

lab 9 : connecting & configuring database instance to a web server

TASK 1

1. Create an EC2 Security Group

The screenshot shows the AWS EC2 Security Groups interface. A success message at the top right states: "Security group (sg-00254ea9682396eb6 | EC2SG) was created successfully". The main card displays the security group details:

Security group name	sg-00254ea9682396eb6	Description	VPC ID
EC2SG	sg-00254ea9682396eb6	Security Group for EC2	vpc-007543eb92d582d02
Owner	619071313125	Inbound rules count 1 Permission entry	Outbound rules count 1 Permission entry

The "Inbound rules" tab is selected, showing one rule:

Name	Security group rule ID	IP version	Type	Protocol
-	sgr-0512e5ebc9869c739	IPv4	SSH	TCP

2. Create an RDS Security Group

EC2

Security group (sg-008da5a8f0278d024 | RDSSG) was created successfully

sg-008da5a8f0278d024 - RDSSG

Details

Security group name RDSSG	Security group ID sg-008da5a8f0278d024	Description Security Group for RDS	VPC ID vpc-007543eb92d582d02
Owner 619071313125	Inbound rules count 1 Permission entry	Outbound rules count 1 Permission entry	

Inbound rules **Outbound rules** **Sharing - new** **VPC associations - new** **Tags**

Inbound rules (1)

Name	Security group rule ID	IP version	Type	Protocol
-	sgr-06e4e55d0923520fb	-	MySQL/Aurora	TCP

3. Create an EC2 instance

Success
Successfully initiated launch of instance (i-0fd30c25adfde803d)

Launch log

Next Steps

What would you like to do next with this instance, for example "create alarm" or "create backup"

Create billing and free tier usage alerts
To manage costs and avoid surprise bills, set up email notifications for billing and free tier usage thresholds.

[Create billing alerts](#)

Connect to your instance
Once your instance is running, log into it from your local computer.

[Connect to instance](#)

[Learn more](#)

Connect an RDS database
Configure the connection between an EC2 instance and a database to allow traffic flow between them.

[Connect an RDS database](#)

[Create a new RDS database](#)

[Learn more](#)

Create EBS snapshot policy
Create a policy that automates the creation, retention, and deletion of EBS snapshots.

[Create EBS snapshot policy](#)

Manage detailed monitoring
Enable or disable detailed monitoring for the instance. If you enable detailed monitoring, the Amazon EC2 console displays monitoring graphs with a 1-minute period.

Create Load Balancer
Create a application, network gateway or classic Elastic Load Balancer

[Create Load Balancer](#)

Create AWS budget
AWS Budgets allows you to create budgets, forecast spend, and take action on your costs and usage from a single location.

Manage CloudWatch alarms
Create or update Amazon CloudWatch alarms for the instance.

[Manage CloudWatch alarms](#)

4. Connect to the EC2 instance using SSH

```
shravastiohol@Shravasti-MacBook-Pro ~ % ssh -i ~/Downloads/instancekeypair.pem ec2-user@ec2-3-111-37-49.ap-south-1.compute.amazonaws.com
'                                     #
~\_ #####_      Amazon Linux 2023
~~ \_#####\
~~ \###|
~~   '/ --- https://aws.amazon.com/linux/amazon-linux-2023
~~   V~' '-->
~~   /
~~--. -/
~/m'/
[ec2-user@ip-172-31-6-223 ~]$
```

5. Update the instance

```
[ec2-user@ip-172-31-6-223 ~]$ sudo yum update -
sudo yum install mysql -y
Amazon Linux 2023 Kernel Livepatch repository
Last metadata expiration check: 0:00:01 ago on Wed Mar 19 12:55:49 2025.
Dependencies resolved.
Nothing to do.
Complete!
Last metadata expiration check: 0:00:02 ago on Wed Mar 19 12:55:49 2025.
No match for argument: mysql
Error: Unable to find a match: mysql
[ec2-user@ip-172-31-6-223 ~]$
```

6. Install MariaDB

```
[ec2-user@ip-172-31-6-223 ~]$ sudo dnf install -y httpd php php-mysqli mariadb105
Last metadata expiration check: 0:06:09 ago on Wed Mar 19 12:55:49 2025.
Dependencies resolved.
=====
 Package           Architecture     Version          Repository    Size
=====
Installing:
 httpd            x86_64          2.4.62-1.amzn2023
 mariadb105        x86_64          3:10.5.25-1.amzn2023.0.1
 php8.3           x86_64          8.3.16-1.amzn2023.0.1
 php8.3-mysqld    x86_64          8.3.16-1.amzn2023.0.1
=====
Installing dependencies:

```

```
Installed:
 apr-1.7.5-1.amzn2023.0.4.x86_64
 apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64
 httpd-2.4.62-1.amzn2023.x86_64
 httpd-filesystem-2.4.62-1.amzn2023.noarch
 libbrotli-1.0.9-4.amzn2023.0.2.x86_64
 libssl1t-1.1.34-5.amzn2023.0.2.x86_64
 mariadb-connector-c-3.1.13-1.amzn2023.0.3.x86_64
 mariadb105-3:10.5.25-1.amzn2023.0.1.x86_64
 mod_http2-2.0.27-1.amzn2023.0.3.x86_64
 nginx-filesystem-1:1.26.3-1.amzn2023.0.1.noarch
 php8.3-8.3.16-1.amzn2023.0.1.x86_64
 php8.3-common-8.3.16-1.amzn2023.0.1.x86_64
 php8.3-mbstring-8.3.16-1.amzn2023.0.1.x86_64
 php8.3-opcache-8.3.16-1.amzn2023.0.1.x86_64
 php8.3-process-8.3.16-1.amzn2023.0.1.x86_64
 php8.3-xml-8.3.16-1.amzn2023.0.1.x86_64
 apr-util-1.6.3-1.amzn2023.0.1.x86_64
 generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch
 httpd-core-2.4.62-1.amzn2023.x86_64
 httpd-tools-2.4.62-1.amzn2023.x86_64
 libsodium-1.0.19-4.amzn2023.x86_64
 mailcap-2.1.49-3.amzn2023.0.3.noarch
 mariadb-connector-c-config-3.1.13-1.amzn2023.0.3.noarch
 mariadb105-common-3:10.5.25-1.amzn2023.0.1.x86_64
 mod_lua-2.4.62-1.amzn2023.x86_64
 perl-Sys-Hostname-1.23-477.amzn2023.0.6.x86_64
 php8.3-cli-8.3.16-1.amzn2023.0.1.x86_64
 php8.3-fpm-8.3.16-1.amzn2023.0.1.x86_64
 php8.3-mysqld-8.3.16-1.amzn2023.0.1.x86_64
 php8.3-pdo-8.3.16-1.amzn2023.0.1.x86_64
 php8.3-sodium-8.3.16-1.amzn2023.0.1.x86_64
=====
Complete!
```

7. Connect to the RDS Database using the MySQL client

```
[ec2-user@ip-172-31-6-223 ~]$ mysql -h taskonedb.cluster-cpw188a8e8aa.ap-south-1.rds.amazonaws.com -u admin -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 82
Server version: 8.0.32 Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MySQL [(none)]> 
```

8. Create tables and insert data

```
MySQL [(none)]> CREATE DATABASE testdb;
Query OK, 1 row affected (0.003 sec)
```

```
MySQL [(none)]> USE testdb;
Database changed
```

```
MySQL [testdb]> CREATE TABLE users (
    ->     id INT AUTO_INCREMENT PRIMARY KEY,
    ->     name VARCHAR(100),
    ->     email VARCHAR(100)
    -> );
Query OK, 0 rows affected (0.019 sec)

MySQL [testdb]> INSERT INTO users (name, email) VALUES ("John Doe", "john.doe@example.com");
Query OK, 1 row affected (0.006 sec)
```

```
MySQL [testdb]> INSERT INTO users (name, email) VALUES ("John Doe", "john.doe@example.com");
Query OK, 1 row affected (0.006 sec)

MySQL [testdb]> SELECT * FROM users;
+----+-----+-----+
| id | name | email |
+----+-----+-----+
| 1 | John Doe | john.doe@example.com |
+----+-----+-----+
1 row in set (0.002 sec)

MySQL [testdb]> 
```

TASK 2

1. Create a Virtual Private Cloud (VPC)

The screenshot shows the AWS search interface with the query 'vpd' entered. The results are categorized under 'Services' and 'Features'.

Services

- VPC**: Isolated Cloud Resources. Top features: Your VPCs, Subnet, Route table, Internet gateway, Egress-only internet gateways.
- AWS Firewall Manager**: Central management of firewall rules.
- Detective**: Investigate and Analyze potential security issues.

Features

- Dashboard**: VPC feature.
- Route 53 VPCs**: Route 53 feature.
- VPC Reachability Analyzer**: VPC feature.

At the bottom, there are buttons for 'Were these results helpful?' (Yes or No) and a 'Resources' link for a focused search.

The screenshot shows the AWS VPC dashboard. On the left, there's a sidebar with navigation links like 'Virtual private cloud', 'Security', 'PrivateLink and Lattice', and 'AWS Network Manager'. The main area displays 'Resources by Region' for the Asia Pacific region, showing counts for VPCs, Subnets, Route Tables, Internet Gateways, Egress-only Internet Gateways, DHCP option sets, NAT Gateways, VPC Peering Connections, Network ACLs, Security Groups, Customer Gateways, and Virtual Private Gateways. There are also buttons for 'Create VPC' and 'Launch EC2 Instances'. On the right, there are sections for 'Service Health', 'Settings', 'Additional Information', and 'AWS Network Manager'.

The screenshot shows the AWS VPC creation process. The top navigation bar includes the AWS logo, a search bar, and links for 'Option+S' and 'Shravasti'. Below the navigation, the breadcrumb trail reads 'VPC > Your VPCs > Create VPC'. The main title 'Create VPC' has an 'Info' link. A descriptive text states: 'A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances. Mouse over a resource to highlight the related resources.'

VPC settings

Resources to create [Info](#)
Create only the VPC resource or the VPC and other networking resources.

VPC only VPC and more

Name tag auto-generation [Info](#)
Enter a value for the Name tag. This will be used to auto-generate Name tags for all resources in the VPC.
 Auto-generate
shravsvpc

IPv4 CIDR block [Info](#)
Determine the starting IP and the size of your VPC using CIDR notation.
10.0.0.0/16 65,536 IPs
CIDR block size must be between /16 and /28.

IPv6 CIDR block [Info](#)
 No IPv6 CIDR block Amazon-provided IPv6 CIDR block

Tenancy [Info](#)
Default

Number of Availability Zones (AZs) [Info](#)
Choose the number of AZs in which to provision subnets. We recommend at least two AZs for high availability.

Preview

VPC [Show details](#)
Your AWS virtual network
shravsvpc-vpc

Subnets (4)
Subnets within this VPC

- ap-south-1a**
shravsvpc-subnet-public1-ap-south-1a
- ap-south-1a**
shravsvpc-subnet-private1-ap-south-1a
- ap-south-1b**
shravsvpc-subnet-public2-ap-south-1b
- ap-south-1b**
shravsvpc-subnet-private2-ap-south-1b

aws   Search [Option+S]     Asia Pacific (Mumbai) ▾ Shravasti ▾

VPC > Your VPCs > Create VPC   

Number of public subnets 

The number of public subnets to add to your VPC. Use public subnets for web applications that need to be publicly accessible over the internet.

0 2 4

Number of private subnets 

The number of private subnets to add to your VPC. Use private subnets to secure backend resources that don't need public access.

0 2 4

► Customize subnets CIDR blocks

NAT gateways (\$) 

Choose the number of Availability Zones (AZs) in which to create NAT gateways. Note that there is a charge for each NAT gateway

None In 1 AZ 1 per AZ

VPC endpoints 

Endpoints can help reduce NAT gateway charges and improve security by accessing S3 directly from the VPC. By default, full access policy is used. You can customize this policy at any time.

None S3 Gateway

DNS options 

- Enable DNS hostnames
- Enable DNS resolution

► Additional tags

Cancel
Preview code
Create VPC

Screenshot of the AWS VPC Create VPC workflow page showing a success message and a list of completed steps.

Create VPC workflow

Success

Details

- ✓ Create VPC: vpc-065a1c6ea8d9a7ffe
- ✓ Enable DNS hostnames
- ✓ Enable DNS resolution
- ✓ Verifying VPC creation: vpc-065a1c6ea8d9a7ffe
- ✓ Create subnet: subnet-0b84faf73ff290be8
- ✓ Create subnet: subnet-0289b97517501d2c5
- ✓ Create subnet: subnet-053c1a07d08d28953
- ✓ Create subnet: subnet-0f6a084b54e65344
- ✓ Create internet gateway: igw-0389771ea0388130b
- ✓ Attach internet gateway to the VPC
- ✓ Create route table: rtb-0e36d1bd4247d9fc1
- ✓ Create route
- ✓ Associate route table
- ✓ Associate route table
- ✓ Create route table: rtb-0859fadec51b148d
- ✓ Associate route table
- ✓ Create route table: rtb-01bd27fe011ad9344
- ✓ Associate route table
- ✓ Verifying route table creation

View VPC

Screenshot of the AWS VPC dashboard for the created VPC (vpc-065a1c6ea8d9a7ffe).

VPC dashboard

vpc-065a1c6ea8d9a7ffe / shravsvpc-vpc

Details

VPC ID: vpc-065a1c6ea8d9a7ffe	State: Available	Block Public Access: Off	DNS hostnames: Enabled
DNS resolution: Enabled	Tenancy: default	DHCP option set: dopt-0b874dee14e2fa899	Main route table: rtb-0354a7c290f8c7f02
Main network ACL: acl-0dd45e5df28959c8a	Default VPC: No	IPv4 CIDR: 10.0.0.0/16	IPv6 pool: -
IPv6 CIDR (Network border group): -	Network Address Usage metrics: Disabled	Route 53 Resolver DNS Firewall rule groups: -	Owner ID: 619071313125

Resource map

Subnets (4)

- ap-south-1a: shravsvpc-subnet-public1-ap-south-1a
- ap-south-1a: shravsvpc-subnet-private1-ap-south-1a
- ap-south-1b: shravsvpc-subnet-public2-ap-south-1b
- ap-south-1b: shravsvpc-subnet-private2-ap-south-1b

Route tables (4)

- shravsvpc-rtb-private1-ap-south-1a
- shravsvpc-rtb-public
- shravsvpc-rtb-private2-ap-south-1b
- rtb-0354a7c290f8c7f02

Network connections (1)

- shravsvpc-igw

2. Edit Inbound Rules for both the default security groups

The screenshot shows the AWS VPC Dashboard. On the left sidebar, there are sections for VPC dashboard, Virtual private cloud, Security, PrivateLink, DNS firewall, and Network Firewall. The main content area displays the following features:

- Features** section:
 - Security groups** (EC2 feature)
 - Security groups** (VPC feature)
 - Input security groups** (MediaLive feature)
- Services** section:
 - EC2** (Virtual Servers in the Cloud)
 - IAM** (Manage access to AWS resources)
 - AWS Firewall Manager** (Central management of firewall rules)
- Resources** section:
 - Introducing resource search**: A tooltip explaining how to show cross-region resources for your account in search results.
 - Documentation** section for Security groups.

On the right side, there are detailed views of network components:

- Public Access** (option set: rtb-0354a7c290f8c7f02)
- DNS hostnames** (Enabled)
- Main route table** (rtb-0354a7c290f8c7f02)
- IPv6 pool** (disabled)
- Owner ID** (619071313125)
- Route tables (4)** (shrvspc-rtb-private1-ap-south-1a, shrvspc-rtb-public, shrvspc-rtb-private2-ap-south-1b, rtb-0354a7c290f8c7f02)
- Network connections (1)** (shrvspc-igw)

The screenshot shows the AWS EC2 Security Groups page. The left sidebar includes sections for EC2, Instances, Images, Elastic Block Store, Network & Security, Load Balancing, and Auto Scaling. The main content area displays the following table:

Name	Security group ID	Security group name	VPC ID	Description
-	sg-07690c7a65ae5d1c1	default	vpc-065a1c6ea8d9a7ffe	default VPC security group
-	sg-000f728cce01d8879	default	vpc-007543eb92d582d02	default VPC security group

At the top right, there are buttons for Actions, Export security groups to CSV, and Create security group. The bottom right shows navigation icons and a footer with copyright information.

Inbound rules

Type	Protocol	Port range	Source	Description - optional
SSH	TCP	22	Anywhere	0.0.0.0/0
HTTP	TCP	80	Anywhere	0.0.0.0/0
SMB	TCP	445	Anywhere	0.0.0.0/0

Add rule

Inbound rules

Type	Protocol	Port range	Source	Description - optional
MySQL/Aurora	TCP	3306	Anywhere	0.0.0.0/0

Add rule

3. Create an RDS Database Instance

Services

- Aurora and RDS
- Database Migration Service
- Kinesis

Features

- Database Insights
- Reserved instances
- Proxies

Resources / for a focused search

Introducing resource search

Enable to show cross-region resources for your account in search results. Takes less than 5 minutes to set up.

Documentation

Amazon RDS

VPC ID	Description
vpc-065a1c6ea8d9a7ffe	default VPC security group
vpc-007543eb92d582d02	default VPC security group
vpc-007543eb92d582d02	private security group
vpc-007543eb92d582d02	public security group

a. Create a subnet group for the instance

The screenshot shows the AWS RDS Subnet groups page. On the left, there's a sidebar with options like Dashboard, Databases, Query Editor, Performance insights, Snapshots, Exports in Amazon S3, Automated backups, Reserved instances, and Proxies. Under Subnet groups, it lists Parameter groups, Option groups, Custom engine versions, and Zero-ETL integrations. Below that are Events, Event subscriptions, Recommendations (0), and Certificate update. The main content area shows a table titled "Subnet groups (1)". The table has columns for Name, Description, Status, and VPC. One row is shown: "default-vpc-007543eb92d582d02" with a status of "Complete" and VPC "vpc-007543eb92d582d02". There are buttons for Edit, Delete, and Create DB subnet group.

The screenshot shows the "Create DB subnet group" wizard. Step 1: Subnet group details. It asks for a Name (private-rds-subnet) and a Description (private rds subnet). It also asks to choose a VPC (shrawansvc-vpc (vpc-065a1c6ea8d9a7fe)). Step 2: Add subnets. It shows availability zones (ap-south-1a, ap-south-1b, ap-south-1c) and subnets (No subnets added to this group). A note says "For Multi-AZ DB clusters, you must select 3 subnets in 3 different Availability Zones." Step 3: Subnets selected (0). The table shows columns for Availability zone, Subnet name, Subnet ID, and CIDR block. The message "No subnets added to this group" is displayed.

Screenshot of the AWS RDS 'Create DB subnet group' wizard.

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.
shravsvpc-vpc (vpc-065a1c6ea8d9a7ffe)
4 Subnets, 2 Availability Zones

Add subnets

Availability Zones:
Choose the Availability Zones that include the subnets you want to add.
Choose an availability zone
ap-south-1a X ap-south-1b X ap-south-1c X

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

shravsvpc-subnet-public1-ap-south-1a Subnet ID: subnet-0b84faf73ff290be8 CIDR: 10.0.0.0/20	shravsvpc-subnet-private1-ap-south-1a Subnet ID: subnet-053c1a07d08d28953 CIDR: 10.0.128.0/20
shravsvpc-subnet-public2-ap-south-1b Subnet ID: subnet-0289b97517501d2c5 CIDR: 10.0.16.0/20	shravsvpc-subnet-private2-ap-south-1b Subnet ID: subnet-0f6a084b54e065344 CIDR: 10.0.144.0/20

Subnets selected (4)

Availability zone	Subnet name	Subnet ID	CIDR block
ap-south-1a	shravsvpc-subnet-public1-ap-south-1a	subnet-0b84faf73ff290be8	10.0.0.0/20
ap-south-1a	shravsvpc-subnet-private1-ap-south-1a	subnet-053c1a07d08d28953	10.0.128.0/20
ap-south-1b	shravsvpc-subnet-public2-ap-south-1b	subnet-0289b97517501d2c5	10.0.16.0/20
ap-south-1b	shravsvpc-subnet-private2-ap-south-1b	subnet-0f6a084b54e065344	10.0.144.0/20

Create

Screenshot of the AWS RDS 'Subnet groups' page.

Amazon RDS

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

Subnet groups

- Parameter groups
- Option groups
- Custom engine versions
- Zero-ETL integrations **New**

Events

Recommendations 0

Subnet groups (2)

Name	Description	Status	VPC
default-vpc-007545eb92d582d02	Created from the RDS Management Console	Complete	vpc-007545eb92d582d02
private-rds-subnet	private rds subnet	Complete	vpc-065a1c6ea8d9a7ffe

4. Create an Amazon Linux EC2 instance

The screenshot shows the AWS EC2 home page. On the left, there's a sidebar with links for RDS, Amazon RDS, Subnet groups, Documentation, Events, and Recommendations. The main content area has sections for Services (EC2, EC2 Image Builder, EC2 Global View), Features (Dashboard, EC2 Instances, AMIs), and Resources (Introducing resource search, Amazon EC2 instance types by Region). A modal window titled 'Introducing resource search' provides information about enabling cross-region search. On the right, a separate browser window shows a list of DB subnet groups with columns for Status, VPC, and ID.

1. Create a key pair

The screenshot shows the AWS Lambda console. In the background, there's a summary of an existing Lambda function: 'Amazon Linux 2023.6 AMI 2023.6.20250503.0.x86_64 HVM kernel-6.1'. The 'Create key pair' dialog is in the foreground, prompting for a key pair name ('ed25519') and key pair type ('RSA'). It also shows options for private key file format ('.pem' or '.ppk') and a note about storing the private key securely. To the right, there's a 'Summary' section with details like the number of instances (1), software image (AMI), virtual server type (t2.micro), and storage (1 volume - 8 GiB). A 'Free tier' note is visible. At the bottom right of the dialog is a 'Create key pair' button.

2. Select existing security group

The screenshot shows the 'Launch an instance' wizard in the AWS EC2 console. The current step is 'Select security group'. A modal window is open, showing a list of security groups. The 'Select existing security group' button is highlighted with a red box. The modal also contains a note about free tier storage usage and a link to 'Add new volume'.

The screenshot shows the 'Launch an instance' wizard in the AWS EC2 console. The current step is 'Select security group'. A green success message at the top states 'Successfully initiated launch of instance (i-0ab2b0285251dd47e)'. Below it, there is a 'Launch log' button. A 'Next Steps' section is displayed with several cards:

- Create billing and free tier usage alerts**: Manage costs and avoid surprise bills. Includes a 'Create billing alerts' button.
- Connect to your instance**: Once your instance is running, log into it from your local computer. Includes a 'Connect to instance' button.
- Connect an RDS database**: Configure the connection between an EC2 instance and a database. Includes a 'Connect an RDS database' button.
- Create EBS snapshot policy**: Create a policy that automates the creation, retention, and deletion of EBS snapshots. Includes a 'Create EBS snapshot policy' button.
- Manage detailed monitoring**: Enable or disable detailed monitoring for the instance. Includes a 'Manage detailed monitoring' button.
- Create Load Balancer**: Create a application, network gateway or classic Elastic Load Balancer. Includes a 'Create Load Balancer' button.
- Create AWS budget**: AWS Budgets allows you to create budgets, forecast spend, and take action on your costs and usage from a single location. Includes a 'Create AWS budget' button.
- Manage CloudWatch alarms**: Create or update Amazon CloudWatch alarms for the instance. Includes a 'Manage CloudWatch alarms' button.

A 'View all instances' button is located in the bottom right corner of the 'Next Steps' section.

5. Create an RDS Database

The screenshot shows the Amazon RDS Databases page. On the left, there's a sidebar with options like Dashboard, Databases (selected), Query Editor, Performance insights, Snapshots, Exports in Amazon S3, Automated backups, Reserved instances, Proxies, Subnet groups, Parameter groups, Option groups, Custom engine versions, Zero-ETL integrations, Events, Event subscriptions, and Recommendations. The main area is titled 'Databases (0)' and shows a table with columns: DB identifier, Status, Role, Engine, Region ..., Size, and Recommendations. A message says 'No instances found'. At the top right, there are buttons for 'Group resources', 'Modify', 'Actions', 'Restore from S3' (highlighted in orange), and 'Create database'.

1. Use standard create method and MySQL engine

The screenshot shows the 'Create database' page. It starts with a section 'Choose a database creation method' with 'Standard create' selected (blue border). There's also an 'Easy create' option. Below this is the 'Engine options' section, which lists several engine types with icons: Aurora (MySQL Compatible) (selected), Aurora (PostgreSQL Compatible), MySQL, PostgreSQL, MariaDB, Oracle, Microsoft SQL Server, and IBM Db2. To the right of the engine list is a sidebar for 'Aurora MySQL-Compatible Edition' which includes a brief description and a bulleted list of features. At the bottom, there's an 'Engine version' dropdown set to 'Aurora MySQL 3.05.2 (compatible with MySQL 8.0.32) - default for major version 8.0'. The footer includes CloudShell, Feedback, and standard AWS links.

2. Choose Test template; Select self managed credentials

The screenshot shows the AWS RDS 'Create database' wizard. In the 'Templates' step, the 'Dev/Test' option is selected. In the 'Settings' step, the 'DB cluster identifier' is set to 'shrawansdatabase'. Under 'Credentials Settings', 'Self managed' is selected for the master password. The 'Master password' field contains a strong password, and the 'Confirm master password' field is identical. The 'Cluster storage configuration' step is shown at the bottom.

3. Instance Configuration → Burstable Classes

The screenshot shows the AWS RDS 'Create database' wizard. In the 'Instance configuration' step, the 'Aurora Standard' storage configuration is selected. Under 'DB instance class', 'Burstable classes (includes t classes)' is selected, and the 'db.t3.medium' option is chosen. In the 'Availability & durability' step, 'Multi-AZ deployment' is set to 'Don't create an Aurora Replica'. The 'Connectivity' step is shown at the bottom.

4. Connectivity → Don't connect to an EC2 instance; Select VPC

Compute resource
 Don't connect to an EC2 compute resource
 Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

Connect to an EC2 compute resource
 Set up a connection to an EC2 compute resource for this database.

Network type [Info](#)
 To use dual-stack mode, make sure that you associate an IPv6 CIDR block with a subnet in the VPC you specify.

IPv4
 Your resources can communicate only over the IPv4 addressing protocol.

Dual-stack mode
 Your resources can communicate over IPv4, IPv6, or both.

Virtual private cloud (VPC) [Info](#)
 Choose the VPC. The VPC defines the virtual networking environment for this DB cluster.

shrawvpc-vpc (vpc-065a1c6ea8d9a7ffe)
 4 Subnets, 2 Availability Zones

Only VPCs with a corresponding DB subnet group are listed.

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

DB subnet group [Info](#)
 Choose the DB subnet group. The DB subnet group defines which subnets and IP ranges the DB cluster can use in the VPC that you selected.

private-rds-subnet
 4 Subnets, 2 Availability Zones

Public access [Info](#)
 Yes
 RDS assigns a public IP address to the cluster. Amazon EC2 instances and other resources outside of the VPC can connect to your cluster. Resources inside the VPC can also connect to the cluster. Choose one or more VPC security groups that specify which resources can connect to the cluster.

No
 RDS doesn't assign a public IP address to the cluster. Only Amazon EC2 instances and other resources inside the VPC can connect to your cluster. Choose one or more VPC security groups that specify which resources can connect to the cluster.

VPC security group (firewall) [Info](#)
 Choose one or more VPC security groups to allow access to your database. Make sure 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 one or more options

Aurora MySQL-Compatible Edition
 Aurora MySQL is Amazon's enterprise-class MySQL-compatible database.

Aurora MySQL offers:

- Up to five times the throughput of MySQL Community Edition
- Up to 128 TB of autoscaling SSD storage
- Six-way replication across three Availability Zones
- Up to 15 read replicas with replica lag under 10-ms
- Automatic monitoring with failover

5. Create database

VPC security group (firewall) [Info](#)
 Choose one or more VPC security groups to allow access to your database. Make sure 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 one or more options

default X

Availability Zone [Info](#)
 ap-south-1a

RDS Proxy
 RDS Proxy is a fully managed, highly available database proxy that improves application scalability, resiliency, and security.

Create an RDS Proxy [Info](#)
 RDS automatically creates an IAM role and a Secrets Manager secret for the proxy. RDS Proxy has additional costs. For more information, see [Amazon RDS Proxy pricing](#).

Certificate authority - optional [Info](#)
 Using a server certificate provides an extra layer of security by validating that the connection is being made to an Amazon database. It does so by checking the server certificate that is automatically installed on all databases that you provision.

rds-ca-rsa2048-g1 (default)
 Expiry: May 20, 2061

If you don't select a certificate authority, RDS chooses one for you.

Additional configuration

Read replica write forwarding
 Turn on local write forwarding [Info](#)
 Issues write operations from reader DB instances within the same DB cluster.

Tags - optional
 A tag consists of a case-sensitive key-value pair.
 No tags associated with the resource.

Add new tag

Aurora MySQL-Compatible Edition
 Aurora MySQL is Amazon's enterprise-class MySQL-compatible database.

Aurora MySQL offers:

- Up to five times the throughput of MySQL Community Edition
- Up to 128 TB of autoscaling SSD storage
- Six-way replication across three Availability Zones
- Up to 15 read replicas with replica lag under 10-ms
- Automatic monitoring with failover

The screenshot shows the 'Create database' wizard for AWS RDS. The first step, 'Configure your new database', is displayed. Key configuration settings include:

- OS metrics granularity:** Set to '60 seconds'.
- Monitoring role for OS metrics:** Set to 'default'.
- Log exports:** Options for Audit log, Error log, General log, iam-db-auth-error log, instance log, and Slow query log are listed.
- IAM role:** A service-linked role for publishing logs to CloudWatch Logs is selected.
- Additional configuration:** Includes options for encryption, failover, backup, backtrack, maintenance, CloudWatch Logs, and delete protection.
- Estimated monthly costs:** Total cost is 86.14 USD.
- Disclaimer:** A note states: "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."

The second step, 'Review and create', is shown with the 'Create database' button highlighted.

The screenshot shows the 'Databases' page in the AWS RDS console. The database 'shravsrdsdatabase' has been successfully created, as indicated by a green success message: 'Successfully created database shravsrdsdatabase'. The database details table shows:

DB identifier	Status	Role	Engine	Region	Size	Recommendations	CPU
shravsrdsdatabase	Available	Regional	Aurora My...	ap-south-1	1 instance	-	-
shravsrdsdatabase-instance-1	Creating	Writer ins...	Aurora My...	ap-south-1a	db.t3.medium	-	-

6. Change key permissions and connect to ssh instance

```
shravastiohol@Shravasti-MacBook-Pro ~ % ssh -i ~/Downloads/shravslinuxkey.pem ec2-user@ec2-15-207-84-165.ap-south-1.compute.amazonaws.com
The authenticity of host 'ec2-15-207-84-165.ap-south-1.compute.amazonaws.com (15.207.84.165)' can't be established.
ED25519 key fingerprint is SHA256:3hkiEYHbifvEoe/jtaPtW+WMhAaY99UUnNCPoFLuXyk.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-15-207-84-165.ap-south-1.compute.amazonaws.com' (ED25519) to the list of known hosts.

      _#
     ~\_\#\#\#_          Amazon Linux 2023
     ~~\_\#\#\#\\
     ~~ \#\#\|
     ~~   \#/ ___  https://aws.amazon.com/linux/amazon-linux-2023
     ~~   V~^ '->
     ~~~   /
     ~~_.-./
     _/_,-/
     /m',.

Last login: Wed Mar 19 09:12:27 2025 from 13.233.177.4
[ec2-user@ip-10-0-8-139 ~]$
```

7. Update

```
[ec2-user@ip-10-0-8-139 ~]$ sudo dnf update -y
Amazon Linux 2023 Kernel Livepatch repository
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-10-0-8-139 ~]$
```

8. Install Apache Web Server and MariaDB

```
[ec2-user@ip-10-0-8-139 ~]$ sudo dnf install -y httpd php php-mysqli mariadb105
Last metadata expiration check: 0:00:50 ago on Wed Mar 19 09:27:14 2025.
Dependencies resolved.
=====
Package           Architecture Version       Repository      Size
=====
Installing:
httpd            x86_64      2.4.62-1.amzn2023   amazonlinux    48 k
mariadb105       x86_64      3:10.5.25-1.amzn2023.0.1  amazonlinux    1.6 M
php8.3           x86_64      8.3.16-1.amzn2023.0.1  amazonlinux    10 k
php8.3-mysqld   x86_64      8.3.16-1.amzn2023.0.1  amazonlinux   148 k
Installing dependencies:
apr              x86_64      1.7.5-1.amzn2023.0.4  amazonlinux   129 k
apr-util         x86_64      1.6.3-1.amzn2023.0.1  amazonlinux    98 k
generic-logos-httdp  noarch    18.0.0-12.amzn2023.0.3  amazonlinux   19 k
httpd-core       x86_64      2.4.62-1.amzn2023   amazonlinux   1.4 M
httpd-filesystem noarch    2.4.62-1.amzn2023   amazonlinux   14 k
httpd-tools      x86_64      2.4.62-1.amzn2023   amazonlinux   81 k
libbrotli        x86_64      1.0.9-4.amzn2023.0.2  amazonlinux   315 k
libsodium         x86_64      1.0.19-4.amzn2023   amazonlinux   176 k
libxslt          x86_64      1.1.34-5.amzn2023.0.2  amazonlinux   241 k
mailcap          noarch    2.1.49-3.amzn2023.0.3  amazonlinux   33 k
mariadb-connector-c  x86_64      3.1.13-1.amzn2023.0.3  amazonlinux   196 k
mariadb-connector-c-config  noarch    3.1.13-1.amzn2023.0.3  amazonlinux   9.2 k
mariadb105-common x86_64      3:10.5.25-1.amzn2023.0.1  amazonlinux   29 k
nginx-filesystem noarch    1:1.26.3-1.amzn2023.0.1  amazonlinux   9.6 k
perl-Sys-Hostname x86_64      1.23-477.amzn2023.0.6  amazonlinux   18 k
php8.3-cli       x86_64      8.3.16-1.amzn2023.0.1  amazonlinux   3.7 M
php8.3-common    x86_64      8.3.16-1.amzn2023.0.1  amazonlinux   744 k
php8.3-pdo       x86_64      8.3.16-1.amzn2023.0.1  amazonlinux   89 k
php8.3-process   x86_64      8.3.16-1.amzn2023.0.1  amazonlinux   45 k
php8.3-xml       x86_64      8.3.16-1.amzn2023.0.1  amazonlinux   154 k
Installing weak dependencies:
=====

```

Complete!

```
[ec2-user@ip-10-0-8-139 ~]$
```

9. Start the web server

```
Complete.
[ec2-user@ip-10-0-8-139 ~]$ sudo systemctl start httpd
[ec2-user@ip-10-0-8-139 ~]$ sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[ec2-user@ip-10-0-8-139 ~]$
```

10. Set file permissions for Apache Web Server

```
[ec2-user@ip-10-0-8-139 ~]$ sudo usermod -a -G apache ec2-user
[ec2-user@ip-10-0-8-139 ~]$
```

11. Exit and login again to access groups

```
[ec2-user@ip-10-0-8-139 ~]$ exit
logout
Connection to ec2-15-207-84-165.ap-south-1.compute.amazonaws.com closed.
shravastiohol@Shravastiis-MacBook-Pro ~ %
```

12. Change the permissions of the /var/www directory

```
connection to ec2-15-207-84-165.ap-south-1.compute.amazonaws.com closed.
shravastiohol@Shravasti-MacBook-Pro ~ % ssh -i ~/Downloads/shravslinuxkey.pem ec2-user@ec2-15-207-84-165.ap-south-1.compute.amazonaws.com
      _#
     ~\_\####_          Amazon Linux 2023
     ~~\_\#####\
     ~~ \###|
     ~~  \#/ ___ https://aws.amazon.com/linux/amazon-linux-2023
     ~~   V~' '-->
     ~~~ /
     ~~.~.~/
     ~~/_/_/
     _/m/`_
Last login: Wed Mar 19 09:26:47 2025 from 117.250.240.128
[ec2-user@ip-10-0-8-139 ~]$
```

```
[ec2-user@ip-10-0-8-139 ~]$ sudo chown -R ec2-user:apache /var/www
[ec2-user@ip-10-0-8-139 ~]$ sudo chmod 2775 /var/www
[ec2-user@ip-10-0-8-139 ~]$
```

```
[ec2-user@ip-10-0-8-139 ~]$ find /var/www -type d -exec sudo chmod 2775 {} \;
[ec2-user@ip-10-0-8-139 ~]$ find /var/www -type f -exec sudo chmod 0664 {} \;
[ec2-user@ip-10-0-8-139 ~]$
```

13. Create the necessary files

```
[ec2-user@ip-10-0-8-139 ~]$ cd /var/www
[ec2-user@ip-10-0-8-139 www]$ mkdir inc
[ec2-user@ip-10-0-8-139 www]$ cd inc
[ec2-user@ip-10-0-8-139 inc]$ >dbinfo.inc
[ec2-user@ip-10-0-8-139 inc]$ nano dbinfo.inc
[ec2-user@ip-10-0-8-139 inc]$
```

```
GNU nano 5.8                                         dbinfo.inc                                         Modified
<?php
define ('DB_SERVER', 'shravsrdsdatabase.cluster-cpwi88a8e8aa.ap-south-1.rds.amazonaws.com');
define ('DB_USERNAME', 'admin');
define ('DB_PASSWORD', 'database#119900');
define ('DB_DATABASE', 'ccslabdatabase');
?>■

^G Help      ^O Write Out   ^W Where Is   ^K Cut        ^T Execute   ^C Location   M-U Undo
^X Exit      ^R Read File   ^\ Replace    ^U Paste     ^J Justify   ^/ Go To Line M-E Redo
```

```
[ec2-user@ip-10-0-8-139 html]$ cat SamplePage.php
<!DOCTYPE html>
<html>
<head>
    <title>Employee Management</title>
</head>
<body>

<?php

/* Add an employee to the table. */
function AddEmployee($connection, $name, $address) {
    $n = mysqli_real_escape_string($connection, $name);
    $a = mysqli_real_escape_string($connection, $address);
    $query = "INSERT INTO EMPLOYEES (NAME, ADDRESS) VALUES ('$n', '$a');";
    if (!mysqli_query($connection, $query)) {
        echo "<p>Error adding employee data: " . mysqli_error($connection) . "</p>";
    } else {
        echo "<p>Employee added successfully.</p>";
    }
}

/* Check whether the table exists and, if not, create it. */
function VerifyEmployeesTable($connection, $dbName) {
    if (!TableExists("EMPLOYEES", $connection, $dbName)) {
        $query = "CREATE TABLE EMPLOYEES (
            ID int(11) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
            NAME VARCHAR(45) NOT NULL,
            ADDRESS VARCHAR(90) NOT NULL
        );";
        if (!mysqli_query($connection, $query)) {
            echo "<p>Error creating table: " . mysqli_error($connection) . "</p>";
        } else {
```

```
ec2-user@ip-10-0-6-0:~/var/www/html - + SamplePage.php Modified
GNU nano 5.8
</body>
</html>
<?php
/* Add an employee to the table. */
function AddEmployee($connection, $name, $address) {
    $n = mysqli_real_escape_string($connection, $name);
    $a = mysqli_real_escape_string($connection, $address);
    $query = "INSERT INTO EMPLOYEES (NAME, ADDRESS) VALUES ('$n', '$a');";
    if(!mysqli_query($connection, $query)) echo("<p>Error adding employee data.</p>");
}
/* Check whether the table exists and, if not, create it. */
function VerifyEmployeesTable($connection, $dbName) {
    if(!TableExists("EMPLOYEES", $connection, $dbName))
    {
        $query = "CREATE TABLE EMPLOYEES (
            ID int(11) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
            NAME VARCHAR(45),
            ADDRESS VARCHAR(90)
        )";
        if(!mysqli_query($connection, $query)) echo("<p>Error creating table.</p>");
    }
}
/* Check for the existence of a table. */
function TableExists($tableName, $connection, $dbName) {
    $t = mysqli_real_escape_string($connection, $tableName);
    $d = mysqli_real_escape_string($connection, $dbName);
    $checktable = mysqli_query($connection,
        "SELECT TABLE_NAME FROM information_schema.TABLES WHERE TABLE_NAME = '$t'
        AND TABLE_SCHEMA = '$d'");
    if(mysqli_num_rows($checktable) > 0) return true;
    return false;
}
?>■

^G Help      ^O Write Out     ^W Where Is     ^K Cut      ^T Execute     ^C Location     M-U Undo     M-A Set Mark
^X Exit      ^R Read File     ^A Replace     ^U Paste     ^J Justify     ^I Go To Line   M-B Redo     M-G Copy
```

14. Output



Sample Page