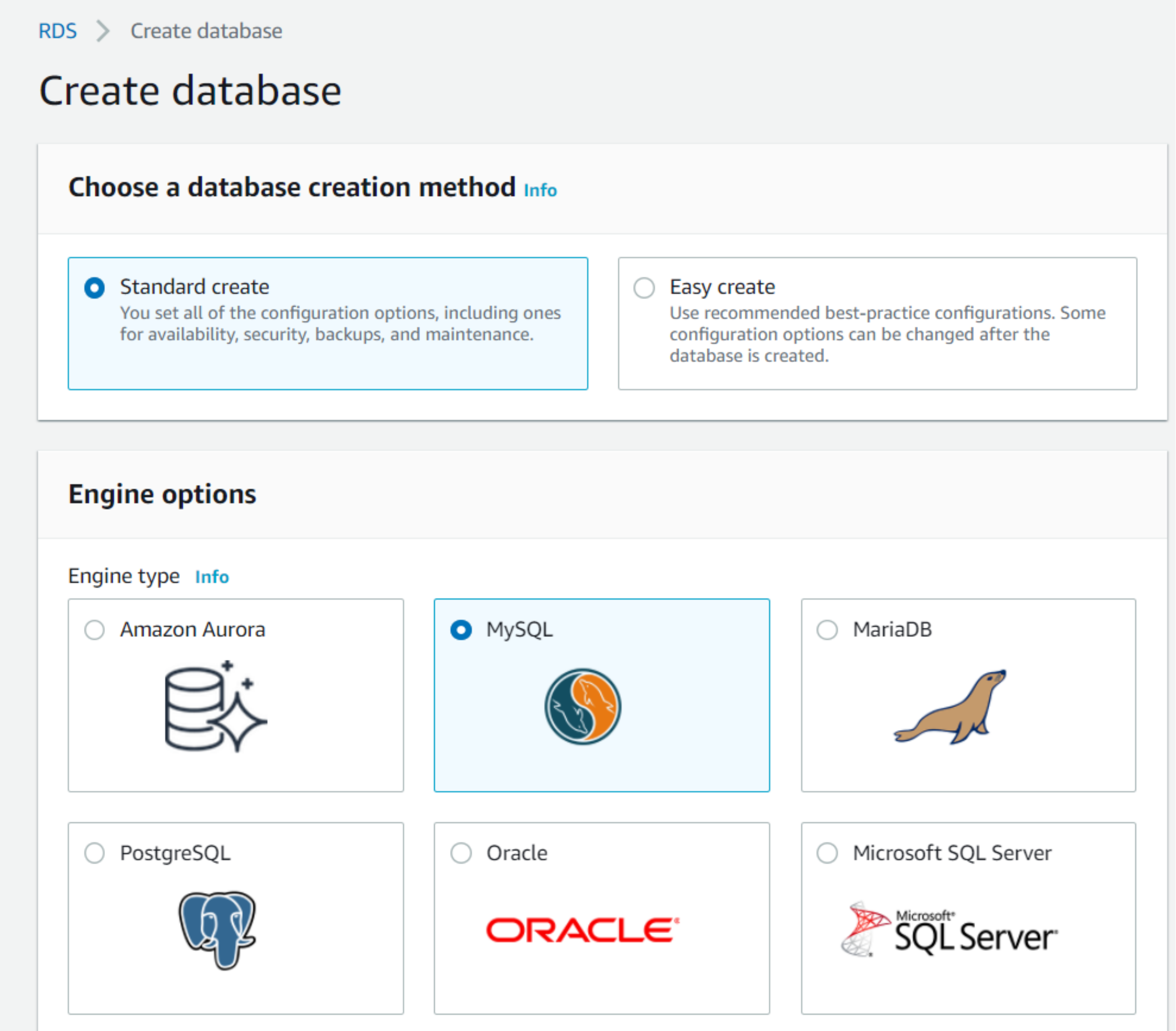
With Aurora Serverless, you simply create an end-point and let the database auto-scale to the capacity requirements of your application. Learn more

Service health

Current status	Details
✓ Amazon Relational Database Service (Mumbai)	Service is operating normally

Additional information

- Select Standard create then MySQL.



RDS > Create database

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

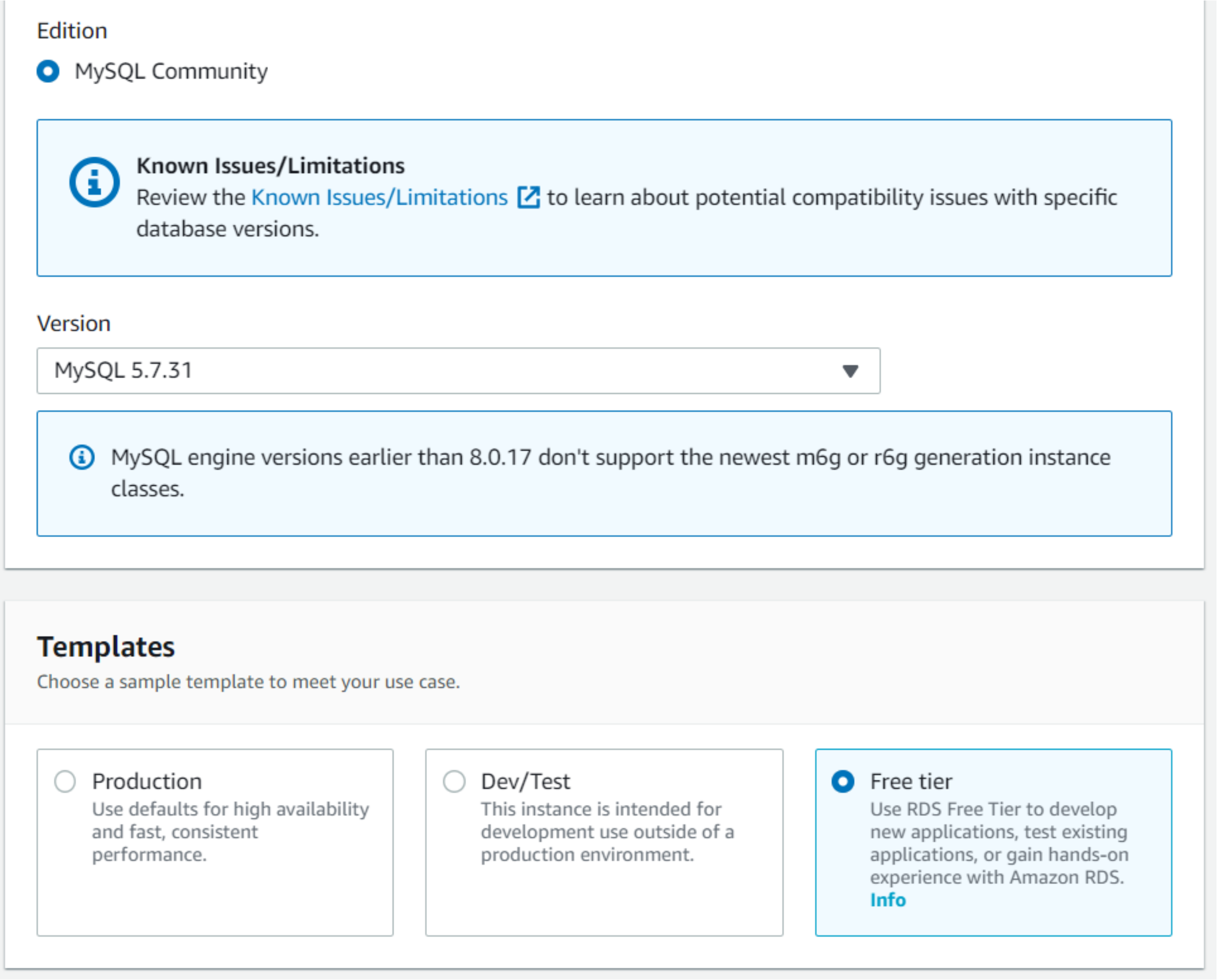
○ MariaDB

○ PostgreSQL

○ Oracle

○ Microsoft SQL Server

- Choose Version: MySQL 5.7.31 & Free Tier. Our purpose is just to test the database, that is why we are choosing free tier option.



Edition

● MySQL Community

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

Version

MySQL 5.7.31

MySQL engine versions earlier than 8.0.17 don't support the newest m6g or r6g generation instance classes.

Templates

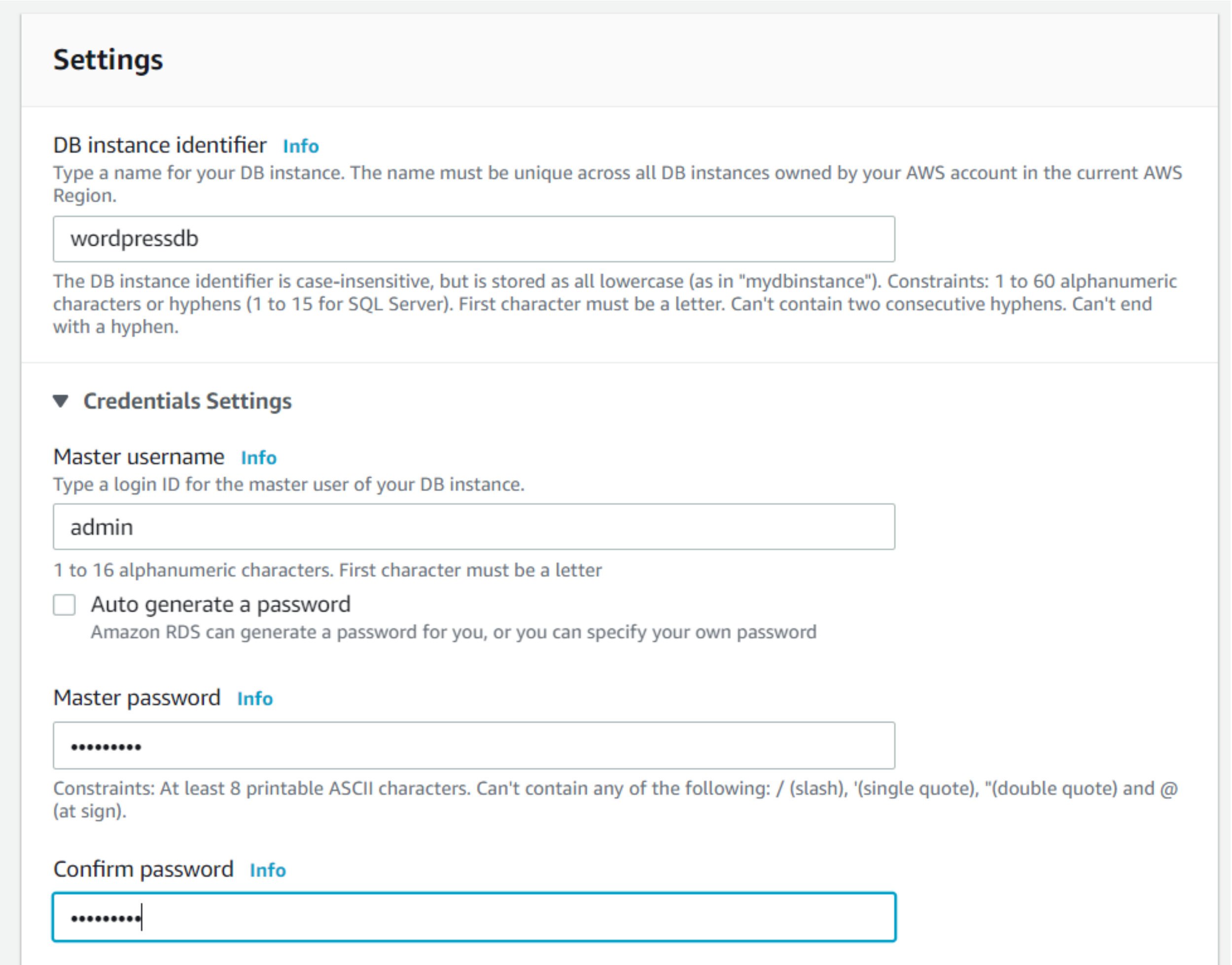
Choose a sample template to meet your use case.

○ 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

- First set your database (DB) instance name & then username & password for your database.



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.

wordpressdb

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens (1 to 15 for SQL Server). 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

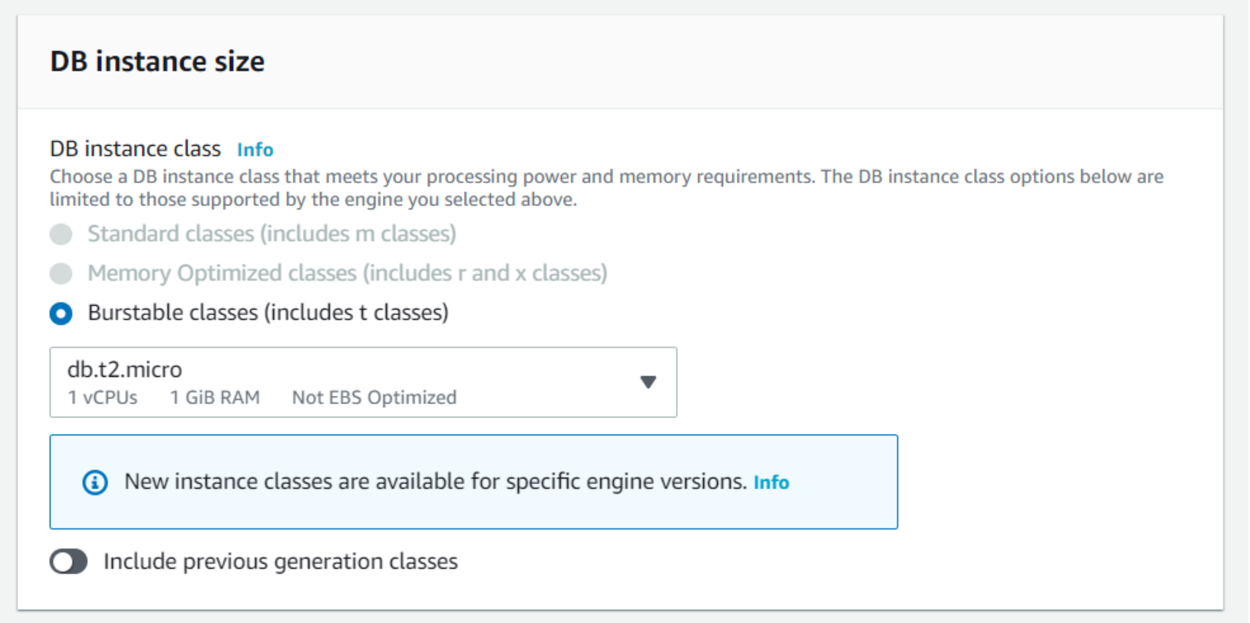
•••••••••

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

•••••••••

- In free tier, we have only option of db.t2.micro having limited resources i.e. 1 GiB RAM & 1 vCPU.



DB instance size

DB instance class Info

Choose a DB instance class that meets your processing power and memory requirements. The DB instance class options below are limited to those supported by the engine you selected above.

○ Standard classes (includes m classes)

○ Memory Optimized classes (includes r and x classes)

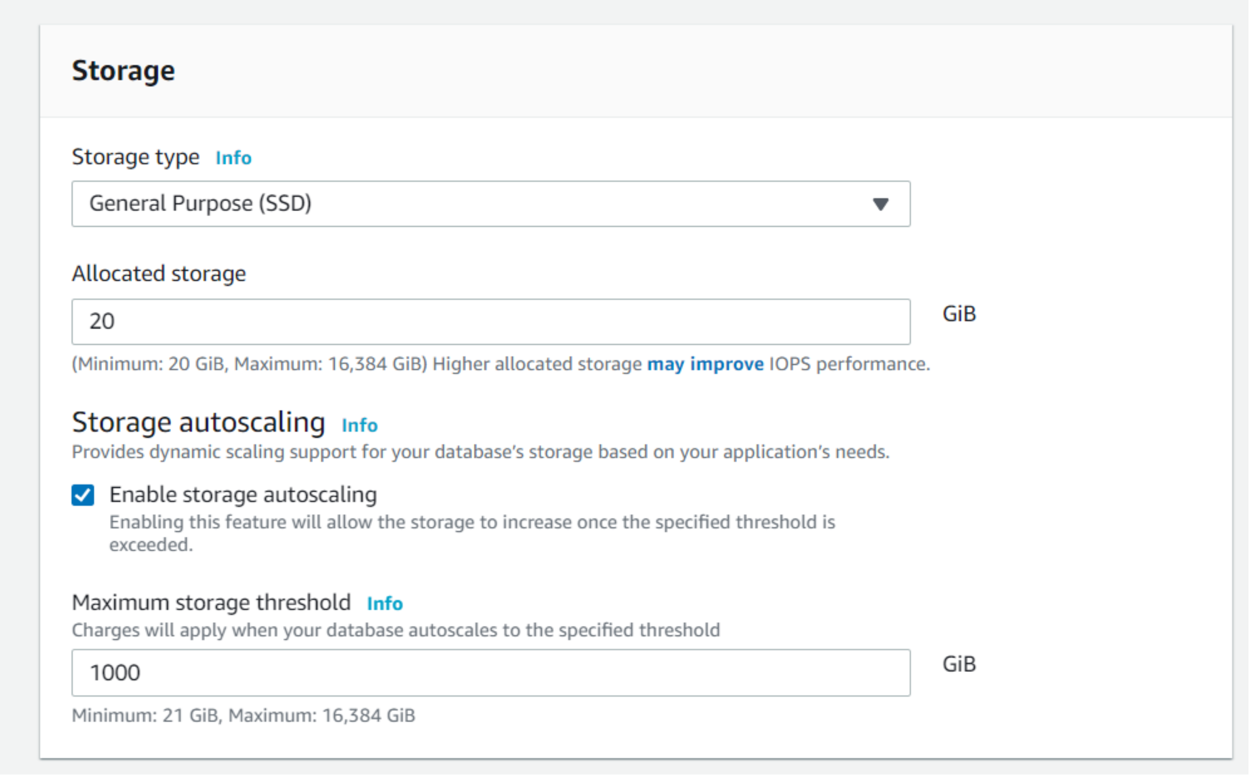
● Burstable classes (includes t classes)

db.t2.micro
1 vCPUs 1 GiB RAM Not EBS Optimized

New instance classes are available for specific engine versions. Info

Include previous generation classes

- By default, we get 20 GiB storage, we can extend it up to 1000 GiB. But 20 GiB is more than enough for our testing.



Storage

Storage type Info

General Purpose (SSD)

Allocated storage

20 GiB

(Minimum: 20 GiB, Maximum: 16,384 GiB) Higher allocated storage **may 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 once the specified threshold is exceeded.

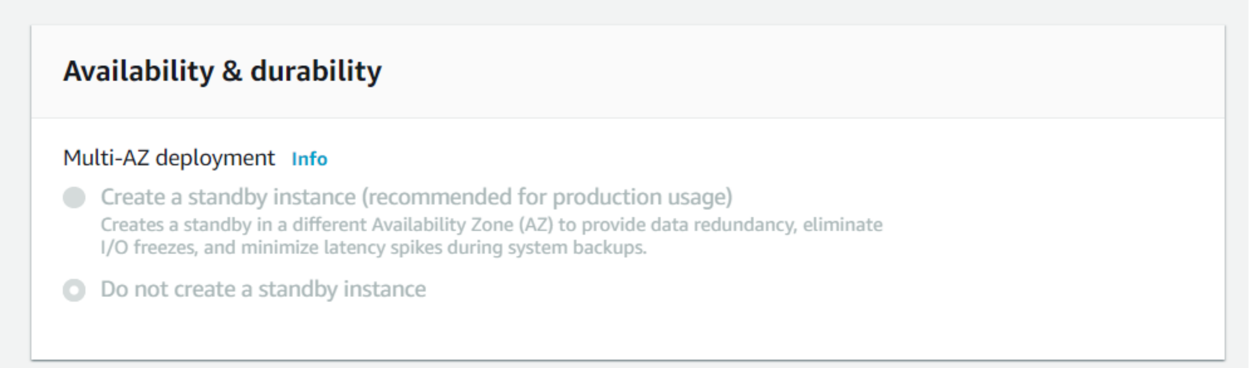
Maximum storage threshold Info

Charges will apply when your database autoscales to the specified threshold

1000 GiB

Minimum: 21 GiB, Maximum: 16,384 GiB

- Amazon also provides a facility of Availability & durability, but it is not available in free tier option.



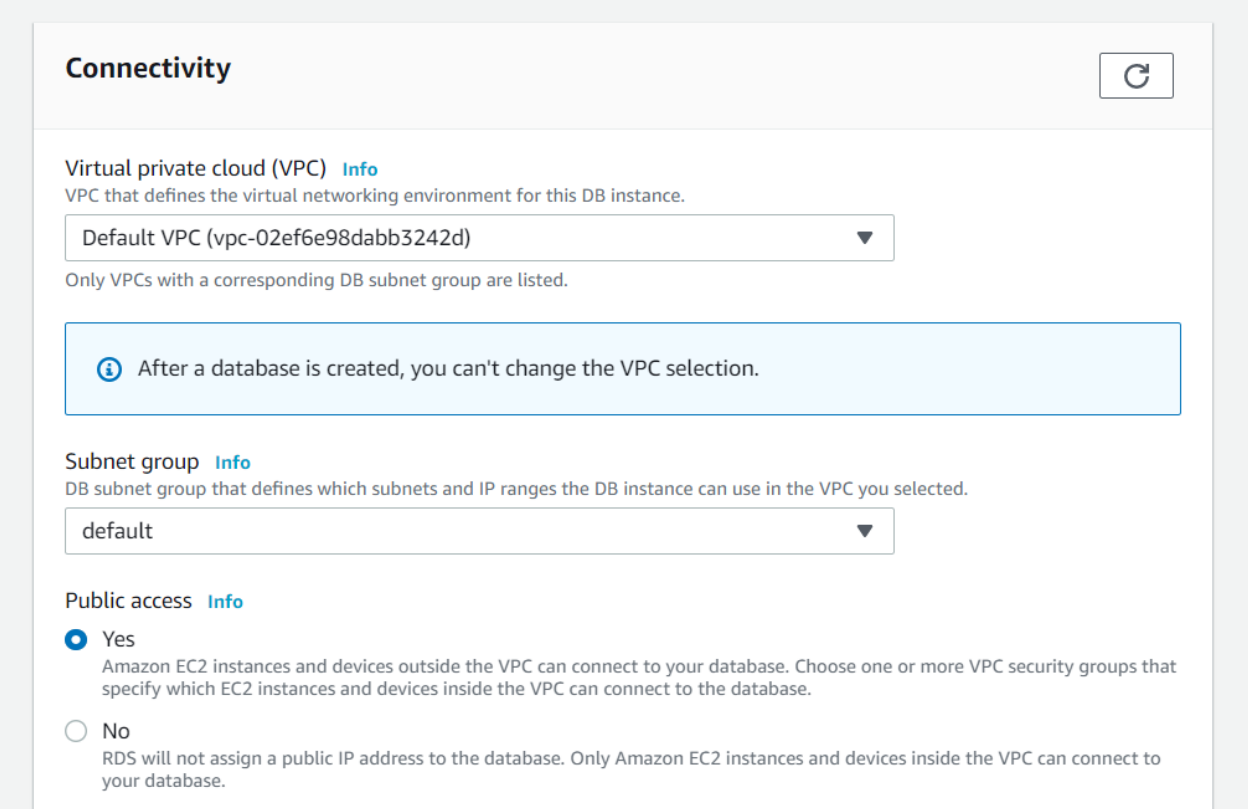
Availability & durability

Multi-AZ deployment Info

○ Create a standby instance (recommended for production usage)
Creates a standby in a different Availability Zone (AZ) to provide data redundancy, eliminate I/O freezes, and minimize latency spikes during system backups.

● Do not create a standby instance

- Select VPC and Subnet group according to your choice. It is not a good practice to give public access but for testing we need public access (Yes).



Connectivity

Virtual private cloud (VPC) Info
VPC that defines the virtual networking environment for this DB instance.
Default VPC (vpc-02ef6e98dabb3242d) ▼
Only VPCs with a corresponding DB subnet group are listed.

ⓘ After a database is created, you can't change the VPC selection.

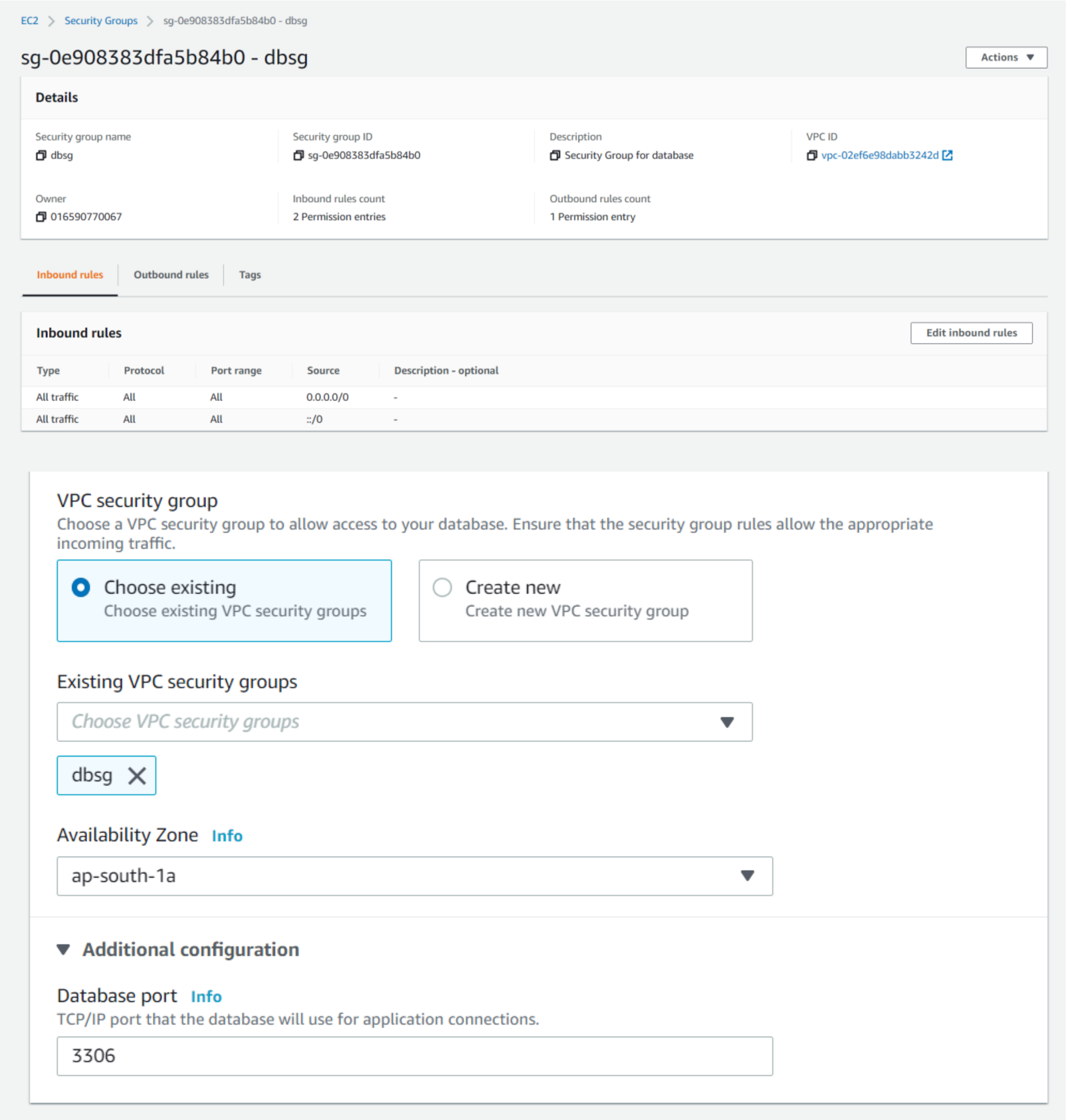
Subnet group Info
DB subnet group that defines which subnets and IP ranges the DB instance can use in the VPC you selected.
default ▼

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.

● Create a VPC security group for your database & allow all traffic in inbound rules. We will update later these inbound rules after creating the instance. I have created “dbsg” security group here. Choose any AZ from the list & keep Database port 3306.



EC2 > Security Groups > sg-0e908383dfa5b84b0 - dbsg

sg-0e908383dfa5b84b0 - dbsg

Details

Security group name dbsg	Security group ID sg-0e908383dfa5b84b0	Description Security Group for database	VPC ID vpc-02ef6e98dabb3242d
Owner 016590770067	Inbound rules count 2 Permission entries	Outbound rules count 1 Permission entry	

Inbound rules

Type	Protocol	Port range	Source	Description - optional
All traffic	All	All	0.0.0.0/0	-
All traffic	All	All	::/0	-

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 ▼

dbsg ✕

Availability Zone Info

ap-south-1a ▼

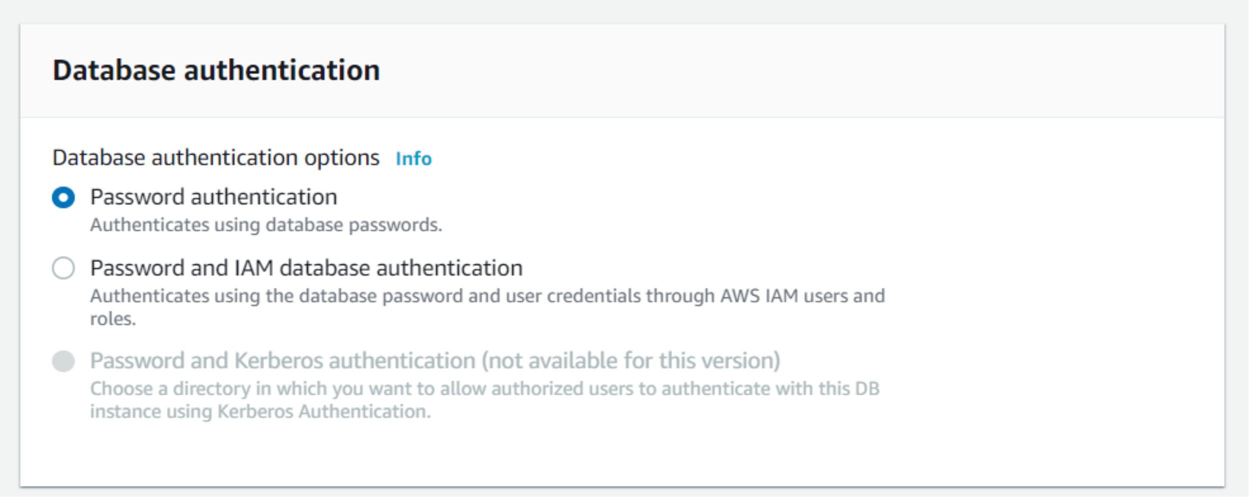
▼ Additional configuration

Database port Info

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

3306

- Select Password authentication option for your database.



Database authentication

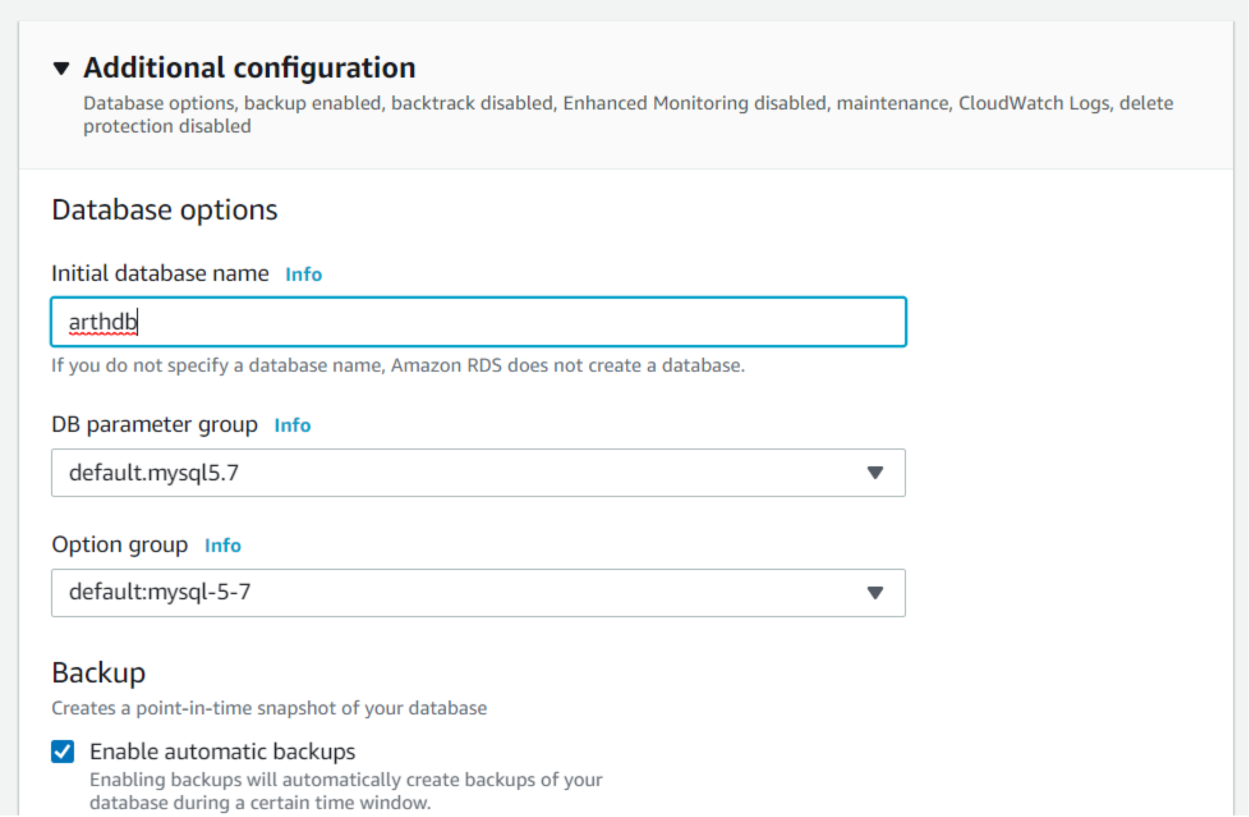
Database authentication options

● 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 (not available for this version)
Choose a directory in which you want to allow authorized users to authenticate with this DB instance using Kerberos Authentication.

- Under Additional configuration, set your database name & all settings leave as by default. You can enable Backup option if you want.



▼ Additional configuration
Database options, backup enabled, backtrack disabled, Enhanced Monitoring disabled, maintenance, CloudWatch Logs, delete protection disabled

Database options

Initial database name

arthdb

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

DB parameter group

default.mysql5.7 ▼

Option group

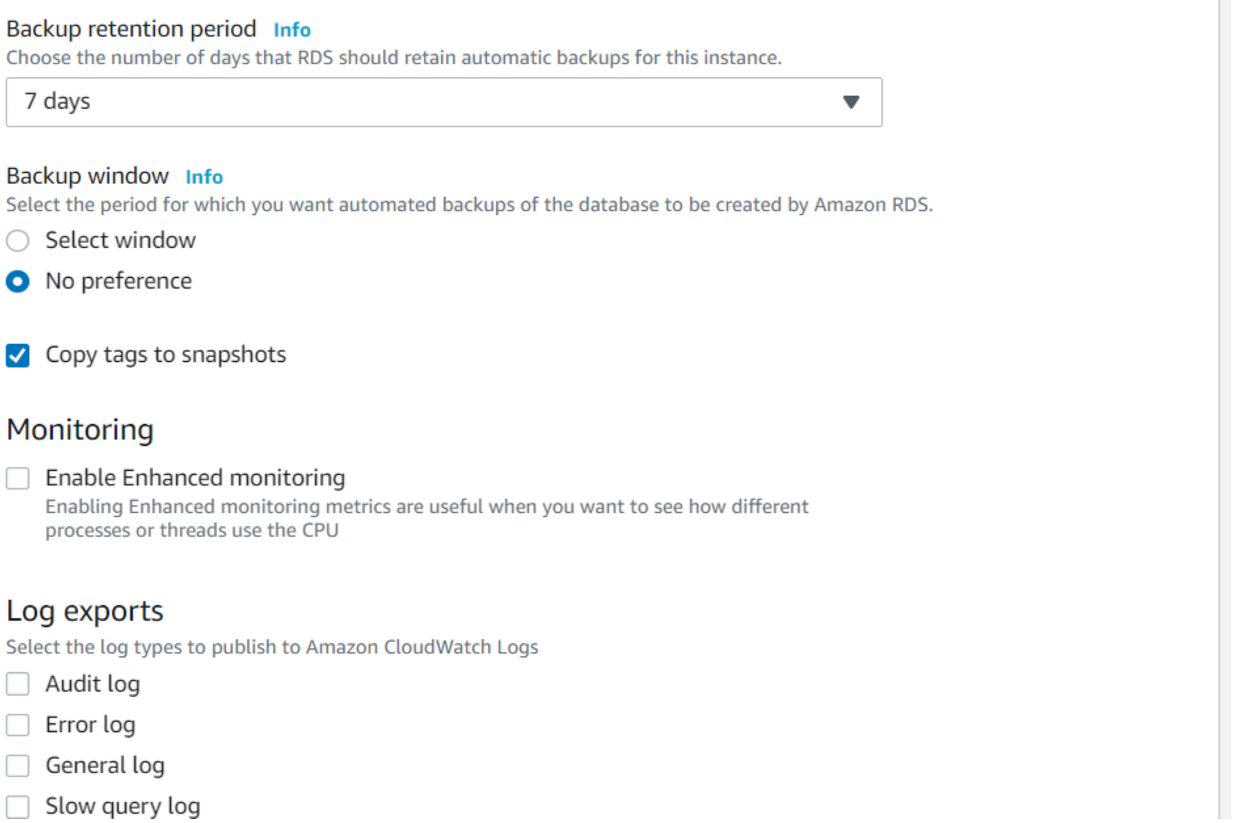
default:mysql-5-7 ▼

Backup
Creates a point-in-time snapshot of your database

☑ Enable automatic backups
Enabling backups will automatically create backups of your database during a certain time window.

- In Backup retention period, you can choose the number of days you want to keep your data as backup. Monitoring & Log exports are used to check the consumption of your resources (RAM/CPU etc.)

In Backup window you can set the time when you want to get the back up of your data to avoid any impact on performance by selecting Select window option.



Backup retention period

Choose the number of days that RDS should retain automatic backups for this instance.

7 days ▼

Backup window

Select the period for which you want automated backups of the database to be created by Amazon RDS.

○ Select window

● No preference

☑ Copy tags to snapshots

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

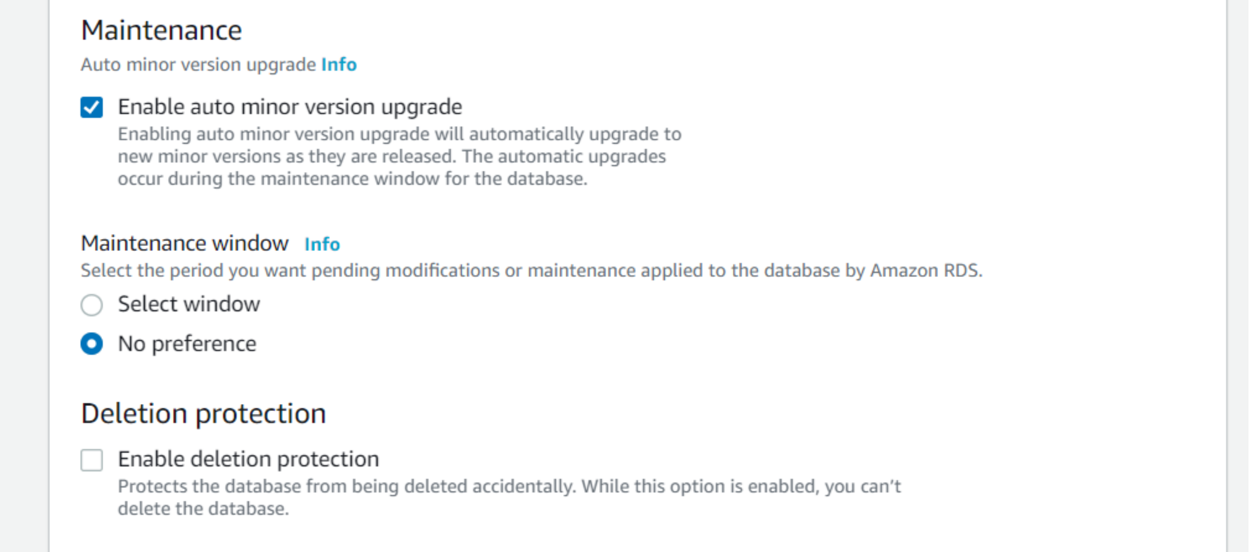
□ Audit log

□ Error log

□ General log

□ Slow query log

- Similarly, we can set the time for Maintenance by choosing Select window option. To protect the database from being deleted accidentally you must enable Deletion protection option.



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.

○ Select window

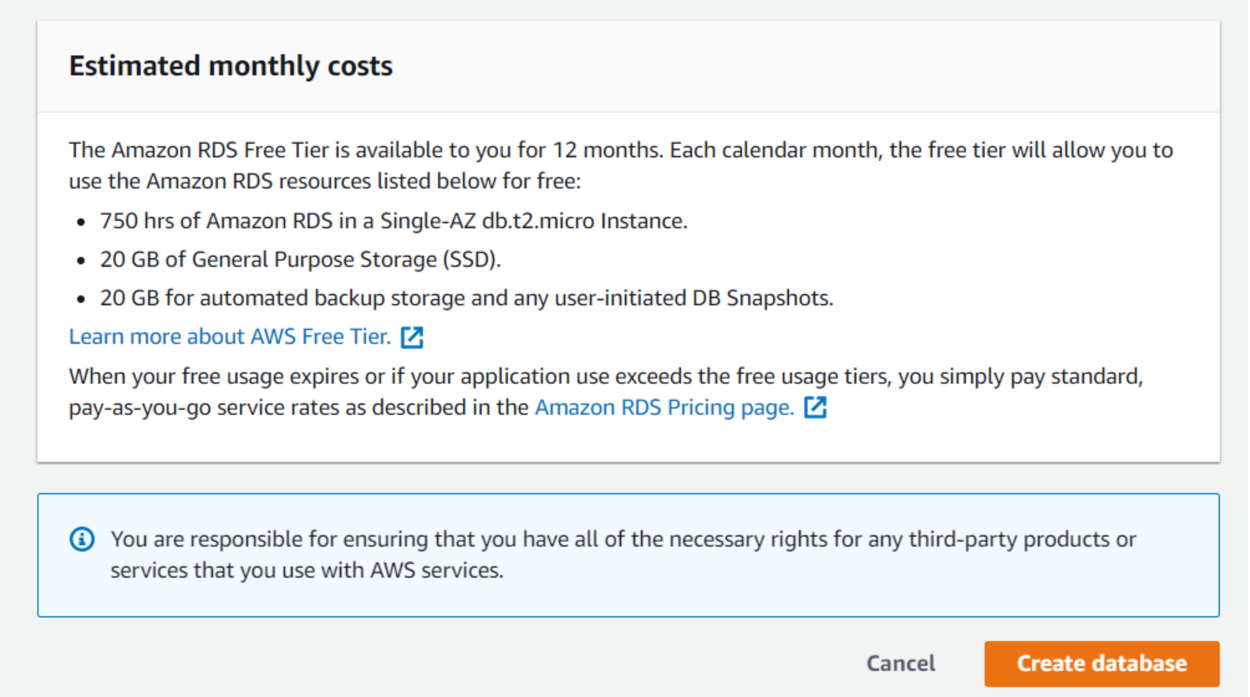
● No preference

Deletion protection

□ Enable deletion protection

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

- Now, we have to click on Create 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:

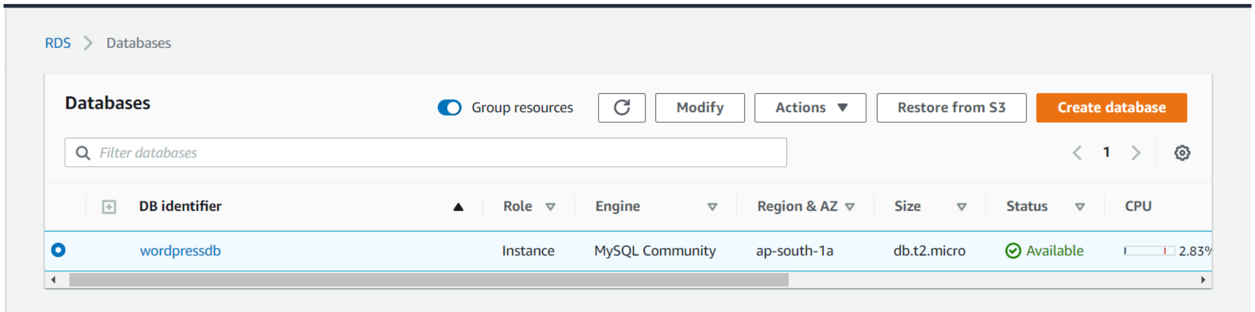
- 750 hrs of Amazon RDS in a Single-AZ db.t2.micro Instance.
- 20 GB of General Purpose Storage (SSD).
- 20 GB for automated backup storage and any user-initiated DB Snapshots.

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.

- It will take some time depends on your network speed to show Status “Available”.

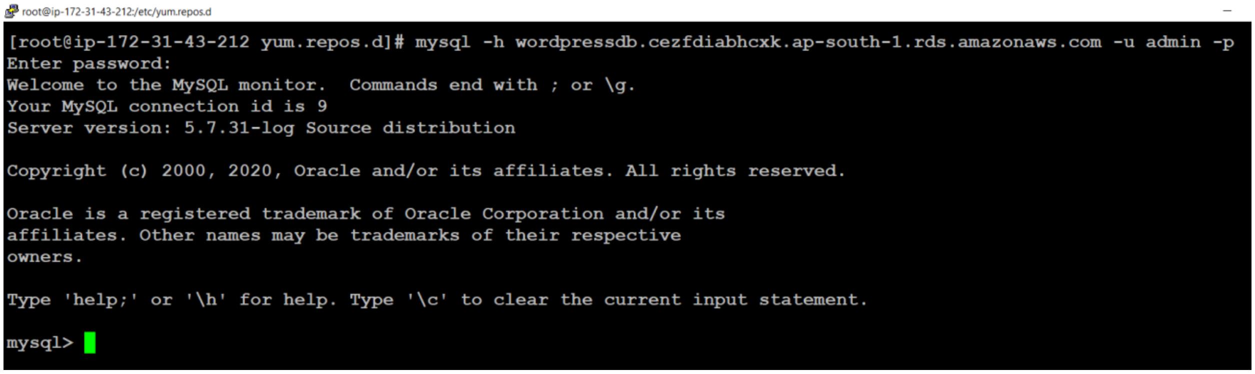


DB identifier	Role	Engine	Region & AZ	Size	Status	CPU
wordpressdb	Instance	MySQL Community	ap-south-1a	db.t2.micro	✓ Available	2.83%

- Finally, our database has been created successfully.

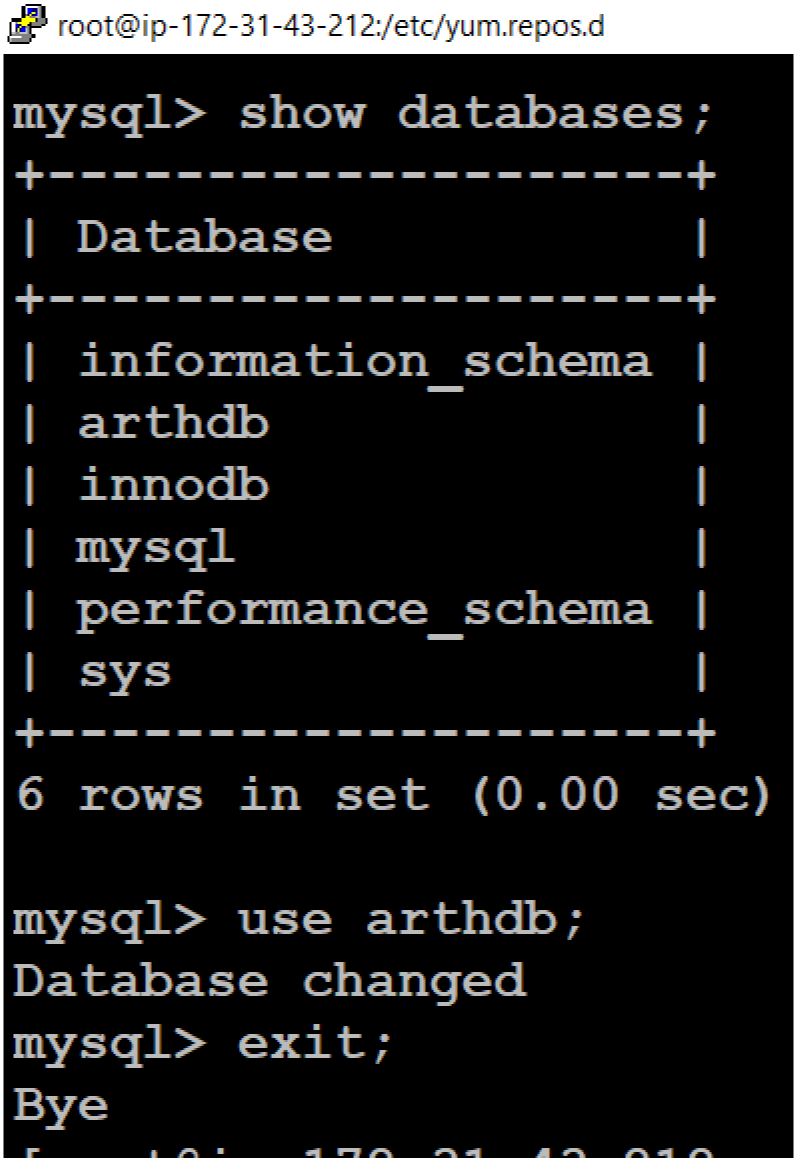
Step 4): Connect to WordpressOS

- Login inside your mysql database, use command:
  mysql -h <db_endpoint_url> -u <user_name> -p



```
[root@ip-172-31-43-212 yum.repos.d]# mysql -h wordpressdb.cezfdiabhcxk.ap-south-1.rds.amazonaws.com -u admin -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 5.7.31-log Source distribution

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> 
```

- To see all databases & to use a specific one, use command:
- Show databases;
- Use arthdb;
- exit;

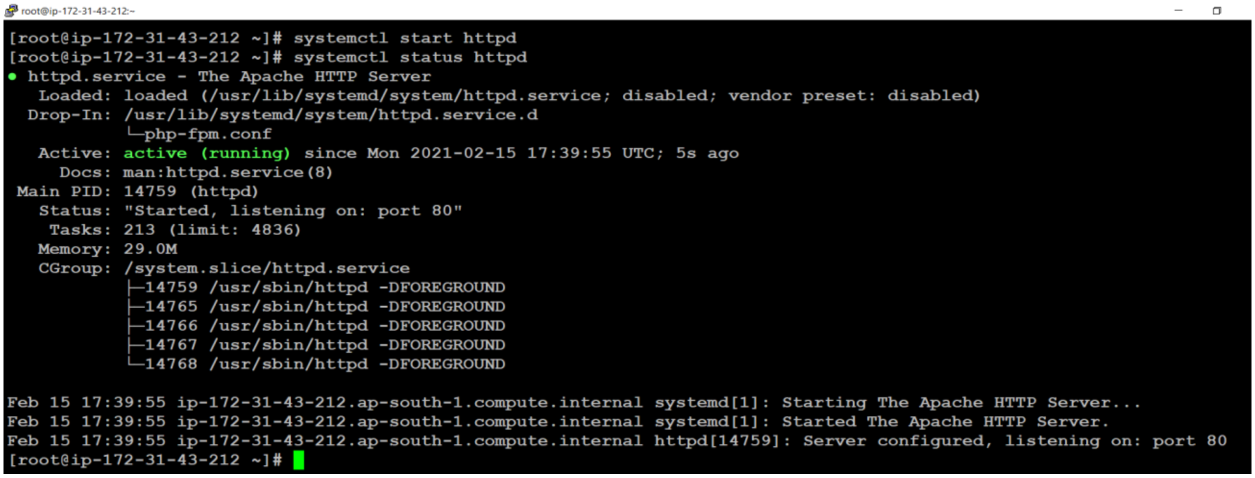


root@ip-172-31-43-212:/etc/yum.repos.d

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| arthdb             |
| innodb             |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
6 rows in set (0.00 sec)

mysql> use arthdb;
Database changed
mysql> exit;
Bye
```

- Start httpd service & check status.

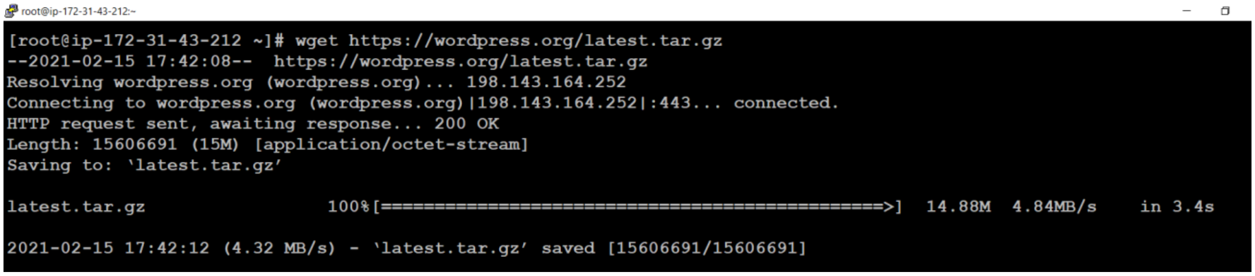
```
systemctl start httpd
systemctl status httpd
```



```
[root@ip-172-31-43-212 ~]# systemctl start httpd
[root@ip-172-31-43-212 ~]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
  Drop-In: /usr/lib/systemd/system/httpd.service.d
           └─php-fpm.conf
   Active: active (running) since Mon 2021-02-15 17:39:55 UTC; 5s ago
     Docs: man:httpd.service(8)
 Main PID: 14759 (httpd)
   Status: "Started, listening on: port 80"
    Tasks: 213 (limit: 4836)
   Memory: 29.0M
   CGroup: /system.slice/httpd.service
           ├─14759 /usr/sbin/httpd -DFOREGROUND
           ├─14765 /usr/sbin/httpd -DFOREGROUND
           ├─14766 /usr/sbin/httpd -DFOREGROUND
           ├─14767 /usr/sbin/httpd -DFOREGROUND
           └─14768 /usr/sbin/httpd -DFOREGROUND

Feb 15 17:39:55 ip-172-31-43-212.ap-south-1.compute.internal systemd[1]: Starting The Apache HTTP Server...
Feb 15 17:39:55 ip-172-31-43-212.ap-south-1.compute.internal systemd[1]: Started The Apache HTTP Server.
Feb 15 17:39:55 ip-172-31-43-212.ap-south-1.compute.internal httpd[14759]: Server configured, listening on: port 80
[root@ip-172-31-43-212 ~]#
```

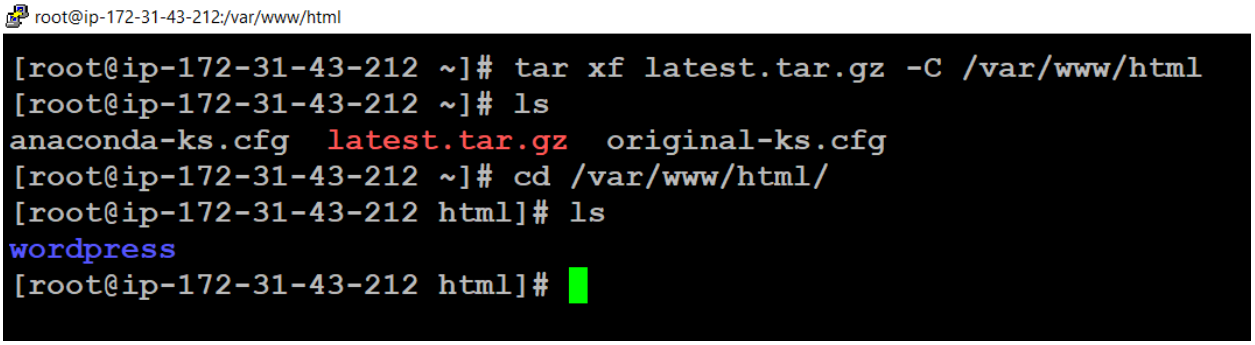
Step 5): Download php application name “WordPress.”

- To download WordPress code, use command:
- wget https://wordpress.org/latest.tar.gz
- It will download the php code inside a zip folder.



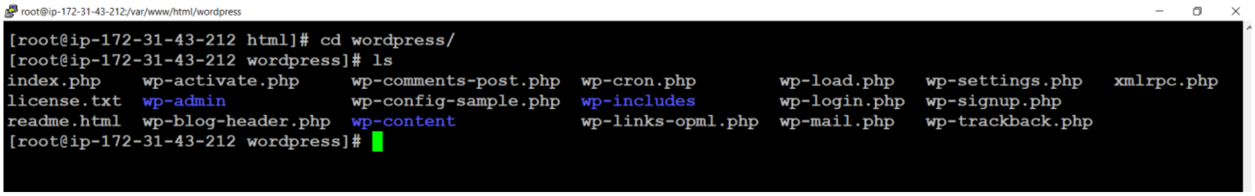
```
[root@ip-172-31-43-212 ~]# wget https://wordpress.org/latest.tar.gz
--2021-02-15 17:42:08--  https://wordpress.org/latest.tar.gz
Resolving wordpress.org (wordpress.org)... 198.143.164.252
Connecting to wordpress.org (wordpress.org)|198.143.164.252|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 15606691 (15M) [application/octet-stream]
Saving to: ‘latest.tar.gz’

latest.tar.gz              100%[==================================================>]  14.88M  4.84MB/s    in 3.4s

2021-02-15 17:42:12 (4.32 MB/s) - ‘latest.tar.gz’ saved [15606691/15606691]
```

- To unzip the code & to transfer inside document root(/var/www/html), use command:
- tar xf latest.tar.gz -C /var/www/html
- After unzip, we get a WordPress folder.



```
[root@ip-172-31-43-212 ~]# tar xf latest.tar.gz -C /var/www/html
[root@ip-172-31-43-212 ~]# ls
anaconda-ks.cfg  latest.tar.gz  original-ks.cfg
[root@ip-172-31-43-212 ~]# cd /var/www/html/
[root@ip-172-31-43-212 html]# ls
wordpress
[root@ip-172-31-43-212 html]#
```

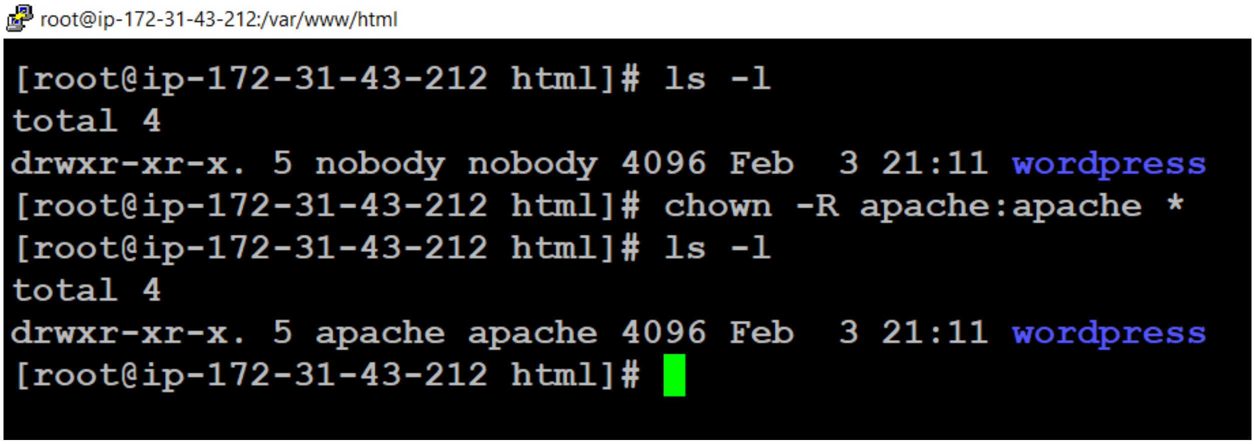
- Use ls command to see the files stored inside the wordpress folder.



root@ip-172-31-43-212:/var/www/html/wordpress

```
[root@ip-172-31-43-212 html]# cd wordpress/
[root@ip-172-31-43-212 wordpress]# ls
index.php    wp-activate.php     wp-comments-post.php  wp-cron.php        wp-load.php   wp-settings.php   xmlrpc.php
license.txt  wp-admin            wp-config-sample.php  wp-includes        wp-login.php  wp-signup.php
readme.html  wp-blog-header.php  wp-content            wp-links-opml.php  wp-mail.php   wp-trackback.php
[root@ip-172-31-43-212 wordpress]#
```

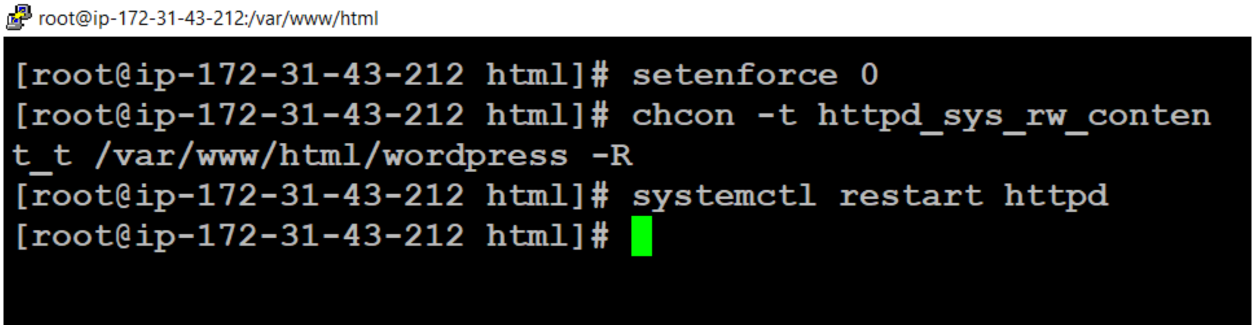
- Make apache as owner of the wordpress folder using command:
- chown -R apache:apache *



root@ip-172-31-43-212:/var/www/html

```
[root@ip-172-31-43-212 html]# ls -l
total 4
drwxr-xr-x. 5 nobody nobody 4096 Feb  3 21:11 wordpress
[root@ip-172-31-43-212 html]# chown -R apache:apache *
[root@ip-172-31-43-212 html]# ls -l
total 4
drwxr-xr-x. 5 apache apache 4096 Feb  3 21:11 wordpress
[root@ip-172-31-43-212 html]#
```

- Httpd cannot write to folder/file because of SELinux so allow it & use the command:
- setenforce 0
- chcon -t httpd_sys_rw_content_t /var/www/html/wordpress -R
- systemctl restart httpd

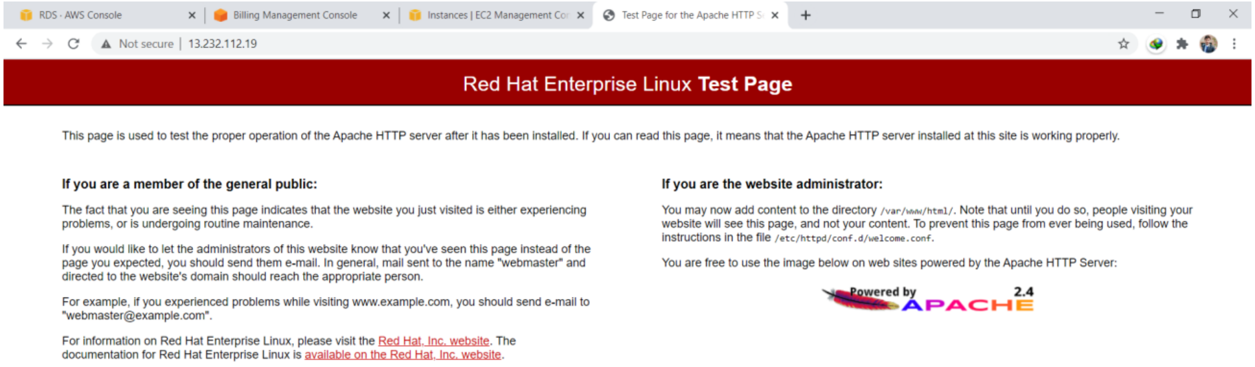


root@ip-172-31-43-212:/var/www/html

```
[root@ip-172-31-43-212 html]# setenforce 0
[root@ip-172-31-43-212 html]# chcon -t httpd_sys_rw_conten
t_t /var/www/html/wordpress -R
[root@ip-172-31-43-212 html]# systemctl restart httpd
[root@ip-172-31-43-212 html]#
```

Step 6): Use Browser to access your WordPress application.

- Paste the public ip of your instance in the browser.
- http://<ec2_public_ip>


RDS · AWS Console
Billing Management Console
Instances | EC2 Management Co
Test Page for the Apache HTTP S
Not secure | 13.232.112.19
Powered by APACHE 2.4

Red Hat Enterprise Linux **Test Page**

This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you can read this page, it means that the Apache HTTP server installed at this site is working properly.

If you are a member of the general public:

The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting www.example.com, you should send e-mail to "webmaster@example.com".

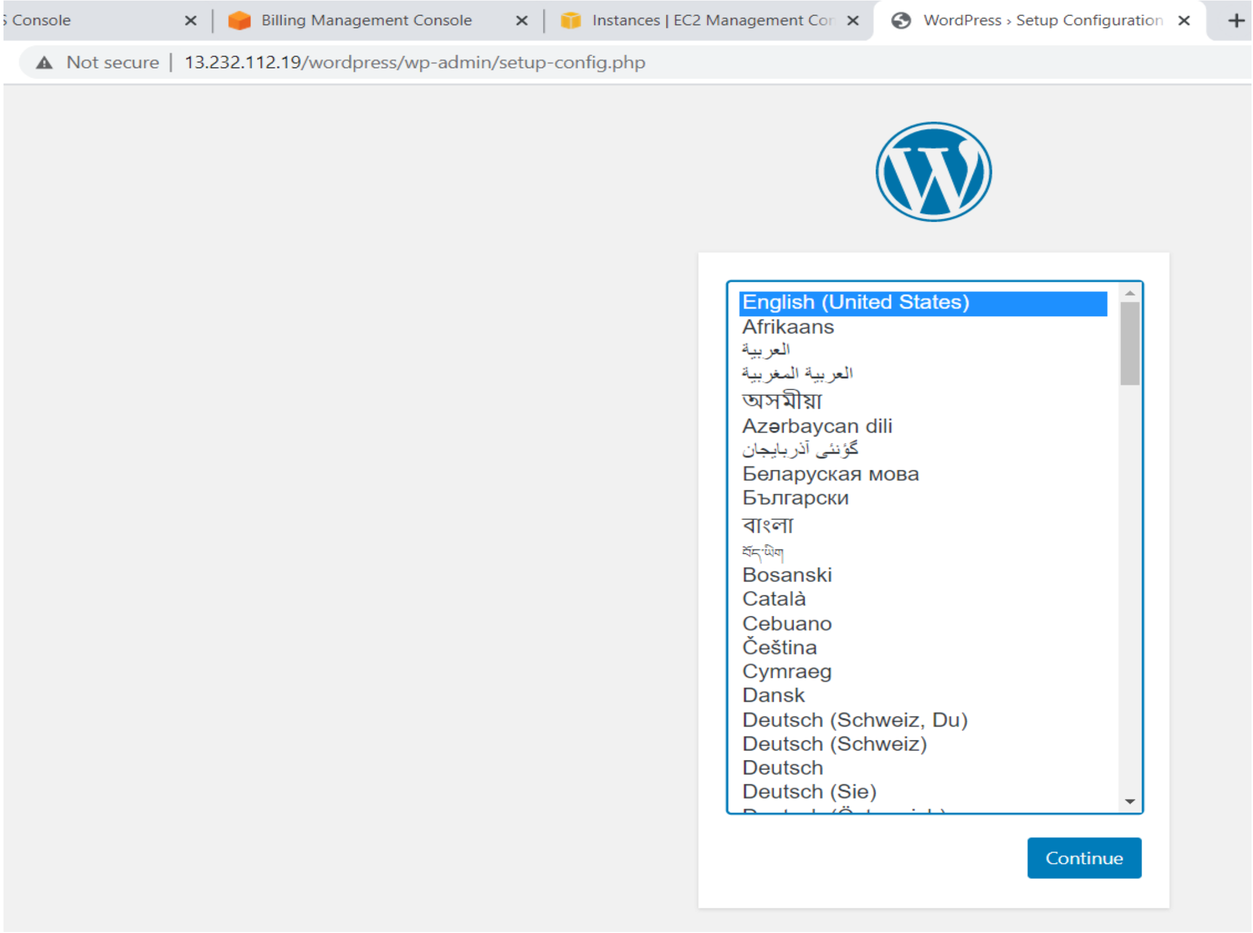
For information on Red Hat Enterprise Linux, please visit the Red Hat, Inc. website. The documentation for Red Hat Enterprise Linux is available on the Red Hat, Inc. website.

If you are the website administrator:

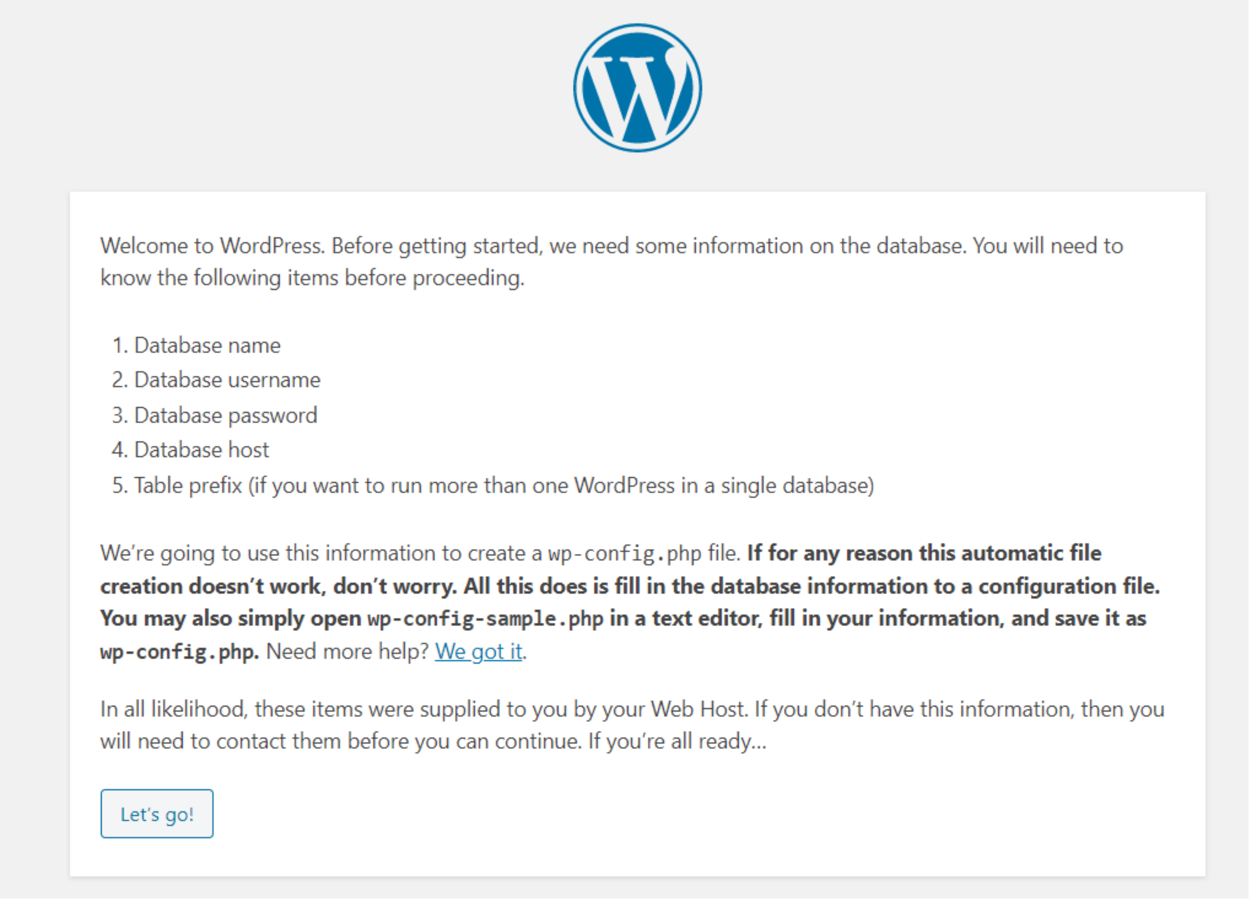
You may now add content to the directory `/var/www/html/`. Note that until you do so, people visiting your website will see this page, and not your content. To prevent this page from ever being used, follow the instructions in the file `/etc/httpd/conf.d/welcome.conf`.

You are free to use the image below on web sites powered by the Apache HTTP Server:

- Just add /wordpress keyword in your url, it will automatically work ahead.
- i.e http://<ec2_public_ip/wordpress
- *Choose any language & click on Continue.*



- *Click on Let's go button but before this click you should have all the required information.*



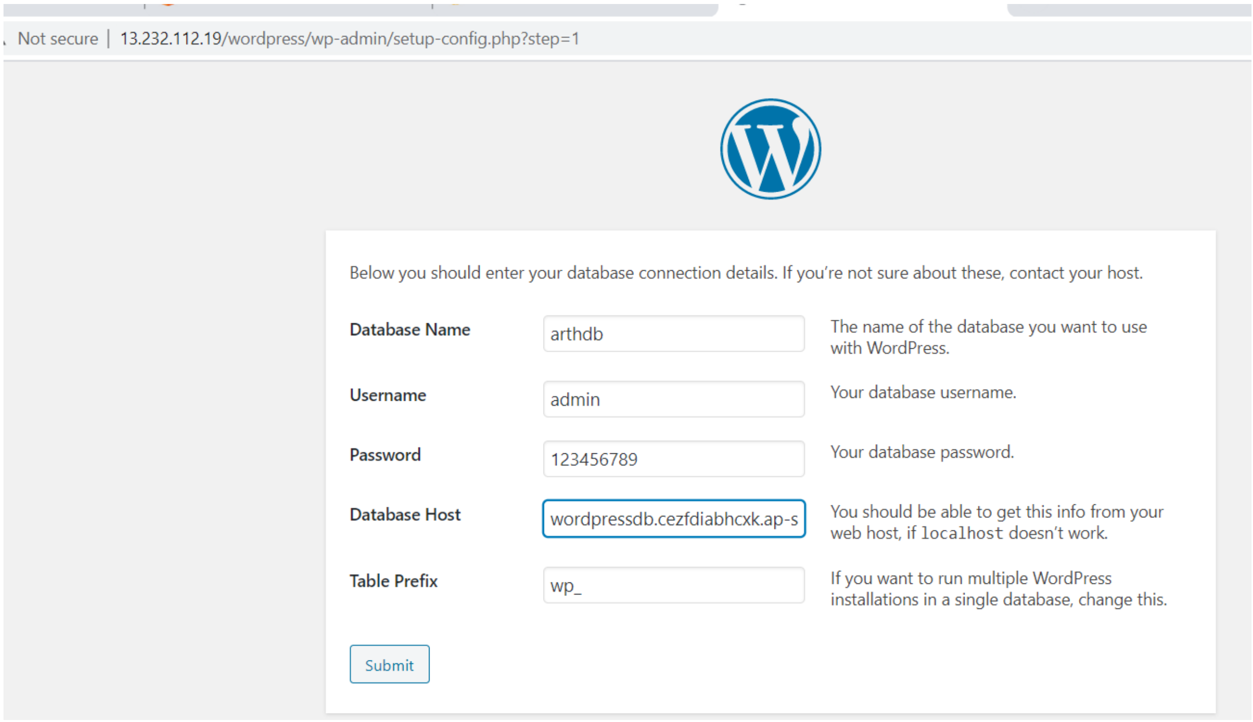
Welcome to WordPress. Before getting started, we need some information on the database. You will need to know the following items before proceeding.

1. Database name
2. Database username
3. Database password
4. Database host
5. Table prefix (if you want to run more than one WordPress in a single database)

We're going to use this information to create a `wp-config.php` file. **If for any reason this automatic file creation doesn't work, don't worry. All this does is fill in the database information to a configuration file. You may also simply open `wp-config-sample.php` in a text editor, fill in your information, and save it as `wp-config.php`.** Need more help? We got it.

In all likelihood, these items were supplied to you by your Web Host. If you don't have this information, then you will need to contact them before you can continue. If you're all ready…

- *Enter your database name, database user name, database password & database endpoint url, then click on submit.*

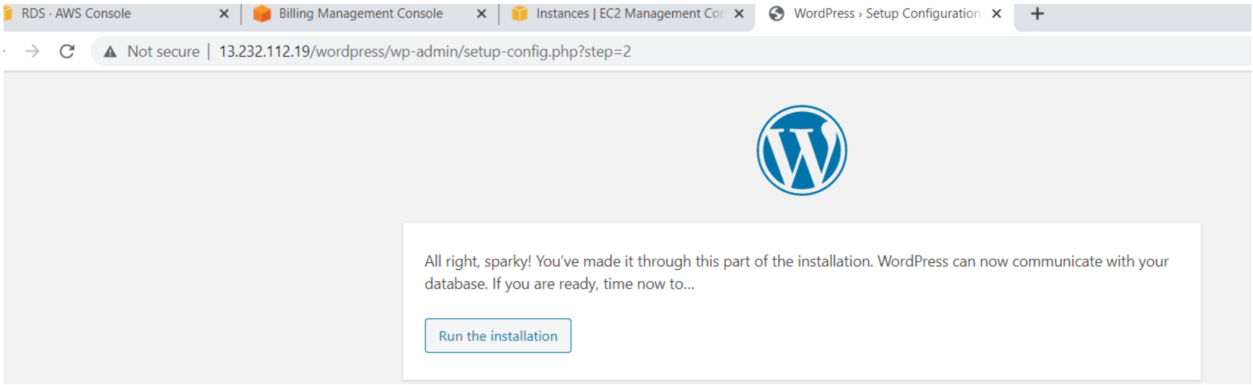


Not secure | 13.232.112.19/wordpress/wp-admin/setup-config.php?step=1

Below you should enter your database connection details. If you're not sure about these, contact your host.

Database Name	arthdb	The name of the database you want to use with WordPress.
Username	admin	Your database username.
Password	123456789	Your database password.
Database Host	wordpressdb.cezfdiabhcxk.ap-s	You should be able to get this info from your web host, if `localhost` doesn't work.
Table Prefix	wp_	If you want to run multiple WordPress installations in a single database, change this.

- *Finally click on "Run the installation" button.*

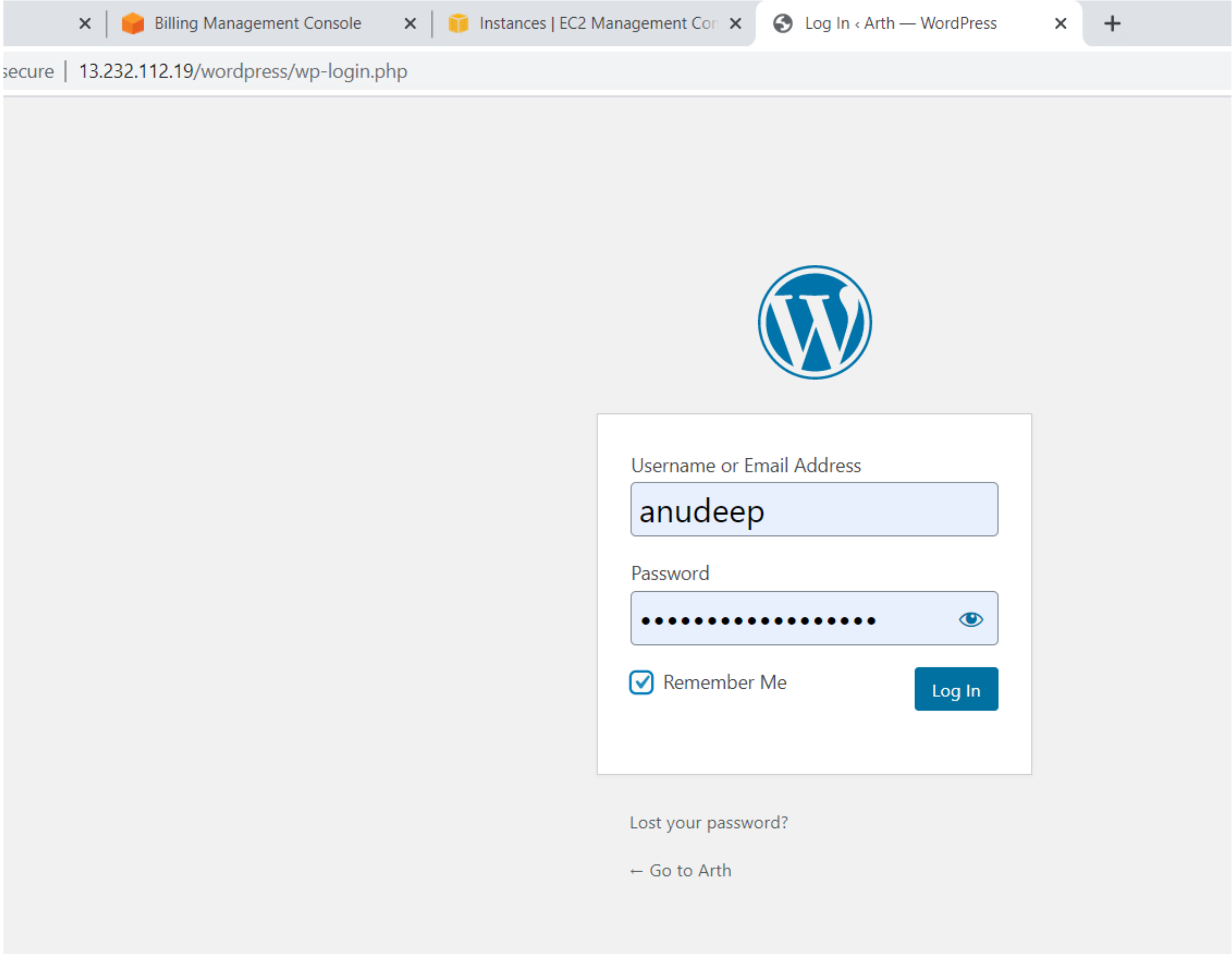


RDS · AWS Console
Billing Management Console
Instances | EC2 Management Cor
WordPress › Setup Configuration
+

Not secure | 13.232.112.19/wordpress/wp-admin/setup-config.php?step=2

All right, sparky! You've made it through this part of the installation. WordPress can now communicate with your database. If you are ready, time now to…

- *Set title, username, password for your WordPress application. Then Login inside your WordPress account and write your first blog on WordPress*


Billing Management Console
Instances | EC2 Management Cor
Log In ‹ Arth — WordPress

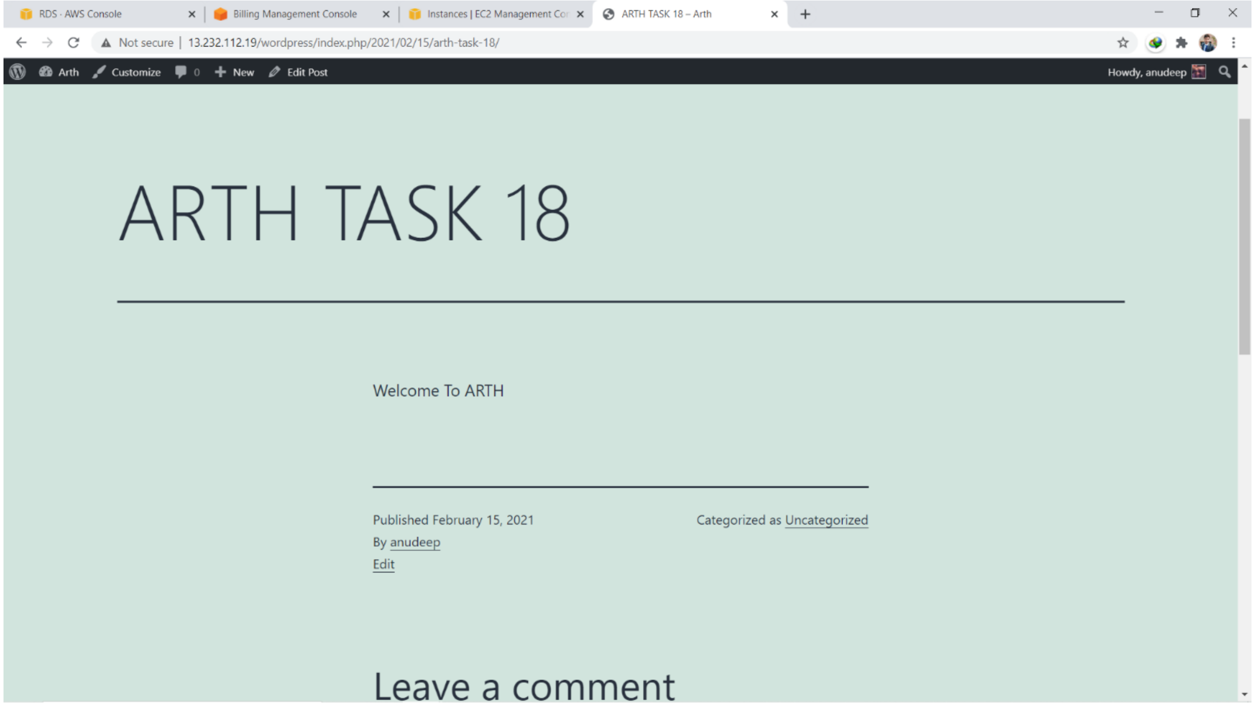
secure | 13.232.112.19/wordpress/wp-login.php

Username or Email Address: anudeep

Password: ••••••••••••••••••

☑ Remember Me

Log In
Lost your password?
← Go to Arth



Not secure | 13.232.112.19/wordpress/index.php/2021/02/15/arth-task-18/

ARTH TASK 18

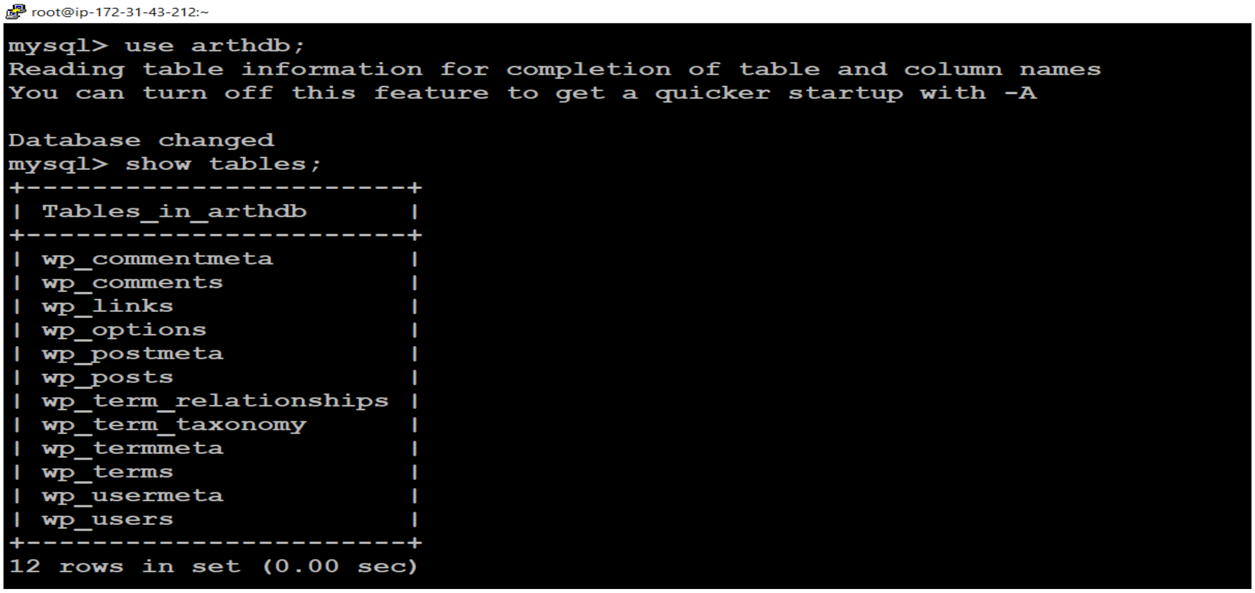
Welcome To ARTH

Published February 15, 2021
By anudeep
Edit

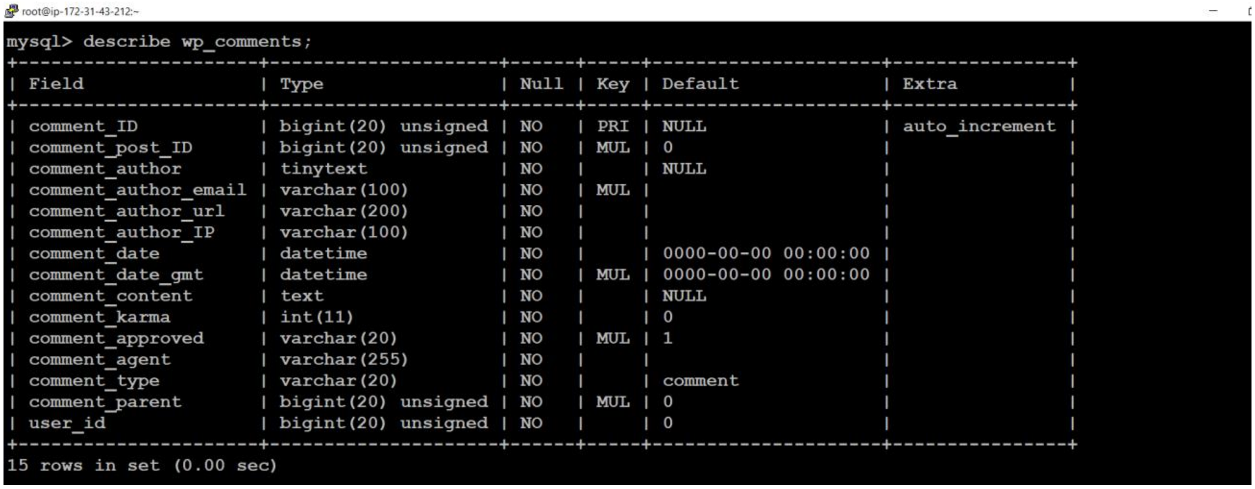
Categorized as Uncategorized

Leave a comment

- *Check data stored in your MySQL database for verification, use command:*
- Show tables;
- describe <data_stored>;



```
mysql> use arthdb;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----------------------+
| Tables_in_arthdb      |
+-----------------------+
| wp_commentmeta        |
| wp_comments           |
| wp_links              |
| wp_options            |
| wp_postmeta           |
| wp_posts              |
| wp_term_relationships |
| wp_term_taxonomy      |
| wp_termmeta           |
| wp_terms              |
| wp_usermeta           |
| wp_users              |
+-----------------------+
12 rows in set (0.00 sec)
```



```
mysql> describe wp_comments;
+----------------------+---------------------+------+-----+---------------------+----------------+
| Field                | Type                | Null | Key | Default             | Extra          |
+----------------------+---------------------+------+-----+---------------------+----------------+
| comment_ID           | bigint(20) unsigned | NO   | PRI | NULL                | auto_increment |
| comment_post_ID      | bigint(20) unsigned | NO   | MUL | 0                   |                |
| comment_author       | tinytext            | NO   |     | NULL                |                |
| comment_author_email | varchar(100)        | NO   | MUL |                     |                |
| comment_author_url   | varchar(200)        | NO   |     |                     |                |
| comment_author_IP    | varchar(100)        | NO   |     |                     |                |
| comment_date         | datetime            | NO   |     | 0000-00-00 00:00:00 |                |
| comment_date_gmt     | datetime            | NO   | MUL | 0000-00-00 00:00:00 |                |
| comment_content      | text                | NO   |     | NULL                |                |
| comment_karma        | int(11)             | NO   |     | 0                   |                |
| comment_approved     | varchar(20)         | NO   | MUL | 1                   |                |
| comment_agent        | varchar(255)        | NO   |     |                     |                |
| comment_type         | varchar(20)         | NO   |     | comment             |                |
| comment_parent       | bigint(20) unsigned | NO   | MUL | 0                   |                |
| user_id              | bigint(20) unsigned | NO   |     | 0                   |                |
+----------------------+---------------------+------+-----+---------------------+----------------+
15 rows in set (0.00 sec)
```

Finally! We have done our todays agenda!

Thankyou