

VCUDE

PROJECT-1 : 3-TIRE ARCHITECTURE

A **3-tier architecture** is a structured approach to software design that separates an application into three logical tiers:

1. **Presentation Tier (Web Tier)** – The user-facing layer that handles UI interactions, displays content, and forwards requests to the application tier. It typically consists of web servers (e.g., EC2 instances in an Auto Scaling Group) behind an Application Load Balancer for distributing traffic efficiently.
2. **Application Tier (Logic Tier)** – The intermediary layer that processes user requests, applies business logic, and communicates with the database. It consists of application servers that execute core functionalities and enforce business rules.
3. **Data Tier (Database Tier)** – The storage layer responsible for data management and retrieval. This includes relational database instances like Amazon RDS, ensuring secure data persistence with controlled access.



Select one Region for 3-tire project region like Mumbai (ap-south-1).

Go to AWS Search bar type vpc you can redirect to VPC and then you can start the project.

Step -1:

- i) Create VPC
- ii) Select VPC Only

- iii) Name it as 3-tire
- iv) CIDR as 10.0.0.0/16
- v) Click on create
- vi) Go to action edit vpc
- vii) Enable DNS hostname

You have successfully modified the settings for vpc-05f2e61cf95c0bc79 / 3-tire.

vpc-05f2e61cf95c0bc79 / 3-tire

Details		Actions	
VPC ID	vpc-05f2e61cf95c0bc79	State	Available
DNS resolution	Enabled	Tenancy	default
Main network ACL	acl-0a605cbe23daedff	Default VPC	No
IPv6 CIDR (Network border group)	-	Network Address Usage metrics	Disabled
		Block Public Access	Enabled
		DHCP option set	dopt-014af8031cf927773
		IPv4 CIDR	10.0.0.0/16
		Route 53 Resolver DNS Firewall rule groups	-
		IPv6 pool	-
		Owner ID	039950891473

Resource map

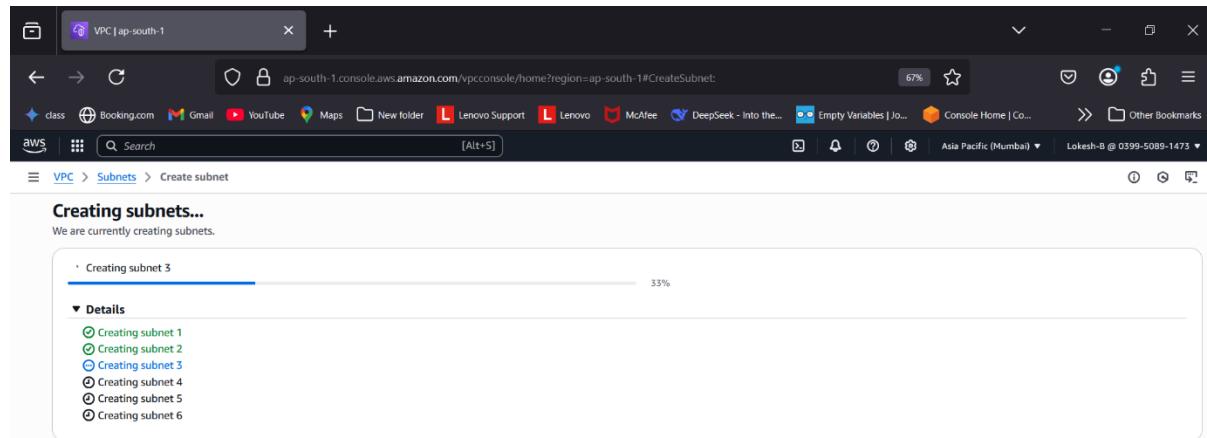
VPC	Subnets	Route tables	Network connections
Show details	(0)	(1)	(0)
Your AWS virtual network	Subnets within this VPC	Route network traffic to resources	Connections to other networks
3-tire		rtb-0008815835c43cfa8	

Step -2:

- I) Go to subnets
- II) Create subnets click on it
- III) Attach VPC for subnets
- IV) Create two public and four private
- V) Public-1a(10.0.1.0/24) in ap-south-1a, public-1b(10.0.2.0/24)in ap-south-1b

VI) Private-1a-1(10.0.3.0/24)in ap-south-1a, private-1b-1(10.0.4.0/24)in ap-south-1b, private-1a-2(10.0.5.0/24)in ap-south-1a, private-1b-2(10.0.6.0/24)in ap-south-1b

VII) Click on create



The screenshot shows the AWS VPC Subnets page. A success message at the top states: "You have successfully created 6 subnets: subnet-0baa9682fc16f9aea, subnet-07493a5b282790193, subnet-0019b4eef09f074bf, subnet-06dba9602e0fbfa8, subnet-084ae0e8dbb41a903, subnet-0408a33013dd46142". The main table lists the following subnets:

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR
private-1a-2	subnet-084ae0e8dbb41a903	Available	vpc-05f2e61cf95c0bc79 3-tire	Off	10.0.5.0/24
public-1b	subnet-07493a5b282790193	Available	vpc-05f2e61cf95c0bc79 3-tire	Off	10.0.2.0/24
public-1a	subnet-0baa9682fc16f9aea	Available	vpc-05f2e61cf95c0bc79 3-tire	Off	10.0.1.0/24
private-1b-1	subnet-06dba9602e0fbfa8	Available	vpc-05f2e61cf95c0bc79 3-tire	Off	10.0.4.0/24
private-1b-2	subnet-0408a33013dd46142	Available	vpc-05f2e61cf95c0bc79 3-tire	Off	10.0.6.0/24
private-1a-1	subnet-0019b4eef09f074bf	Available	vpc-05f2e61cf95c0bc79 3-tire	Off	10.0.3.0/24

Step-3:

- i) Create internet gateway for communication through internet
- ii) Name it as “igw”, and click on create
- iii) Attach it to VPC

The screenshot shows the AWS VPC console interface. On the left, there's a navigation sidebar with sections like 'Virtual private cloud' (Your VPCs, Subnets, Route tables), 'Internet gateways' (Egress-only internet gateways, DHCP option sets, Elastic IPs, Managed prefix lists, NAT gateways, Peering connections), 'Security' (Network ACLs, Security groups), and 'PrivateLink and Lattice' (Getting started, Endpoints, Endpoint services). The main content area has a breadcrumb path: 'VPC > Internet gateways > igw-0ae1f95ade9a2bcd6'. A green notification bar at the top says 'Internet gateway igw-0ae1f95ade9a2bcd6 successfully attached to vpc-05f2e61cf95c0bc79'. Below this, the 'igw-0ae1f95ade9a2bcd6 / igw' page displays details: Internet gateway ID (igw-0ae1f95ade9a2bcd6), State (Attached), VPC ID (vpc-05f2e61cf95c0bc79 | 3-tire), and Owner (039950891473). There's also a 'Tags' section with one entry: Name (igw). On the right, there's an 'Actions' dropdown.

Step-4:

- i) Now goto Route Tables
- ii) Select main VPC and edit it add route
- iii) Edit subnets associations
- iv) Select all subnets , so it can connect to the internet for better interaction

The screenshot shows the AWS VPC Route Tables console. A success message at the top states: "You have successfully updated subnet associations for rtb-0008815835c43cfa8 / 3-tire." The route table details are as follows:

- Route table ID:** rtb-0008815835c43cfa8
- Main:** Yes
- Owner ID:** 039950891473
- VPC:** vpc-05f2e61cf95c0bc79 | 3-tire
- Explicit subnet associations:** 6 subnets
- Edge associations:** -

The **Routes** tab shows two routes:

Destination	Target	Status	Propagated
0.0.0.0/0	igw-0ae1f95ade9a2bcd6	Active	No
10.0.0.16	local	Active	No

The screenshot shows the configuration of explicit subnet associations for the route table. A success message at the top states: "You have successfully updated subnet associations for rtb-0008815835c43cfa8 / 3-tire." The route table details are identical to the first screenshot.

The **Subnet associations** tab shows six explicit subnet associations:

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
private-1a-2	subnet-084ae0e8dbb41a903	10.0.5.0/24	-
public-1b	subnet-07493a5b282790193	10.0.2.0/24	-
public-1a	subnet-0baa9682fc16f9aea	10.0.1.0/24	-
private-1b-1	subnet-06dba9602e0f8bfa8	10.0.4.0/24	-
private-1b-2	subnet-0408a33013d46142	10.0.6.0/24	-
private-1a-1	subnet-0019b4eff09f074bf	10.0.3.0/24	-

The **Subnets without explicit associations** section shows 0 subnets.

Step-5:

- Now create 2 NAT gateway
- Select public access and select subnets public(public-1a, public-1b)
- Allocate elastic ip for both NAT

NAT gateways (2) Info

Name	NAT gateway ID	Connectivity...	State	State message	Primary public I...	Primary private I...
nat-1a	nat-00710d683e754f3b3	Public	Available	-	13.232.78.53	10.0.1.197
nat-1b	nat-0f3e50504e9892aca	Public	Available	-	13.126.109.44	10.0.2.64

Select a NAT gateway

Step-6 :

- i) Navigate to EC2 in Console
- ii) Create instances names as (public-1a, public-1b, private-1a-1, private-1b-1)
- iii) Select application and osi (Amazon Linux 2023)for all instances
- iv) Select t2micro for all instances
- v) Create one key pair and use one key pair for all instances
- vi) Edit network settings
 - a) Select VPC
 - b) Select subnets for specific instances as same as subnets for (ex:- instance as public-1a and subnet as public -1a)
 - c) Enable auto assign ip
- vii) Create one security group and use this for all instances
 - a) SSH-22port, anywhere – 0.0.0.0/0
 - b) HTTP-80port, anywhere – 0.0.0.0/0

viii) Click create instance

The screenshot shows the 'Launch an instance' wizard in the AWS Management Console. The first step, 'Name and tags', has 'public-1a' entered. The second step, 'Application and OS Images (Amazon Machine Image)', shows a catalog of AMIs including Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux, and Debian. An 'Amazon Linux 2023 kernel-5.1 AMI' is selected. The third step, 'Summary', shows the configuration: 1 instance, Amazon Linux 2023 AMI, t2.micro instance type, New security group, and 1 volume(s) - 8 GiB storage. A note about the free tier is displayed. The 'Launch instance' button is at the bottom right.

The screenshot shows the 'Instances' page in the AWS Management Console. It lists four instances: 'public-1b' (Running), 'private-1b-1' (Running), 'private-1a-1' (Running), and 'public-1a' (Running). The 'Launch instances' button is visible at the top right of the table.

Step-7:

- i) Create launch template
 - a) Name as template
 - b) Description as template asg

- c) Application and osi as recently used
- d) Select instance type and key pair
- e) In network settings security group

Create launch template

Creating a launch template allows you to create a saved instance configuration that can be reused, shared and launched at a later time. Templates can have multiple versions.

Launch template name and description

Launch template name - required

template

Template version description

3-tire

Auto Scaling guidance

Select this if you want to use this template with EC2 Auto Scaling

Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

Template tags

Source template

Launch template contents

Specify the details of your launch template below. Leaving a field blank will result in the field not being included in the launch template.

Application and OS Images (Amazon Machine Image) - required

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

Search our full catalog including 1000s of application and OS Images

Recently Quick Start

Recently launched Currently in use

Browse more AMIs

Amazon Machine Image (AMI)

ami2023-ami-2023.7.20250609.0-kernel-6.1-x86_64

Amazon Linux 2023 AMI 2023.7.20250609.0.x86_64 HVM kernel-6.1

Architecture: x86_64 AMI ID: ami-0b09627181cfd5778 Verified provider

Description

Amazon Linux 2023 AMI 2023.7.20250609.0.x86_64 HVM kernel-6.1

Architecture

x86_64 AMI ID: ami-0b09627181cfd5778 Verified provider

Instance type

t2.micro

Family: t2 1 vCPU 1 GiB Memory Current generation: true On-Demand Linux base pricing: 0.0124 USD per Hour

On-Demand Windows base pricing: 0.0124 USD per Hour On-Demand Linux Govt base pricing: 0.0124 USD per Hour

On-Demand Linux Pro base pricing: 0.0142 USD per Hour On-Demand Govt Pro base pricing: 0.0142 USD per Hour

Free tier eligible

All generations

Compare instance types

Key pair (login)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

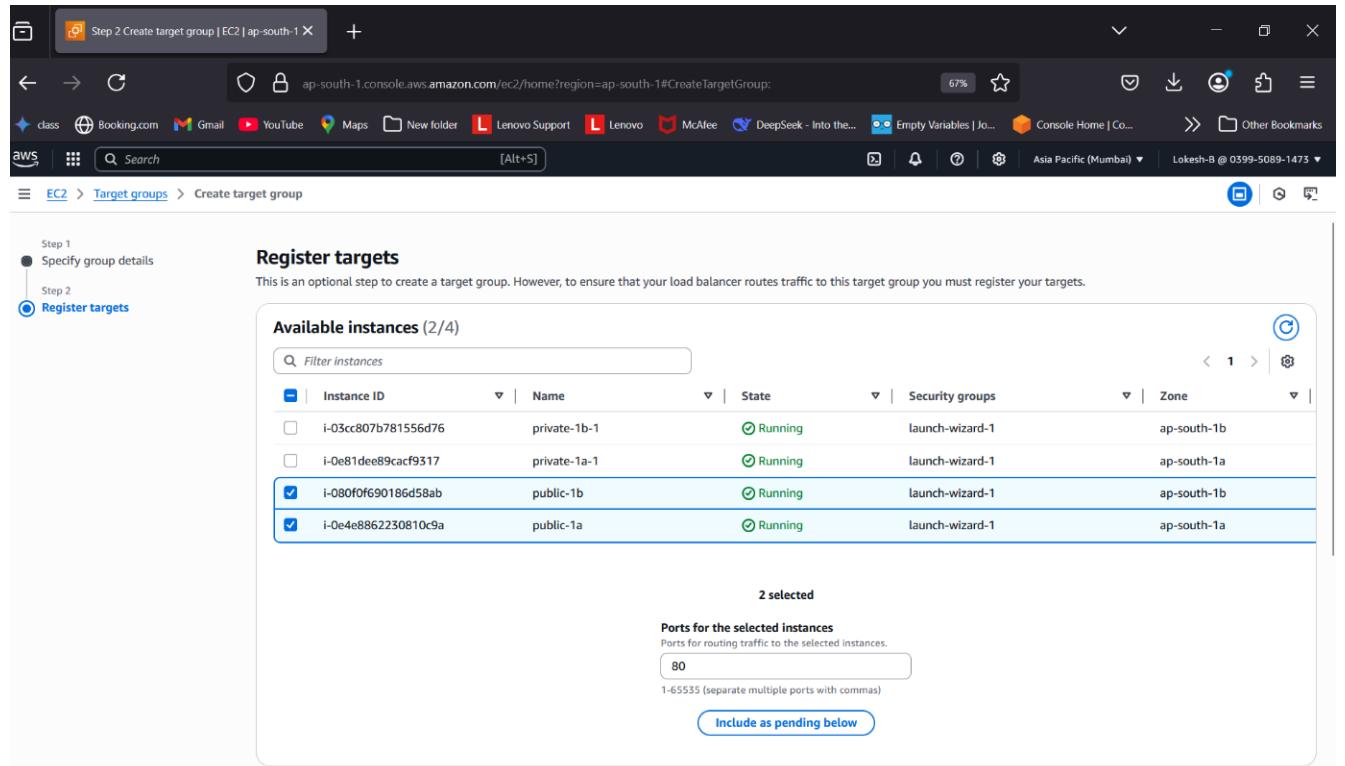
Key pair name

3-tire

Create new key pair

Step-8:

- i) Create targets groups for to show data
- ii) Name it as public target, private target
- iii) In register targets select public target (public-1a, public-1b), private target(private-1a-1, private-1b-1)
- iv) Select include as pending below
- v) Create target group



Total targets	Healthy	Unhealthy	Unused	Initial	Draining
2	0	0	2	0	0

Instance ID	Name	Port	Zone	Health status	Health status details	Administrative override	Override details	Launch time	Anomaly detection
I-080f0690188d58ab	public-1b	80	ap-south-1b (a...)	Unused	Target group is not co...	-	-	June 13, 2023	Normal
I-0e4e886230810a9a	public-1a	80	ap-south-1a (a...)	Unused	Target group is not co...	-	-	June 13, 2023	Normal

Total targets	Healthy	Unhealthy	Unused	Initial	Draining
2	0	0	2	0	0

Instance ID	Name	Port	Zone	Health status	Health status details	Administrative override	Override details	Launch time	Anomaly detection
I-03c8070781556d76	private-1b-1	80	ap-south-1b (a...)	Unused	Target group is not co...	-	-		
I-0e81de89rcaf9517	private-1a-1	80	ap-south-1a (a...)	Unused	Target group is not co...	-	-		

Step-9:

- Create load balancer for balancing load between to servers
- Create load balancer names as (public-lb, private-lb)
- Select target groups for specified load balancer only

CloudFront-Demo

Status: Active

Origin: public-1b (ap-south-1)

Behaviors

- Path: /

Step-10:

- Go to target groups and edit load balancer for private and attach target existing for load balancer

The screenshot shows the AWS EC2 Target groups page. On the left, there's a sidebar with navigation links like Instances, Images, Elastic Block Store, Network & Security, Load Balancing, Auto Scaling, and Settings. The main area displays 'Target groups (1/2)'. It lists two target groups: 'private' and 'public'. The 'private' target group is selected. A tooltip for the 'private' target group states: 'Target group isn't associated to a load balancer. There is no load balancer configured to route traffic to this target group. You can configure an Application Load Balancer with a listener that routes traffic to this target group.' Below this, there are buttons to 'Associate with a new load balancer' or 'Associate with an existing load balancer'. The 'public' target group is associated with a VPC endpoint named 'public-eb'. The 'Details' tab for the 'private' target group shows the following configuration:

Target type	Protocol : Port	Protocol version
Instance	HTTP: 80	HTTP1
IP address type	Load balancer IPv4	VPC vpc-05f2e61cf95c0bc79
Total targets	2 0 Healthy 0 Anomalous	0 Unused 0 Initial 0 Draining

Step-11:

- i) Create Auto scaling for both load balancer
- ii) Create asg and attach created launch template
- iii) Select instance launch option
- iv) Attach existing load balancers for both asg show in fig above
- v) Create asg
- vi) Now you can see desired instance launched in instances
- vii) Edit it and name it

Create Auto Scaling group | EC2 | ap-south-1

ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#CreateAutoScalingGroup:

50%

class Booking.com Gmail YouTube Maps New folder Lenovo Support Lenovo McAfee DeepSeek - Into the... Empty Variables | Jo... Console Home | Co... Other Bookmarks Asia Pacific (Mumbai) Lokesh-B @ 0399-5089-1473

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1 Choose launch template

- Choose launch template
- Choose instance launch options
- Step 3 - optional
- Integrate with other services
- Step 4 - optional
- Configure group size and scaling
- Step 5 - optional
- Add notifications
- Step 6 - optional
- Add tags
- Step 7 Review

Choose launch template Info
Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group.

Name
Auto Scaling group name
Enter a name to identify the group.
public-asg
Must be unique to this account in the current Region and no more than 255 characters.

Launch template Info
For accounts created after May 1, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.

Launch template
Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.
template

Create a launch template version
Version
Default (1)

Create a launch template version

Description 3-tire	Launch template template lt-0c59425d4025759d5	Instance type t2.micro
AMI ID ami-0b09627181c8d5778	Security groups -	Request Spot Instances No
Key pair name 3-tire	Security group IDs sg-0456e7753e948cf1	

Additional details

Storage (volumes)
Date created
Fri Jun 13 2025 15:14:43 GMT+0530 (India Standard Time)

CloudShell Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences ENG 15:22

Step 2 Choose instance launch options

- Step 3 - optional
- Integrate with other services
- Step 4 - optional
- Configure group size and scaling
- Step 5 - optional
- Add notifications
- Step 6 - optional
- Add tags
- Step 7 Review

Instance type requirements Info
You can keep the same instance attributes or instance type from your launch template, or you can choose to override the launch template by specifying different instance attributes or manually adding instance types.

Override launch template

Launch template
template
lt-0c59425d4025759d5

Version
Default

Description
3-tire

Network Info
For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.

VPC
Choose the VPC that defines the virtual network for your Auto Scaling group.
vpc-051261cf95c06c79 (3-tire)

Create a VPC

Availability Zones and subnets
Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.
Select Availability Zones and subnets

ap-south-1a | subnet-0baa9682fc16f9ara (public-1a)
10.0.1.0/24

ap-south-1b | subnet-07493a5b282790193 (public-1b)
10.0.2.0/24

Create a subnet

Availability Zone distribution - new
Auto Scaling automatically balances instances across Availability Zones. If launch failures occur in a zone, select a strategy.

Balanced best effort
If one or more Availability Zones are unhealthy, Auto Scaling will attempt to launch in another healthy Availability Zone.

Balanced only
If launches fail in one Availability Zone, Auto Scaling will continue to attempt to launch in the unhealthy Availability Zone to preserve balanced distribution.

Cancel **Skip to review** **Previous** **Next**

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Create Auto Scaling group | EC2 | ap-south-1

ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#CreateAutoScalingGroup:

Step 1: Choose launch template
Step 2: Choose instance launch options
Step 3 - optional:
Integrate with other services (optional)
Step 4 - optional:
Step 5 - optional:
Step 6 - optional:
Step 7: Review

Integrate with other services - optional Info
Use a load balancer to distribute network traffic across multiple servers. Enable service-to-service communications with VPC Lattice. Shift resources away from impaired Availability Zones with zonal shift. You can also customize health check replacements and monitoring.

Load balancing Info
Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

No load balancer
Traffic to your Auto Scaling group will not be fronted by a load balancer.

Attach to an existing load balancer
Choose from your existing load balancers.

Attach to a new load balancer
Quickly create a basic load balancer to attach to your Auto Scaling group.

Attach to an existing load balancer
Select the load balancers that you want to attach to your Auto Scaling group.

Choose from your load balancer target groups
This option allows you to attach Application, Network, or Gateway Load Balancers.

Choose from Classic Load Balancers

Select target groups
private | HTTP Application Load Balancer: private-ib public | HTTP Application Load Balancer: public-ib

VPC Lattice integration options Info
To improve networking capabilities and scalability, integrate your Auto Scaling group with VPC Lattice. VPC Lattice facilitates communications between AWS services and helps you connect and manage your applications across compute services in AWS.

No VPC Lattice service
VPC Lattice will not manage your Auto Scaling group's network access and connectivity with other services.

Attach to VPC Lattice service
Incoming requests associated with specified VPC Lattice target groups will be routed to your Auto Scaling group.

Create new VPC Lattice service

Application Recovery Controller (ARC) zonal shift - new Info

public-asg Capacity overview

[Edit](#)

Desired capacity 2	Scaling limits (Min - Max) 2 - 25	Desired capacity type Units (number of instances)	Status <input checked="" type="radio"/> Updating capacity
-----------------------	--------------------------------------	--	--

Date created
Fri Jun 13 2025 15:25:12 GMT+0530 (India Standard Time)

Details **Integrations - new** **Automatic scaling** **Instance management** **Instance refresh** **Activity** **Monitoring**

Launch template

Launch template lt-0c59a425d4025759ds template	AMI ID ami-0b09627181c8d5778	Instance type t2.micro	Owner arnawsiam:039950891473:user/Lokesh-B
Version Default	Security groups sg-0415ed7753d948cf1	Storage (volumes)	Create time Fri Jun 13 2025 15:14:43 GMT+0530 (India Standard Time)
Description 5-loc		Key pair name 5-loc	Request Spot Instances No

[View details in the launch template console](#)

Network

Availability Zones ap-south-1b, ap-south-1a	Subnet ID subnet-07493a5b282790193, subnet-0baa9652fc16f9aea	Availability Zone distribution Balanced best effort
--	---	--

Instance type requirements

Your Auto Scaling group adheres to the launch template for purchase option and instance type.

public-asg

ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#AutoScalingGroupDetails?id=public-asg&view=det

Auto Scaling group details | EC2 | ap-south-1

ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#AutoScalingGroupDetails?id=public-asg&view=det

CloudShell Feedback

35°C Cloudy

Search

CloudShell Feedback

CloudShell Feedback

private-asg

private-asg Capacity overview

Desired capacity	Scaling limits (Min - Max)	Desired capacity type	Status
2	2 - 25	Units (number of instances)	-

Date created: Fri Jun 13 2025 15:27:46 GMT+0530 (India Standard Time)

Launch template

Launch template	AMI ID	Instance type	Owner
il-0c59425d4025759d5 template	ami-0b09627181c8d5778	t2.micro	arn:aws:iam::039950891473:user/Lokesh-B
Version	Security groups	Security group IDs	Create time
Default	-	sg-0435e67753d948cf1	Fri Jun 13 2025 15:14:43 GMT+0530 (India Standard Time)
Description	Storage (volumes)	Key pair name	Request Spot Instances
3-tre	-	3-tre	No

Network

Availability Zones	Subnet ID	Availability Zone distribution
ap-south-1b, ap-south-1a	subnet-06db9602e0f8fbfa8, subnet-0019b4ee0f9f74bf	Balanced best effort

Instance type requirements

Your Auto Scaling group adheres to the launch template for purchase option and instance type.

Instances (8) Info

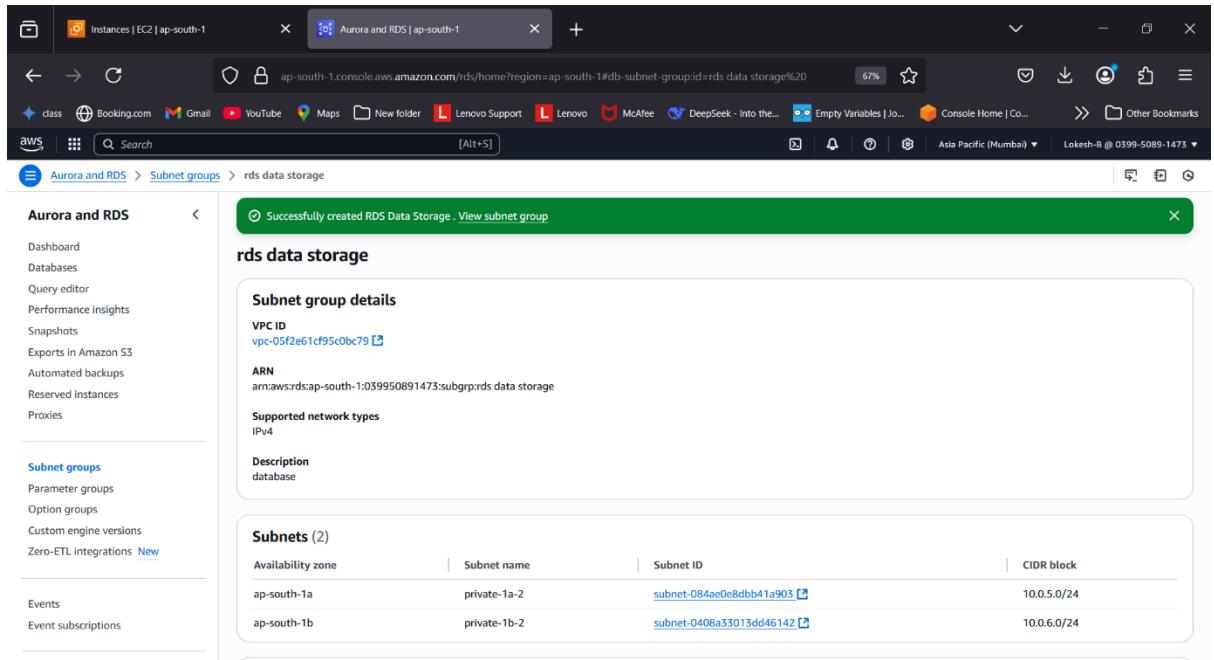
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
private-asg-1	i-095e387b9a49fe316	Running	t2.micro	Initializing	View alarms +	ap-south-1b	-
public-1b	i-080f0f690186d58ab	Running	t2.micro	2/2 checks passec	View alarms +	ap-south-1b	ec2-13-201-168-254.ap...
public-asg-1	i-0646a7899009eb5a0	Running	t2.micro	2/2 checks passec	View alarms +	ap-south-1b	-
private-1b-1	i-03cc807b781556d76	Running	t2.micro	2/2 checks passec	View alarms +	ap-south-1b	ec2-15-206-179-6.ap.s...
private-1a-1	i-0e81de89acf9317	Running	t2.micro	2/2 checks passec	View alarms +	ap-south-1a	ec2-43-205-206-11.ap...
private-asg-2	i-0db2cf24c5a0e65e3	Running	t2.micro	Initializing	View alarms +	ap-south-1a	-
public-1a	i-0e4e8862230810c9a	Running	t2.micro	2/2 checks passec	View alarms +	ap-south-1a	ec2-13-126-227-192.ap...
public-asg-2	i-01814256821affca3	Running	t2.micro	2/2 checks passec	View alarms +	ap-south-1a	-

Select an instance

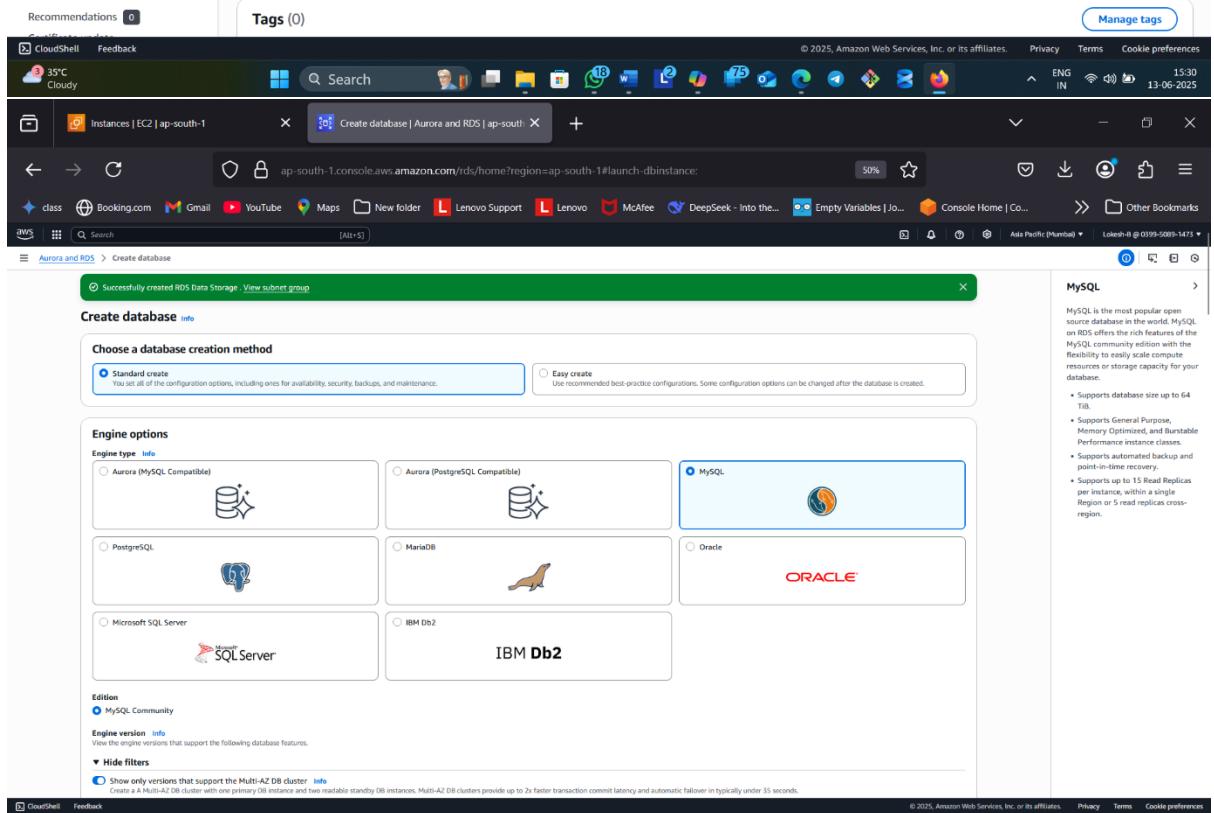
Step-12:

- i) Create rds
- ii) Create subnets groups where we can launch databases
- iii) Select private-1a-2, private-1b-2
- iv) Now come to database

- v) Create database and select mysql engine type
- vi) Select 2 instance type of rds
- vii) Edit name in user and self managed password
- viii) Select public access as yes
- ix) Create rds



The screenshot shows the AWS RDS Subnet group creation page. A success message at the top says "Successfully created RDS Data Storage . View subnet group". The "rds data storage" section displays subnet group details, including VPC ID (vpc-05f2e61cf95c0bc79), ARN (arn:aws:rds:ap-south-1:039950891473:subgrp:rds data storage), supported network types (IPv4), and a description (database). Below this is a table titled "Subnets (2)" showing two subnets: private-1a-2 and private-1b-2, each associated with an availability zone (ap-south-1a and ap-south-1b respectively), a subnet ID, and a CIDR block (10.0.5.0/24 and 10.0.6.0/24).



The screenshot shows the AWS RDS Create database page. A success message at the top says "Successfully created RDS Data Storage . View subnet group". The "Create database" section starts with a "Choose a database creation method" step, offering "Standard create" (selected) and "Easy create". The "Engine options" section shows various database engines: Aurora (MySQL Compatible), Aurora (PostgreSQL Compatible), MySQL (selected), PostgreSQL, MariaDB, Oracle, Microsoft SQL Server, and IBM Db2. On the right, a sidebar for MySQL lists its features: supports database size up to 64 TiB, supports general purpose, memory optimized, and burstable performance modes, supports automated backup and point-in-time recovery, and supports up to 15 read replicas per instance, within a single Region or 5 read replicas cross-region.

Instances | EC2 | ap-south-1 **Create database | Aurora and RDS | ap-south-1**

ap-south-1.console.aws.amazon.com/rds/home?region=ap-south-1#launch-dbinstance: 50%

class Booking.com Gmail YouTube Maps New folder Lenovo McAfee DeepSeek - Into the... Empty Variables | Jo... Console Home | Co... Other Bookmarks Asia Pacific (Mumbai) Lokesh-B @ 0399-5089-1473

Aurora and RDS > Create 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.

database-1 The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 63 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

Credentials Settings

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

admin 1 to 16 alphanumeric characters. The first character must be a letter.

Credentials manager You can use AWS Secrets Manager or manage your master user credentials.

Managed in AWS Secrets Manager - most secure RDS generates a password for you and manages it throughout its lifecycle using AWS Secrets Manager.

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

Master password Info **Very strong** Minimum constraints: At least 8 printable ASCII characters. Can't contain any of the following symbols: / \ * @

Confirm master password Info

MySQL MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance Instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

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 instance.

3-tire (ip-05f261cf950c0c79) 6 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 instance can use in the VPC that you selected.

rds-data-storage 2 Subnets, 2 Availability Zones

Public access Info

- Yes** RDS assigns a public IP address to the database. Amazon EC2 instances and other resources outside of the VPC can connect to your database. Resources inside the VPC can also connect to the database. Choose one or more VPC security groups that specify which resources can connect to the database.
- No** RDS doesn't assign a public IP address to the database. Only Amazon EC2 instances and other resources inside the VPC can connect to your database. Choose one or more VPC security groups that specify which resources can connect to the 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

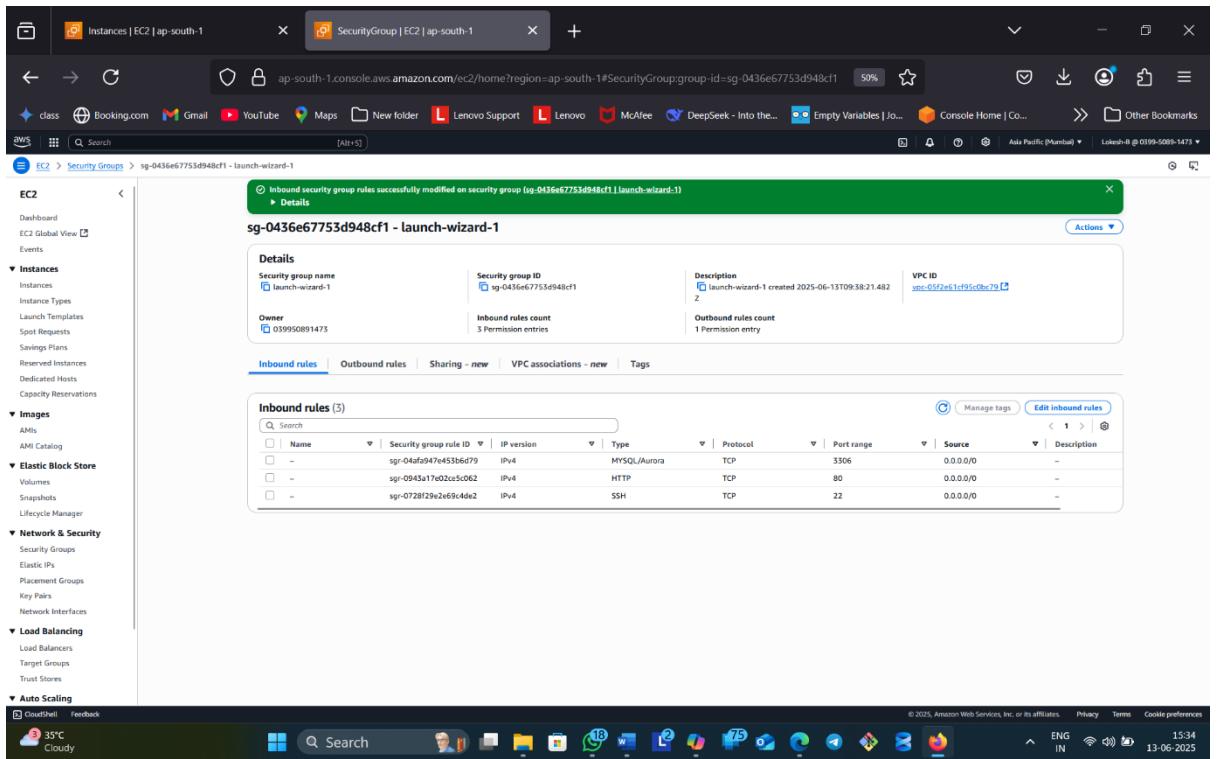
Existing VPC security groups Choose one or more options **Choose existing** Choose existing VPC security groups **Create new** Create new VPC security group

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.

CloseShell Feedback 35°C Cloudy Search ENG IN 15:32 13-06-2025



Step-13:

- After creating RDS come to ec2 console and select private-1b-1 (or) private-1a-1 instance
 - Connect it and copy ssh
 - Go to git bash paste ssh and connect
- Command for my sql launch in server
- Sudo -i
 - sudo dnf install https://dev.mysql.com/get/mysql80-community-release-el9-1.noarch.rpm -y
 - sudo rpm --import <https://repo.mysql.com/RPM-GPG-KEY-mysql-2022>
 - sudo dnf install mysql-community-client --nogpgcheck -y
 - sudo dnf install mysql-community-client -y
 - mysql –version

- g) mysql -h database-1.cham60wge5sx.ap-south-1.rds.amazonaws.com -u admin -p
- h) enter password
- i) show databases;
- j) create database loki;
- k) use loki;
- l) CREATE TABLE Users (
- ```
id INT AUTO_INCREMENT PRIMARY KEY,
name VARCHAR(100),
email VARCHAR(100));
```
- m) INSERT INTO Users (name, email) VALUES ('loki', 'loki@gmail.com');
- n) INSERT INTO Users (name, email) VALUES ('lokesh b', 'lokeshbheem@gmail.com');
- o) SELECT \* FROM Users;

The screenshot shows the AWS Management Console interface for an EC2 instance. The main content area displays the 'Instance summary' for instance `i-0e81dee89cacf9317`. Key details include:

- Public IPv4 address:** 43.205.206.11
- Private IP DNS name (IPv4 only):** ip-10-0-3-55.ap-south-1.compute.internal
- VPC ID:** vpc-05f2e61c995c0bc79
- Subnet ID:** subnet-0019b4eeff09074bf
- Instance ARN:** arnaws:ec2:ap-south-1:039950891473:instance/i-0e81dee89cacf9317

The left sidebar contains navigation links for EC2 services like Instances, Images, and Auto Scaling.

```
ec2-user@ip-10-0-3-!: ~ + - ×
C:\Users\lokes>cd desktop
C:\Users\lokes\Desktop>ssh -i "3-tire.pem" ec2-user@ec2-43-205-206-11.ap-south-1.compute.amazonaws.com
The authenticity of host 'ec2-43-205-206-11.ap-south-1.compute.amazonaws.com (43.205.206.11)' can't be established.
ED25519 key fingerprint is SHA256:xmYlpHlTEaVBAAjFm+P1cU1NDxkJGan2nlaHK53Pc.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-43-205-206-11.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~' '-->
~~ /
~~ .- /-
~~ /_ /-
~/'
[ec2-user@ip-10-0-3-55 ~]$ |
```

```
root@ip-10-0-3-55:~: ~ + - ×
C:\Users\lokes>cd desktop
C:\Users\lokes\Desktop>ssh -i "3-tire.pem" ec2-user@ec2-43-205-206-11.ap-south-1.compute.amazonaws.com
The authenticity of host 'ec2-43-205-206-11.ap-south-1.compute.amazonaws.com (43.205.206.11)' can't be established.
ED25519 key fingerprint is SHA256:xmYlpHlTEaVBAAjFm+P1cU1NDxkJGan2nlaHK53Pc.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-43-205-206-11.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~' '-->
~~ /
~~ .- /-
~~ /_ /-
~/'
[ec2-user@ip-10-0-3-55 ~]$ sudo -i
[root@ip-10-0-3-55 ~]# sudo dnf install https://dev.mysql.com/get/mysql80-community-release-el9-1.noarch.rpm -y |
```

```

root@ip-10-0-3-55:~ + ~
Amazon Linux 2023 Kernel Livepatch repository
mysql80-community-release-el9-1.noarch.rpm
Dependencies resolved.
=====
 Package Architecture Version Repository Size
=====
Installing:
 mysql80-community-release noarch el9-1 @commandline 10 k
Transaction Summary
=====
Install 1 Package

Total size: 10 k
Installed size: 5.7 k
Downloading Packages:
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
 Preparing :
 Installing : mysql80-community-release-el9-1.noarch
 Verifying : mysql80-community-release-el9-1.noarch
 1/1
 1/1
 1/1
Installed:
 mysql80-community-release-el9-1.noarch

Complete!
[root@ip-10-0-3-55 ~]# |
```

```

root@ip-10-0-3-55:~ + ~
Total download size: 6.7 M
Installed size: 96 M
Downloading Packages:
(1/4): mysql-community-common-8.0.42-1.el9.x86_64.rpm 24 MB/s | 555 kB 00:00
(2/4): mysql-community-client-8.0.42-1.el9.x86_64.rpm 73 MB/s | 3.4 MB 00:00
(3/4): mysql-community-libs-8.0.42-1.el9.x86_64.rpm 40 MB/s | 1.5 MB 00:00
(4/4): mysql-community-client-plugins-8.0.42-1.el9.x86_64.rpm 20 MB/s | 1.4 MB 00:00
Total 95 MB/s | 6.7 MB 00:00
MySQL 8.0 Community Server
GPG key at file:///etc/pki/rpm-gpg/RPM-GPG-KEY-mysql-2022 (0x3A79BD29) is already installed
The GPG keys listed for the "MySQL 8.0 Community Server" repository are already installed but they are not correct for this package.
Check that the correct key URLs are configured for this repository.. Failing package is: mysql-community-client-8.0.42-1.el9.x86_64
GPG Keys are configured as: file:///etc/pki/rpm-gpg/RPM-GPG-KEY-mysql-2022
Public key for mysql-community-client-plugins-8.0.42-1.el9.x86_64.rpm is not installed. Failing package is: mysql-community-client-plugins-8.0.42-1.el9.x86_64
GPG Keys are configured as: file:///etc/pki/rpm-gpg/RPM-GPG-KEY-mysql-2022
Public key for mysql-community-common-8.0.42-1.el9.x86_64.rpm is not installed. Failing package is: mysql-community-common-8.0.42-1.el9.x86_64
GPG Keys are configured as: file:///etc/pki/rpm-gpg/RPM-GPG-KEY-mysql-2022
Public key for mysql-community-libs-8.0.42-1.el9.x86_64.rpm is not installed. Failing package is: mysql-community-libs-8.0.42-1.el9.x86_64
GPG Keys are configured as: file:///etc/pki/rpm-gpg/RPM-GPG-KEY-mysql-2022
The downloaded packages were saved in cache until the next successful transaction.
You can remove cached packages by executing 'dnf clean packages'.
Error: GPG check FAILED
-bash: mysql: command not found
[root@ip-10-0-3-55 ~]# sudo dnf install mysql-community-client --nogpgcheck -y |
```

```

root@ip-10-0-3-55:~ + ~
Downloading Packages:
[SKIPPED] mysql-community-client-8.0.42-1.el9.x86_64.rpm: Already downloaded
[SKIPPED] mysql-community-client-plugins-8.0.42-1.el9.x86_64.rpm: Already downloaded
[SKIPPED] mysql-community-common-8.0.42-1.el9.x86_64.rpm: Already downloaded
[SKIPPED] mysql-community-libs-8.0.42-1.el9.x86_64.rpm: Already downloaded
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
 Preparing :
 Installing : mysql-community-client-plugins-8.0.42-1.el9.x86_64
 Installing : mysql-community-common-8.0.42-1.el9.x86_64
 Installing : mysql-community-libs-8.0.42-1.el9.x86_64
 Running scriptlet: mysql-community-libs-8.0.42-1.el9.x86_64
 Installing : mysql-community-client-8.0.42-1.el9.x86_64
 Running scriptlet: mysql-community-client-8.0.42-1.el9.x86_64
 Verifying : mysql-community-client-8.0.42-1.el9.x86_64
 Verifying : mysql-community-client-plugins-8.0.42-1.el9.x86_64
 Verifying : mysql-community-common-8.0.42-1.el9.x86_64
 Verifying : mysql-community-libs-8.0.42-1.el9.x86_64
 1/1
 1/4
 2/4
 3/4
 3/4
 4/4
 4/4
 1/4
 2/4
 3/4
 4/4
Installed:
 mysql-community-client-8.0.42-1.el9.x86_64 mysql-community-client-plugins-8.0.42-1.el9.x86_64
 mysql-community-common-8.0.42-1.el9.x86_64 mysql-community-libs-8.0.42-1.el9.x86_64

Complete!
[root@ip-10-0-3-55 ~]# mysql --version
mysql Ver 8.0.42 for Linux on x86_64 (MySQL Community Server - GPL)
[root@ip-10-0-3-55 ~]# |
```

```

root@ip-10-0-3-55:~ + ~
Downloading Packages:
[SKIPPED] mysql-community-client-8.0.42-1.el9.x86_64.rpm: Already downloaded
[SKIPPED] mysql-community-plugins-8.0.42-1.el9.x86_64.rpm: Already downloaded
[SKIPPED] mysql-community-common-8.0.42-1.el9.x86_64.rpm: Already downloaded
[SKIPPED] mysql-community-libs-8.0.42-1.el9.x86_64.rpm: Already downloaded
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing : 1/1
Installing : mysql-community-client-plugins-8.0.42-1.el9.x86_64 1/4
Installing : mysql-community-common-8.0.42-1.el9.x86_64 2/4
Installing : mysql-community-libs-8.0.42-1.el9.x86_64 3/4
Running scriptlet: mysql-community-libs-8.0.42-1.el9.x86_64 3/4
Installing : mysql-community-client-8.0.42-1.el9.x86_64 4/4
Running scriptlet: mysql-community-client-8.0.42-1.el9.x86_64 4/4
Verifying : mysql-community-client-8.0.42-1.el9.x86_64 4/4
Verifying : mysql-community-client-plugins-8.0.42-1.el9.x86_64 1/4
Verifying : mysql-community-common-8.0.42-1.el9.x86_64 2/4
Verifying : mysql-community-libs-8.0.42-1.el9.x86_64 3/4
Complete! 4/4
[root@ip-10-0-3-55 ~]# mysql --version
mysql Ver 8.0.42 for Linux on x86_64 (MySQL Community Server - GPL)
[root@ip-10-0-3-55 ~]# mysql -h database-1.cham60wge5sx.ap-south-1.rds.amazonaws.com -u admin -p
root@ip-10-0-3-55:~ + ~
Running scriptlet: mysql-community-libs-8.0.42-1.el9.x86_64 3/4
Installing : mysql-community-client-8.0.42-1.el9.x86_64 4/4
Running scriptlet: mysql-community-client-8.0.42-1.el9.x86_64 4/4
Verifying : mysql-community-client-8.0.42-1.el9.x86_64 1/4
Verifying : mysql-community-client-plugins-8.0.42-1.el9.x86_64 2/4
Verifying : mysql-community-common-8.0.42-1.el9.x86_64 3/4
Verifying : mysql-community-libs-8.0.42-1.el9.x86_64 4/4
Complete! 4/4
[root@ip-10-0-3-55 ~]# mysql --version
mysql Ver 8.0.42 for Linux on x86_64 (MySQL Community Server - GPL)
[root@ip-10-0-3-55 ~]# mysql -h database-1.cham60wge5sx.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 28
Server version: 8.0.41 Source distribution

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

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> |

root@ip-10-0-3-55:~ + ~
Complete!
[root@ip-10-0-3-55 ~]# mysql --version
mysql Ver 8.0.42 for Linux on x86_64 (MySQL Community Server - GPL)
[root@ip-10-0-3-55 ~]# mysql -h database-1.cham60wge5sx.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 28
Server version: 8.0.41 Source distribution

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

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> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.00 sec)

mysql> |

```

```

root@ip-10-0-3-55:~ +
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.00 sec)

mysql> create database loki;
Query OK, 1 row affected (0.00 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| loki |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.00 sec)

mysql> |

root@ip-10-0-3-55:~ +
mysql> use loki;
Database changed
mysql> CREATE TABLE Users (
 -> id INT AUTO_INCREMENT PRIMARY KEY,
 -> name VARCHAR(100),
 -> email VARCHAR(100)
 ->);
Query OK, 0 rows affected (0.02 sec)

mysql> INSERT INTO Users (name, email) VALUES ('loki', 'loki@gmail.com');
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO Users (name, email) VALUES ('Lokesh', 'Lokeshbheemagani@gmail.com');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO Users (name, email) VALUES ('Lokesh b', 'Lokeshbheem@gmail.com');
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM Users;
+----+----+----+
| id | name | email |
+----+----+----+
1	loki	loki@gmail.com
2	Lokesh	Lokeshbheemagani@gmail.com
3	Lokesh b	Lokeshbheem@gmail.com
+----+----+----+
3 rows in set (0.00 sec)

mysql> |

```

## Step-14:

- I) Now select one instance in web
- II) Select one instance (public-1a,public-1b)
- III) Connect it and copy ssh
- IV) Paste ssh in git bash

Command for my sql launch in server

a) Sudo -i

- b)sudo dnf install https://dev.mysql.com/get/mysql80-community-release-el9-1.noarch.rpm -y
- c)sudo rpm --import <https://repo.mysql.com/RPM-GPG-KEY-mysql-2022>
- d)sudo dnf install mysql-community-client --nogpgcheck -y
- e)sudo dnf install mysql-community-client -y
- f)mysql –version
- g)mysql -h database-1.cham60wge5sx.ap-south-1.rds.amazonaws.com -u admin -p
- h)enter password
- i)show databases;
- j) use loki;
- k)show Tables;
- l)SELECT\*FROM Users;

Instance details | EC2 | ap-south-1 Database Details | Aurora and RDS | ap-south-1

ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#instanceDetailsinstanceId=i-080f0f690186d58ab

**Instance summary for i-080f0f690186d58ab (public-1b)**

Updated less than a minute ago

**Instance ID**: i-080f0f690186d58ab

**Public IPv4 address**: 13.201.168.254 | open address

**Instance state**: Running

**Private IP DNS name (IPv4 only)**: ip-10-0-2-232.ap-south-1.compute.internal

**Instance type**: t2.micro

**VPC ID**: vpc-05f2e61c9550bc79 (Single) | open

**Subnet ID**: subnet-07493a5b282790195 (public-1b) | open

**Instance ARN**: arn:aws:ec2:ap-south-1:039950891473:instance/i-080f0f690186d58ab

**Elastic IP addresses**: 10.0.2.232

**Private DNS**: ec2-13-201-168-254.ap-south-1.compute.amazonaws.com | open address

**AWS Compute Optimizer finding**: Opt-in to AWS Compute Optimizer for recommendations. | Learn more

**Auto Scaling Group name**: -

**Managed**: false

**Details** | Status and alarms | Monitoring | Security | Networking | Storage | Tags

**AMI ID**: ami-0b09627181bd5778

**Monitoring**: disabled

**AMI name**: al2023-ami-2023.7.20250609.0-kernel-6.1-x86\_64

**Allowed Image**: -

**Launch time**: Fri Jun 13 2023 15:10:52 GMT+0530 (India Standard Time) (38 minutes)

**AMI location**: amazon/al2023-ami-2023.7.20250609.0-kernel-6.1-x86\_64

**Instance reboot migration**: Default (On)

**Instance auto-recovery**: Default

**AMI Launch index**: 0

**Credit specification**: -

**Platform details**: Linux/UNIX

**Termination protection**: Disabled

**Kernel ID**: 5-tire

**Lifecycle**: normal

**Key pair assigned at launch**: 5-tire

**CloudShell Feedback**

root@ip-10-0-2-232:~

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> show databases;
+-----+
| Database |
+-----+
| information_schema |
| loki |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.00 sec)

mysql> use loki;
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> SELECT * FROM.table_name;
[1]+ Stopped mysql -h database-1.cham60wge5sx.ap-south-1.rds.amazonaws.com -u admin -p
[root@ip-10-0-2-232 ~]# SELECT * FROM loki.Users;
-bash: SELECT: command not found
[root@ip-10-0-2-232 ~]# SELECT * FROM lokidb.Users;
-bash: SELECT: command not found
[root@ip-10-0-2-232 ~]# |
```

```
root@ip-10-0-2-232: ~ + ~
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 37
Server version: 8.0.41 Source distribution

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

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> show databases;
+-----+
| Database |
+-----+
| information_schema |
| loki |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.00 sec)

mysql> use loki;
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_loki |
+-----+
| Users |
+-----+
1 row in set (0.00 sec)

mysql> SELECT * FROM Users;
+----+-----+-----+
| id | name | email |
+----+-----+-----+
1	loki	loki@gmail.com
2	lokesh	lokeshbheemagani@gmail.com
3	lokesh b	lokeshbheem@gmail.com
+----+-----+-----+
3 rows in set (0.00 sec)

35°C
Cloudy
Search
ENG IN 13-06-2025 15:58
```

Lokesh Bheemagani

lokeshbheemagani@gmail.com