

# Creating simple 2-tier web architecture using AWS

Create a VPC and 6 subnet using this CIDR

170.20.0.0/20 for your VPC

Use those CIDR for ur subnet (optional)

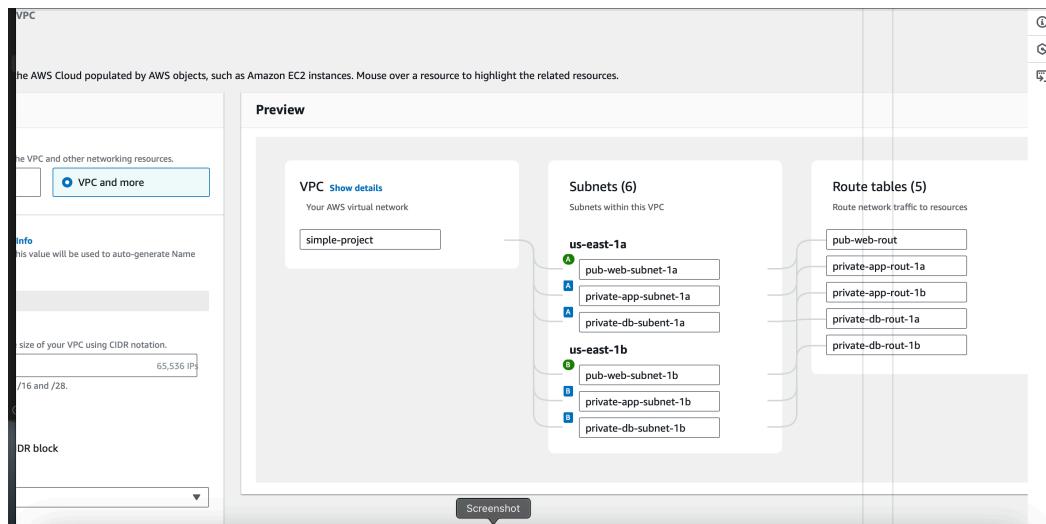
170.20.1.0/24                    170.20.4.0/24                    170.20.5.0/24

170.20.2.0/24                    170.20.3.0/24                    170.20.6.0/24

[PUB-WEB-subnet-01]

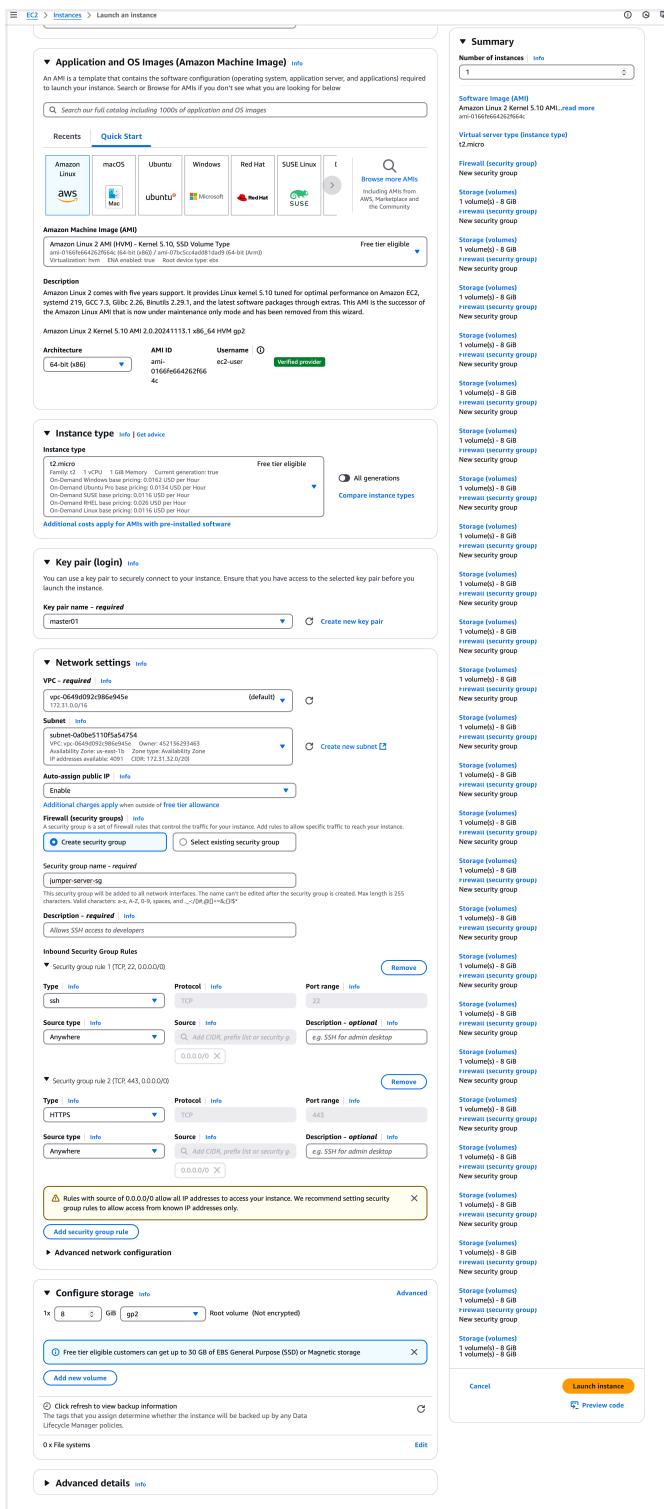
[private-app-subnet-01]

[private-db-subnet]



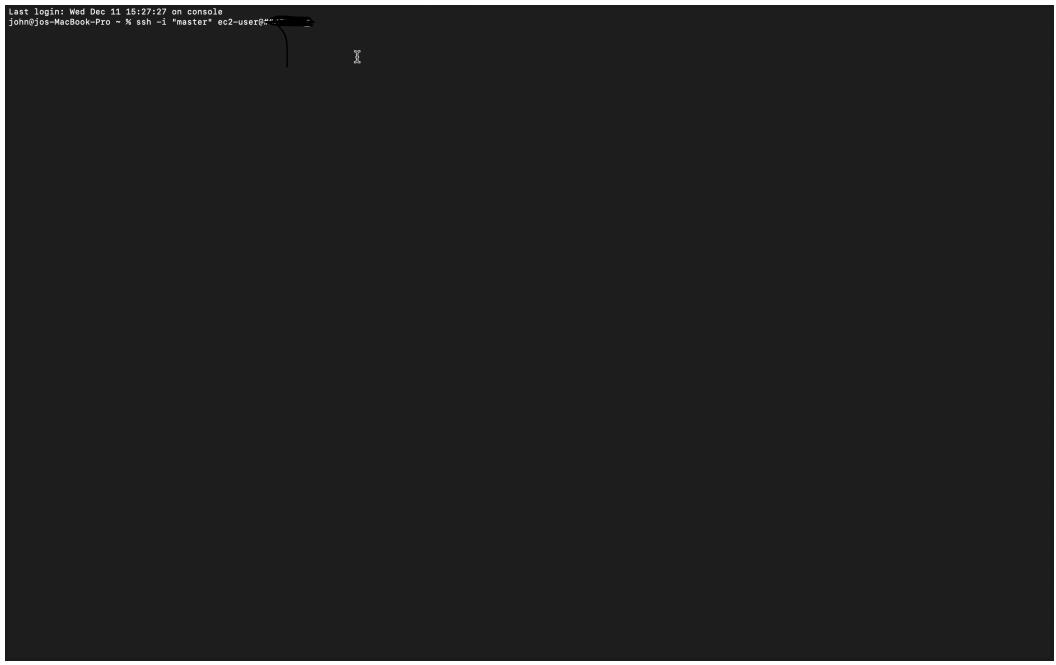
After creating and configure your VPC now, let create a ec2-instances  
1, creating ec2 instances

- Give name and choose operating system could-be (ubuntu, os, or amazon linux )
- select instanc type (t2.small )
- create key-pair
- create new security group
- configure your networking



Once you have created your jumper server, now let create another ec2 instances for our LAMP ApacheMySQLPHP server  
Using the above steps create another instances (one thing to remember use the private-app-subnet. And create them on both Availability Zones

Once completed navigate to your terminal login to ur server using SSH-keys, and deploy LAMP server using this documentation (<https://docs.aws.amazon.com/linux/al2/ug/ec2-lamp-amazon-linux-2.html>)



```
Last login: Wed Dec 11 18:27:27 on console
John@jos-MacBook-Pro ~ % ssh -i "master" ec2-user@"
```

Once you have completed configure you apache and php server, on both Availability Zones now lets Create Application-load-balancer and RDS MySQL-database\

Creating application load balancer

- Give name
- Load balance type = internet facing (HTTP-HTTPS)
- Select ip adders of = ipv4
- Chose your preview VPC
- Create a security group ([alb-security.sg](#)) allow request of =HTTP / HTTPS
- Add listeners ( create target groups

You can configure cloud front, WAF and global accelerator.(optional)

**Create Application Load Balancer** [info](#)

The Application Load Balancer distributes incoming HTTP and HTTPS traffic across multiple targets such as Amazon EC2 instances, microservices, and containers, based on request attributes. When the load balancer receives a connection request, it evaluates the listener rules in priority order to determine which rule to apply, and if applicable, it selects a target from the target group for the rule action.

**How Application Load Balancers work**

**Basic configuration**

**Load balancer name**  
Leave this field empty to generate your ALB name and can't be changed after the load balancer is created.  
**Simple-project-alb**

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

**Scheme** [info](#)  
Scheme can't be changed after the load balancer is created.  
 **Internet-facing**

- Serves internet-facing traffic.
- Targets can be private IP addresses.
- DNS name is publicly resolvable.
- Requires a public IP address.

 **Internal**

- Serves internal traffic.
- Targets can be private IP addresses.
- DNS name is not publicly resolvable.
- Incompatible with the IPv4 and Dualstack IP address types.

**Load balancer IP address type** [info](#)  
Select the preferred IP address type to assign to the load balancer. The VPC and subnets mapped to this load balancer must include the selected IP address type. Public IPv4 addresses have an additional cost.

**IPv4**  
Includes only IPv4 addresses.

**IPv6**  
Includes IPv4 and IPv6 addresses.

**Dualstack**  
Requires a public IPv4 address, and private IPv4 and IPv6 addresses. Compatible with Internet-facing load balancers only.

**Network mapping** [info](#)  
The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

**VPC** [info](#)  
The load balancer will enter and route within the selected VPC. The selected VPC is also where the load balancer targets must be located unless routing to Lambda or on-premises targets, or if using VPC peering. To confirm the VPC for your targets, view [target group](#) [info](#) For a new VPC, [create a VPC](#) [info](#)

Select a VPC: [Select a VPC](#) [info](#)

**Mappings** [info](#)  
Select at least two Availability Zones and one subnet per zone. The load balancer routes traffic to targets in these Availability Zones only. Availability Zones that are not supported by the load balancer or the VPC are not available for selection.

**Availability Zones**

**Security groups** [info](#)  
A security group is a set of firewall rules that control the traffic to your load balancer. Select an existing security group, or you can create a new security group [info](#)

**Security groups**  
Select up to 5 security groups [info](#)

**Listeners and routing** [info](#)  
A Listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets.

**Listener: HTTP80**  
Protocol: **HTTP** Port: **80** Default action: [info](#) Forward to: [Select a target group](#) [info](#) [Create target group](#) [info](#)

**Listener tags - optional**  
Consider adding tags to your listener. Tags enable you to categorize your AWS resources so you can more easily manage them.

[Add listener tag](#) You can add up to 50 more tags. [Add listener](#)

**Load balancer tags - optional**  
Consider adding tags to your load balancer. Tags enable you to categorize your AWS resources so you can more easily manage them. The 'Key' is required, but 'Value' is optional. For example, you can have Key = production-webscraper, or Key = webscraper, and Value = production.

**Optimize with service integrations - optional**  
Optimize your load balancing architecture by integrating third-party services with this load balancer at launch. You can also add these and other services after your load balancer is created by reviewing the load balancer's "Integrations" tab.

**Amazon CloudFront + AWS Web Application Firewall (WAF) - new** [info](#)  
Optimizes Performance, Availability Security  
 **Apply application layer acceleration and security protections - in front of the load balancer**  
Automatically configures and creates a CloudFront distribution with the basic recommended AWS WAF security protections, and associates it to your load balancer. [Additional charges apply](#) [info](#)

**Benefits and considerations**

**AWS Web Application Firewall (WAF)** [info](#)  
Optimizes Security  
 **Apply application layer security protections - in front of targets**  
Your choice of either a pre-defined security configuration with basic recommended AWS WAF security protections, or associate any of your existing WAF configurations for custom protections. [Additional charges apply](#) [info](#)

**Benefits and considerations**

**AWS Global Accelerator** [info](#)  
Optimizes Performance, Availability  
 **Apply global load balancing across multiple regions**  
Creates a global endpoint that uses global static IPs that act as a fixed entry point to your load balancer. If you do not need global static IPs or traffic management across multiple regions, select Amazon CloudFront. [Additional charges apply](#) [info](#)

**Benefits and considerations**

**Review**  
Review the load balancer configurations and make changes if needed. After you finish reviewing the configurations, choose **Create load balancer**.

Summary	Review and confirm your configurations. <a href="#">Estimate cost</a> <a href="#">info</a>	Service integrations	Attributes
<b>Basic configuration</b> <a href="#">info</a> Simple-project-alb <ul style="list-style-type: none"> <li>Internet-facing</li> <li>IPv4</li> </ul>	<b>Security groups</b> <a href="#">info</a> Security group not defined	<b>Network mapping</b> <a href="#">info</a> VPC Subnet not defined	<b>Listeners and routing</b> <a href="#">info</a> <ul style="list-style-type: none"> <li>HTTP:80 (default to Target group not defined)</li> </ul>
<b>Tags</b> <a href="#">info</a> None			<b>Certain default attributes will be applied to your load balancer. You can view and edit them after creating the load balancer.</b>

**Creation workflow and status**

**Server-side tasks and status**  
After completing and submitting the above steps, all server-side tasks and their statuses become available for monitoring.

[Cancel](#) [Create load balancer](#)

## Create target group for your alb listener

- Give name
- Choose a target type (instance)
- Choose Protocol : HTTP
- Choose ip address type = ipv4
- Choose preview VPC
- After that Register you instance that we created earlier

EC2 > Target groups > Create target group

**Specify group details**

Your load balancer routes requests to the targets in a target group and performs health checks on the targets.

**Basic configuration**

Settings in this section can't be changed after the target group is created.

**Choose a target type**

Instances

- Supports load balancing to instances within a specific VPC.
- Facilitates the use of [Amazon EC2 Auto Scaling](#) to manage and scale your EC2 capacity.

IP addresses

- Supports load balancing to VPC and on-premises resources.
- Facilities routing to multiple IP addresses and network interfaces on the same instance.
- Offers flexibility with microservice based architectures, simplifying inter-application communication.
- Supports IPv6 targets, enabling end-to-end IPv6 communication, and IPv4-to-IPv6 NAT.

Lambda function

- Facilitates routing to a single Lambda function.
- Accessible to Application Load Balancers only.

Application Load Balancer

- Offers the flexibility for a Network Load Balancer to accept and route TCP requests within a specific VPC.
- Facilitates using static IP addresses and PrivateLink with an Application Load Balancer.

**Target group name**

example

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

**Protocol : Port**

Choose a protocol for your target group that corresponds to the Load Balancer type that will route traffic to it. Some protocols now include anomaly detection for the targets and you can set mitigation options once your target group is created. This choice cannot be changed after creation

HTTP 80 ▾ 1-65535

**IP address type**

Only targets with the indicated IP address type can be registered to this target group.

IPv4

Each instance has a default network interface (eth0) that is assigned the primary private IPv4 address. The instance's primary private IPv4 address is the one that will be applied to the target.

IPv6

Each instance you register must have an assigned primary IPv6 address. This is configured on the instance's default network interface (eth0). [Learn more](#)

**VPC**

Select the VPC with the instances that you want to include in the target group. Only VPCs that support the IP address type selected above are available in this list.

vpc-0648d092c986845e  
IPv4 VPC CIDR: 172.31.0.0/16

**Protocol version**

HTTP1

Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.

HTTP2

Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available.

gRPC

Send requests to targets using gRPC. Supported when the request protocol is gRPC.

**Health checks**

The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.

**Health check protocol**

HTTP

**Health check path**

Use the default path of "/" to perform health checks on the root, or specify a custom path if preferred.  
/

Up to 1024 characters allowed.

**Advanced health check settings**

**Attributes**

Certain default attributes will be applied to your target group. You can view and edit them after creating the target group.

**Tags - optional**

Consider adding tags to your target group. Tags enable you to categorize your AWS resources so you can more easily manage them.

Cancel Next

Once we confirm our traffic request is been distributed we can continue on creating our database subnet groups for Multi-AZ DB clusters

- Give name and select 3 AZ ("us-east-1a", "us-east-1b", "us-east-1c")
- Select you preview VPC
- And choose subnet ("private-db-subnet-1a", "private-db-subnet-1b", "private-db-subnet-1c")

RDS > Subnet groups > Create DB subnet group

### Create DB subnet group

To create a new subnet group, give it a name and a description, and choose an existing VPC. You will then be able to add subnets related to that VPC.

#### Subnet group details

**Name**  
You won't be able to modify the name after your subnet group has been created.

Must contain from 1 to 255 characters. Alphanumeric characters, spaces, hyphens, underscores, and periods are allowed.

**Description**

**VPC**  
Choose a VPC identifier that corresponds to the subnets you want to use for your DB subnet group. You won't be able to choose a different VPC identifier after your subnet group has been created.

6 Subnets, 6 Availability Zones

#### Add subnets

**Availability Zones**  
Choose the Availability Zones that include the subnets you want to add.

**Subnets**  
Choose the subnets that you want to add. The list includes the subnets in the selected Availability Zones.  
  

- Subnet ID: subnet-0a0be5110f5a54754 CIDR: 172.31.32.0/20
- Subnet ID: subnet-06dbc81eff3e3c52c CIDR: 172.31.16.0/20
- Subnet ID: subnet-0f5da994017f3a61a CIDR: 172.31.0.0/20

For Multi-AZ DB clusters, you must select 3 subnets in 3 different Availability Zones.

**Subnets selected (3)**

Availability zone	Subnet name	Subnet ID	CIDR block
us-east-1b	-	subnet-0a0be5110f5a54754	172.31.32.0/20
us-east-1a	-	subnet-06dbc81eff3e3c52c	172.31.16.0/20
us-east-1c	-	subnet-0f5da994017f3a61a	172.31.0.0/20

Now let's create our database

RDS > Databases

**Amazon RDS**

- Dashboard
- Databases**
- Query Editor
- Performance insights
- Snapshots
- Exports in Amazon S3
- Automated backups
- Reserved instances
- Proxies

**Introducing Global Database writer endpoint**  
Each global cluster now has a writer endpoint that you can use to send your application's requests to the writer instance in the primary cluster of your Global Database. Aurora automatically updates the endpoint upon a cross-region failover or switchover operation, ensuring that requests are routed to the writer instance in the new primary cluster without the need for changes to your application code or configuration. [Learn more](#)

**Easy path homogeneous data migrations from EC2 database to RDS**  
With integrated homogeneous data migration powered by AWS DMS, the Amazon RDS console leverages simple and performant data migration from EC2 database to equivalent RDS database. To get started, select an existing RDS database and choose the [Migrate data from EC2 database](#) in the Actions menu. Make sure you check the supported engine types and feature limitations. [Learn more](#)

**Databases (0)**  Group resources  Actions

- Select MySQL data base
- Choose template = production
- Deployment options = Multi-AZ DB cluster and Choose name
- Give credentials ( username and password)
- Select your database Instance type and storage type as well
- Configure your VPC, subnet groups and denied public access
- Create security groups, (optional ) you can create RDS proxy.
- Monitoring is optional it's best to turn it on

Once the database initialization is completed copy the end point url and place it in config.inc.php this file is located in /var/www/html/phpMyAdmin/ )

```

<?php

/*
 * phpMyAdmin sample configuration, you can use it as base for
 * manual configuration. For easier setup you can use setup/
 * All directives are explained in documentation in the doc/ folder
 * or at <https://docs.phpmyadmin.net/>.
 */

declare(strict_types=1);

/** This is needed for cookie based authentication to encrypt the cookie,
 * Needs to be a 32-bytes long string of random bytes. See FAQ 2.10.
 */
$cfg['blowfish_secret'] = ''; /* YOU MUST FILL IN THIS FOR COOKIE AUTH! */

/** Servers configuration
 */
$server = 0;

/** First server
 */
$server++;

/** Authentication type */
$cfg['Servers'][$server]['auth_type'] = 'cookie';
// Server parameters
$cfg['Servers'][$server]['host'] = 'web-alb-01-cxyg9gcau2.us-east-1.rds.amazonaws.com';
$cfg['Servers'][$server]['port'] = false;
$cfg['Servers'][$server]['AllowNoPassword'] = false;

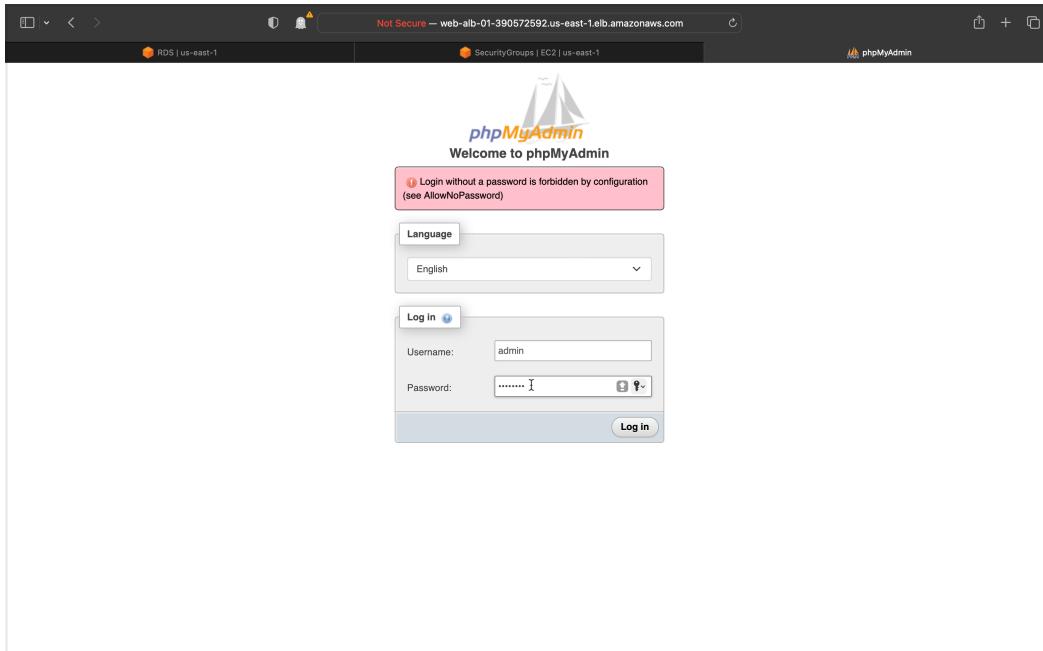
/** phpMyAdmin configuration storage settings.
 */
// User used to manipulate with storage
// $cfg['Servers'][$server]['controlhost'] = '';
// $cfg['Servers'][$server]['controlport'] = '';
// $cfg['Servers'][$server]['controluser'] = 'pma';
// $cfg['Servers'][$server]['controlpass'] = 'pmapass';

/** Storage database and tables */
// $cfg['Servers'][$server]['pmadb'] = 'phpmyadmin';
// $cfg['Servers'][$server]['bookmarktable'] = 'pma__bookmark';
// $cfg['Servers'][$server]['relation'] = 'pma__relation';
// $cfg['Servers'][$server]['table_info'] = 'pma__table_info';
// $cfg['Servers'][$server]['table_coords'] = 'pma__table_coords';
// $cfg['Servers'][$server]['table_uiprefs'] = 'pma__table_uiprefs';
// $cfg['Servers'][$server]['tracking'] = 'pma__tracking';
// $cfg['Servers'][$server]['userconfig'] = 'pma__userconfig';
// $cfg['Servers'][$server]['recent'] = 'pma__recent';
// $cfg['Servers'][$server]['favorite'] = 'pma__favorite';
// $cfg['Servers'][$server]['users'] = 'pma__users';
// $cfg['Servers'][$server]['usergroups'] = 'pma__usergroups';
// $cfg['Servers'][$server]['privileges'] = 'pma__privileges';
// $cfg['Servers'][$server]['privilegehiding'] = 'pma__privilegehiding';
// $cfg['Servers'][$server]['savedsearches'] = 'pma__savedsearches';
// $cfg['Servers'][$server]['central_columns'] = 'pma__central_columns';

34,3           Top

```

Navget to the browser <http://web-alb-01-390572592.us-east-1.elb.amazonaws.com>



You should receive an output like this now put the credential that you created when luching mySql database

Once you log-in it should look like this

Not Secure — web-alb-01-390572592.us-east-1.elb.amazonaws.com

RDS | us-east-1 SecurityGroups | EC2 | us-east-1

phpMyAdmin

Databases SQL Status User accounts Export Import Settings Binary log Replication Variables More

General settings

Change password Server connection collation: utf8mb4\_unicode\_ci More settings

Appearance settings

Language English Theme pmahomme View all

Database server

- Server: web-rds-01.cxygw0gcau24.us-east-1.rds.amazonaws.com via TCP/IP
- Server type: MySQL
- Server connection: SSL is not being used
- Server version: 8.0.35 - Source distribution
- Protocol version: 10
- User: admin@170.20.4.210
- Server charset: UTF-8 Unicode (utf8mb4)

Web server

- Apache/2.4.62 ()
- Database client version: libmysql - mysqlnd 8.2.19
- PHP extension: mysqli curl
- PHP version: 8.2.19

phpMyAdmin

- Version information: 5.2.1 (up to date)
- Documentation
- Official Homepage
- Contribute
- Get support
- List of changes
- License

Console mbstring PHP extension was not found and you seem to be using a multibyte charset. Without the mbstring extension phpMyAdmin is unable to split strings correctly and it may result in

