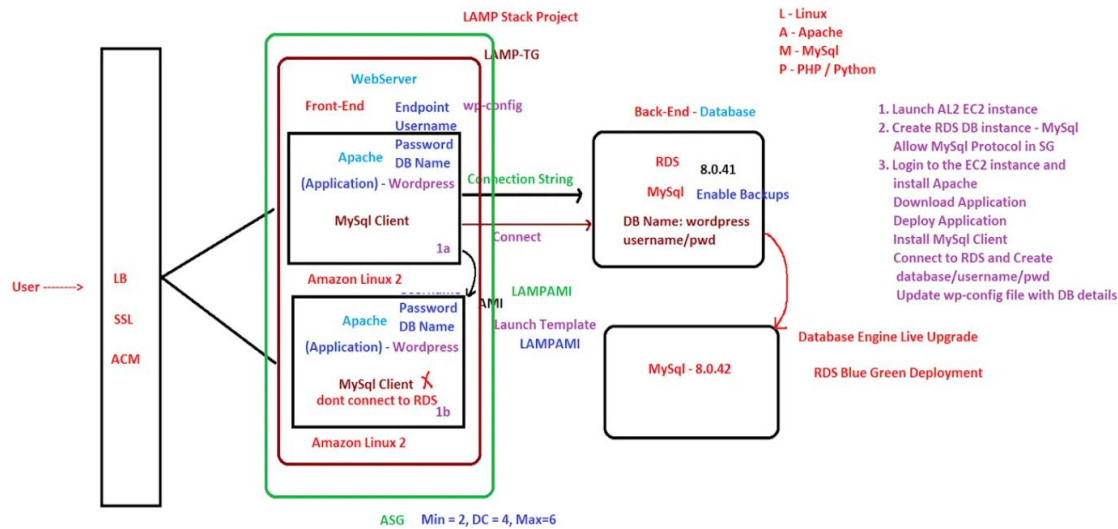
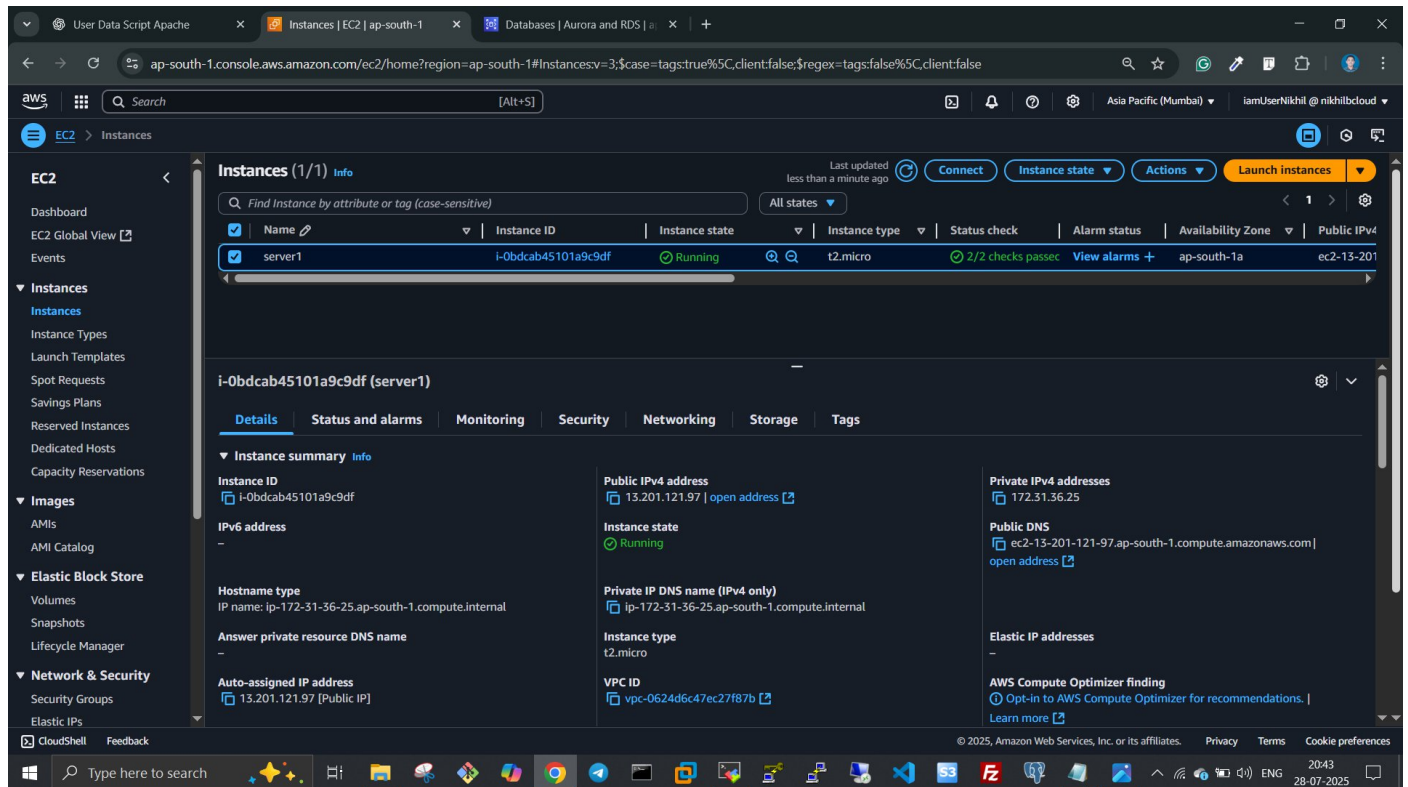


Scalable WordPress Hosting on AWS with LAMP Stack and Blue/Green RDS Upgrade

Architecture Diagram of Project -



1. Launching ec2 instance with instance type Amazon Linux 3



2. Creating Database instance using RDB with MYSQL engine 8.0.41 ensure enable automatic backup

The screenshot displays the AWS Management Console for an Aurora RDS instance. The left sidebar shows the navigation menu with 'Aurora and RDS' selected. The main content area shows the 'db-inst-primary' instance details. The 'Summary' section indicates the instance is 'Available' with a CPU usage of 4.55%. The 'Connectivity & security' tab is active, showing the endpoint 'db-inst-primary.cp688y8cu9gw.ap-south-1.rds.amazonaws.com' on port 3306. The 'Networking' section shows the instance is in the 'ap-south-1a' Availability Zone, using VPC 'vpc-0624d6c47ec27f87b' and Subnet 'subnet-0260a6122c357697b'. The 'Security' section shows the VPC security group 'default (sg-018cd02756047ec6c)' is active and publicly accessible. The 'DB instance certificate expiration date' is May 20, 2061, 00:10 (UTC+05:30).

3. Allow protocols in SG, MySQL or anywhere

The screenshot displays the 'Edit inbound rules' page for a security group. The page shows a list of inbound rules with columns for 'Security group rule ID', 'Type', 'Protocol', 'Port range', 'Source', and 'Description - optional'. The rules are as follows:

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-03e911294f4b2daf7	SSH	TCP	22	Custom	0.0.0.0/0
sgr-08045d844d384de78	RDP	TCP	3389	Custom	45.250.227.156/32
sgr-0d1165145f59303ae	NFS	TCP	2049	Custom	0.0.0.0/0
sgr-0447aa5438dd50eb6	MYSQL/Aurora	TCP	3306	Custom	0.0.0.0/0
sgr-0ada80843c3ed7a49	HTTP	TCP	80	Custom	0.0.0.0/0
sgr-0c3861304bb643183	All traffic	All	All	Custom	sg-018cd02756047ec6c
sgr-010e2453b52b2fc16	All traffic	All	All	Custom	0.0.0.0/0

The 'Add rule' button is visible at the bottom left of the rules list.

4. Installing MYSQL client as we need to connect to DB instance and creating database, username and pass.

5. Installing Apache webserver and installing WordPress app, extracting it and configuring config file that is setting connection string so that application should connect to database.

6. Copy all WordPress files to Apache directory /var/www/html then restart apache server

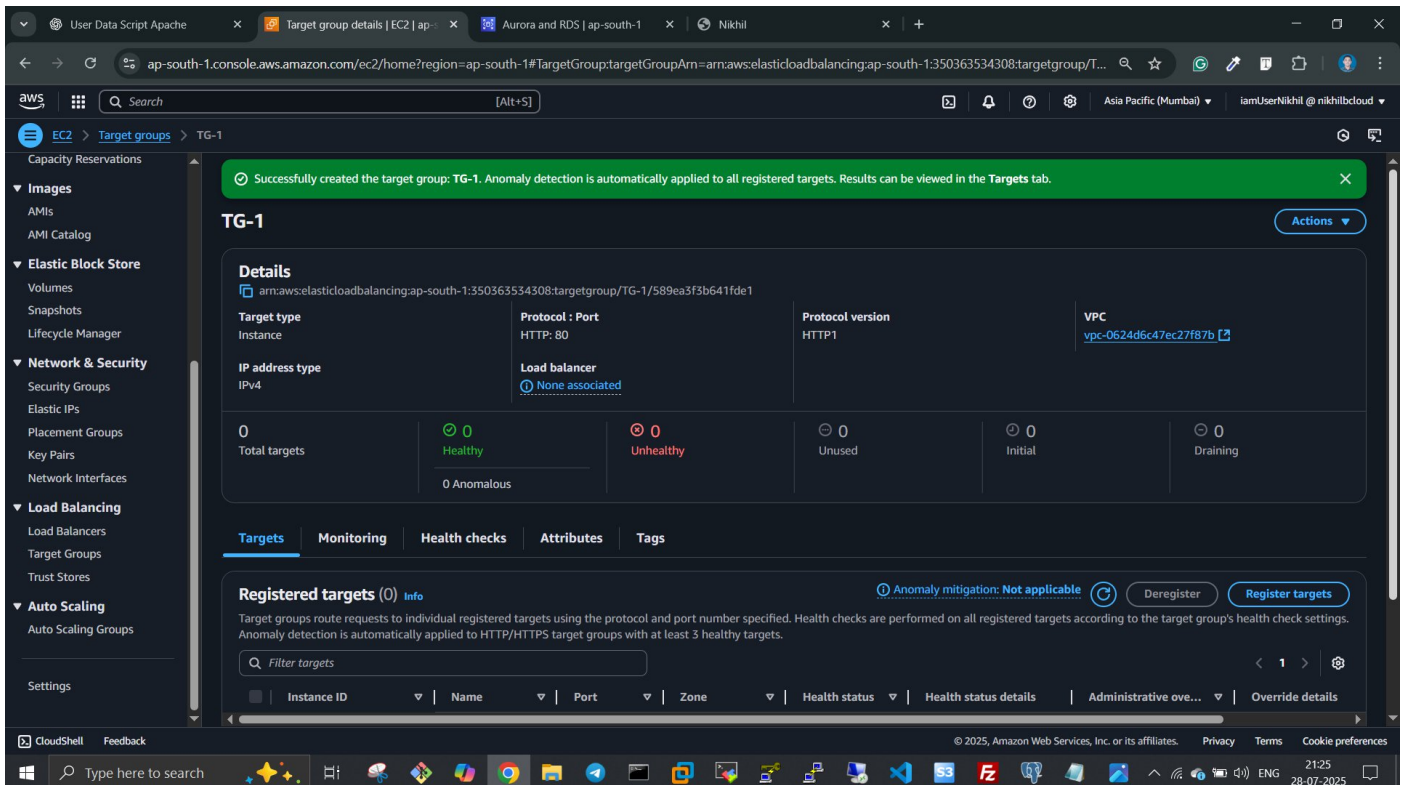
7. Now what if this ec2 goes down so we have to set Auto Scaling

So first create AMI from this ec2 where all configuration are done so that we can use same AMI in launch template

Amazon Machine Images (AMIs) (1/1) Info			
Name	AMI name	AMI ID	Source
server1-image	ami-07fcbccd9e3f6b139	350363534308/server1-image	350363534308

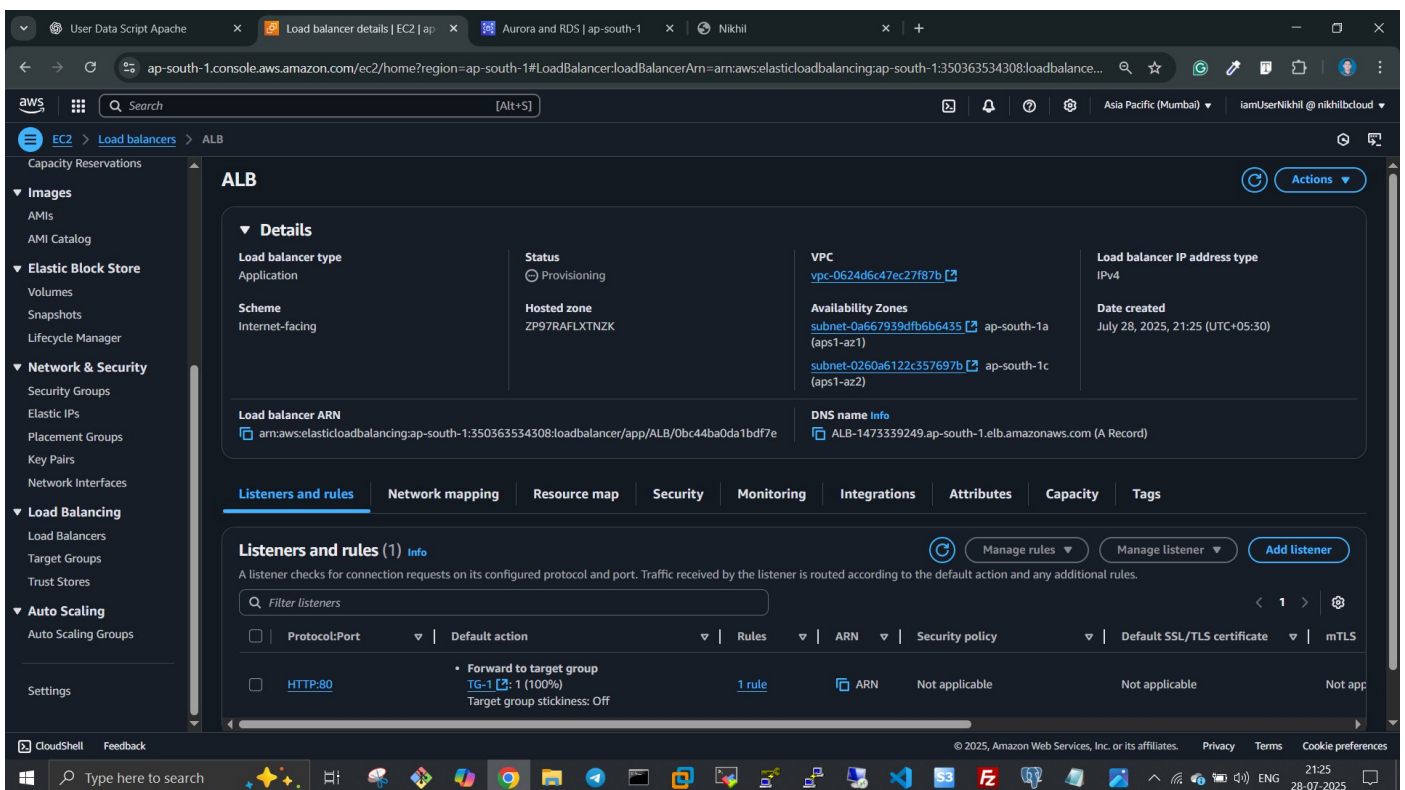
AMI ID: ami-07fcbccd9e3f6b139			
Details	Permissions	Storage	Tags
AMI ID ami-07fcbccd9e3f6b139	Image type machine	Platform details Linux/UNIX	Root device type EBS
AMI name server1-image	Owner account ID 350363534308	Architecture x86_64	Usage operation RunInstances
Root device name /dev/xvda	Status Available	Source 350363534308/server1-image	Virtualization type hvm
Boot mode uefi-preferred	State reason -	Creation date 2025-07-28T15:48:05.000Z	Kernel ID -
Description -	Product codes -	RAM disk ID -	Deprecation time -
Last launched time -	Block devices /dev/xvda=snap-01e649f8fabdb13f18:true:gp3	Deregistration protection Disabled	Allowed image -

8. Creating Empty Target Group giving health check path / only (ie: root path)



The screenshot shows the AWS Management Console interface for a newly created Target Group (TG-1). The console is in the 'ap-south-1' region. A green notification banner at the top states: "Successfully created the target group: TG-1. Anomaly detection is automatically applied to all registered targets. Results can be viewed in the Targets tab." The left sidebar shows the navigation menu with categories like Capacity Reservations, Images, Elastic Block Store, Network & Security, Load Balancing, and Auto Scaling. The main content area displays the details of TG-1, including its ARN, target type (Instance), IP address type (IPv4), protocol (HTTP), and port (80). A summary table shows 0 total targets, 0 healthy, 0 unhealthy, 0 unused, 0 initial, and 0 draining. Below the details, there are tabs for Targets, Monitoring, Health checks, Attributes, and Tags. The 'Targets' tab is active, showing 'Registered targets (0)' and a 'Filter targets' search bar. A table at the bottom lists columns for Instance ID, Name, Port, Zone, Health status, Health status details, Administrative overview, and Override details.

9. Creating Application Load Balancer – distributing traffic to diff ec2 servers



The screenshot shows the AWS Management Console interface for a newly created Application Load Balancer (ALB). The console is in the 'ap-south-1' region. The left sidebar shows the navigation menu. The main content area displays the details of the ALB, including its type (Application), status (Provisioning), scheme (Internet-facing), VPC (vpc-0624d6c47ec27f87b), and availability zones (subnet-0a667939dfb6b6435 and subnet-0260a6122c357697b). The load balancer ARN is 'arn:aws:elasticloadbalancing:ap-south-1:350363534308:loadbalancer/app/ALB/0bc44ba0da1bd7fe'. Below the details, there are tabs for Listeners and rules, Network mapping, Resource map, Security, Monitoring, Integrations, Attributes, Capacity, and Tags. The 'Listeners and rules' tab is active, showing 'Listeners and rules (1)'. A table lists the listener details: Protocol (HTTP), Port (80), Default action (Forward to target group TG-1), Rules (1 rule), ARN, Security policy (Not applicable), and Default SSL/TLS certificate (Not applicable).

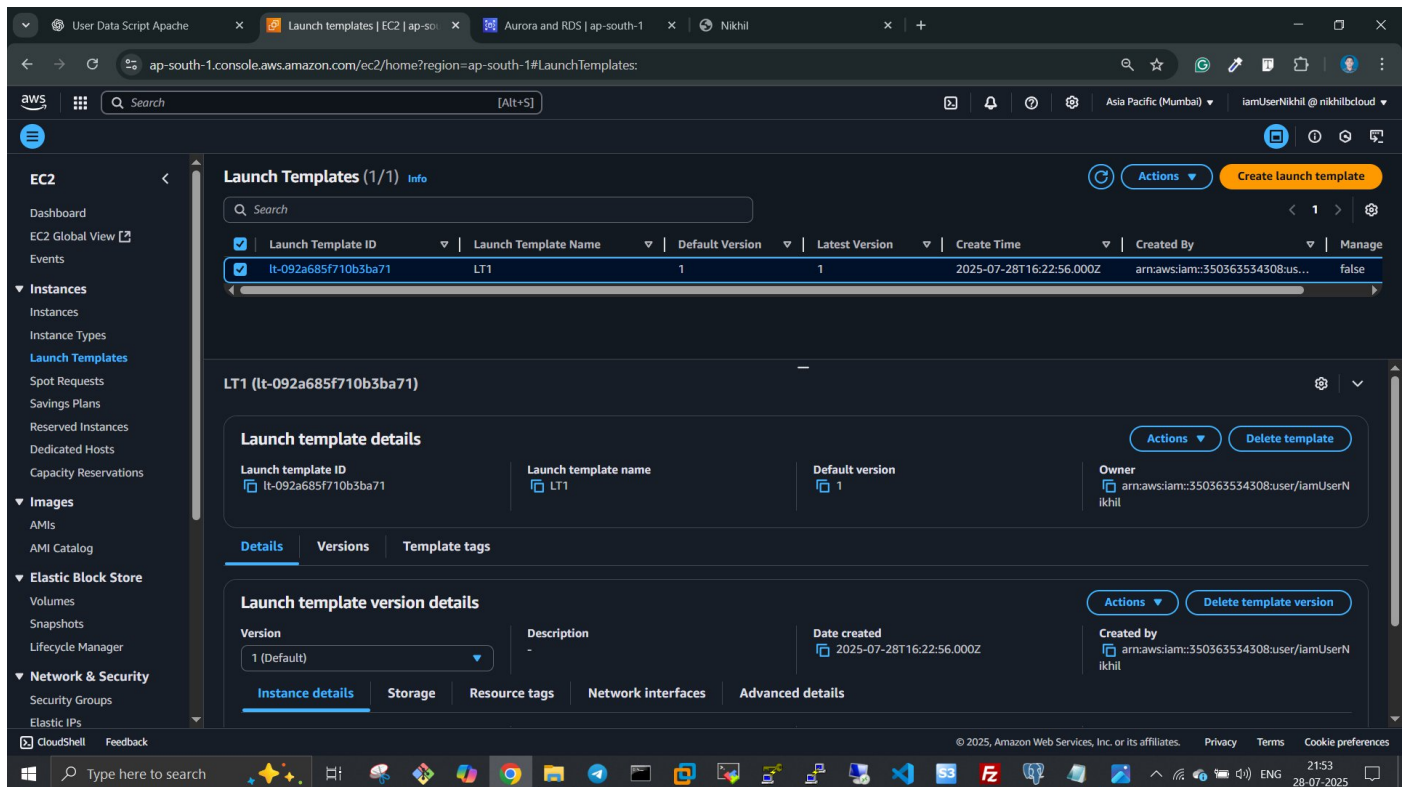
10. For Creating ASG we need to launch first launch template

while launching launch template use same AMI and in user data mention -

systemctl restart httpd

systemctl restart php-fpm

because when ASG launches new ec2 we need to restart apache and php



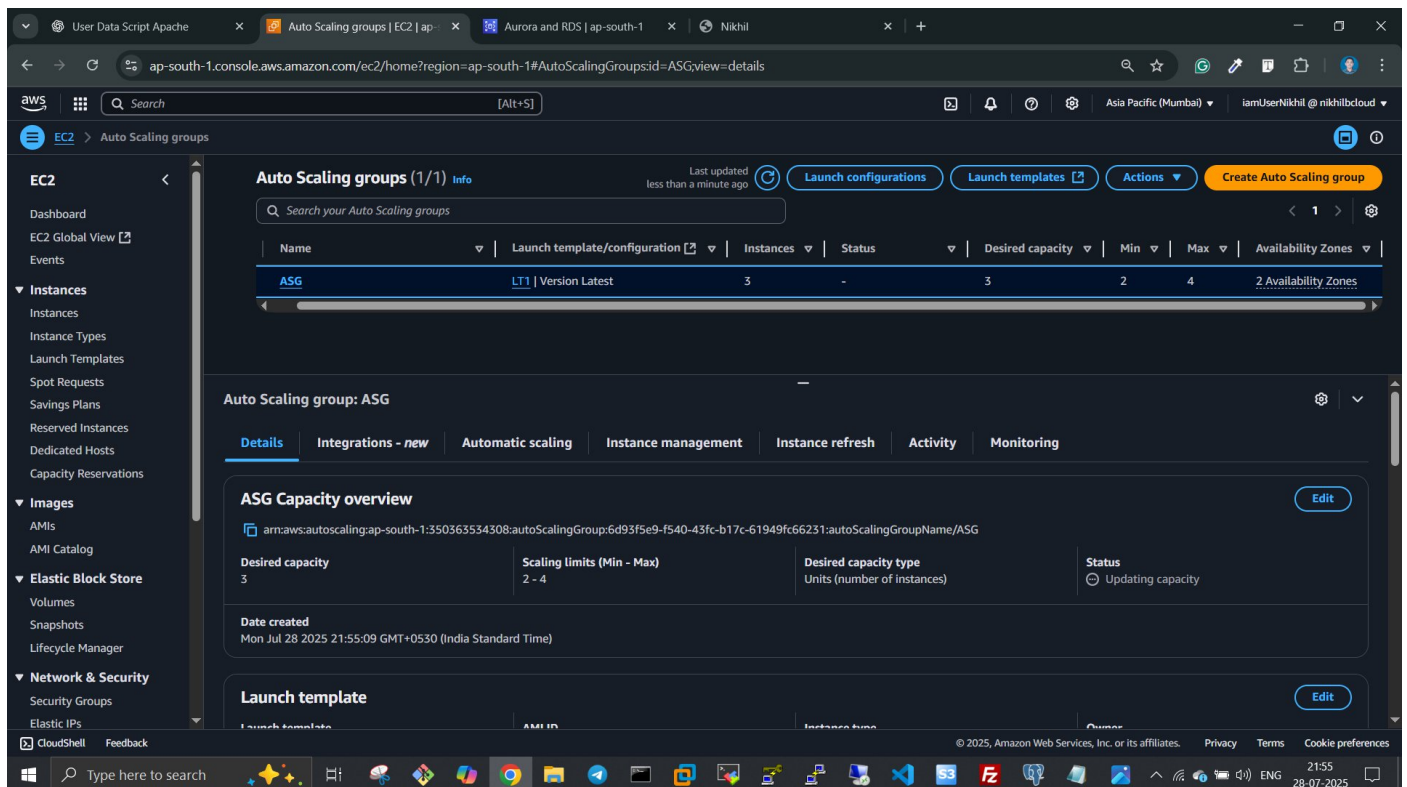
The screenshot shows the AWS Management Console for the 'ap-south-1' region. The 'Launch Templates' page is active, displaying a table with one launch template, LT1. The details section for LT1 shows the following information:

Launch template ID	Launch template name	Default version	Owner
lt-092a685f710b3ba71	LT1	1	arn:aws:iam::350363534308:user/iamUserNikhil

The 'Launch template version details' section shows the following information:

Version	Description	Date created	Created by
1 (Default)	-	2025-07-28T16:22:56.000Z	arn:aws:iam::350363534308:user/iamUserNikhil

Create ASG



The screenshot shows the AWS Management Console for the 'ap-south-1' region. The 'Auto Scaling groups' page is active, displaying a table with one auto scaling group, ASG. The details section for ASG shows the following information:

Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones
ASG	LT1 Version Latest	3	-	3	2	4	2 Availability Zones

The 'Auto Scaling group: ASG' details section shows the following information:

ASG Capacity overview	Launch template
<p>arn:aws:autoscaling:ap-south-1:350363534308:autoScalingGroup:6d93f5e9-f540-43fc-b17c-61949fc66231:autoScalingGroupName/ASG</p> <p>Desired capacity: 3</p> <p>Scaling limits (Min - Max): 2 - 4</p> <p>Desired capacity type: Units (number of instances)</p> <p>Status: Updating capacity</p> <p>Date created: Mon Jul 28 2025 21:55:09 GMT+0530 (India Standard Time)</p>	

11. After this you will see ec2 instances in TG where you can check health check and access wordpress using ALB

The screenshot shows the AWS Management Console interface for Target Groups in the ap-south-1 region. The left sidebar contains navigation links for various AWS services. The main content area displays the 'Target groups (1/1)' page. A table lists the target groups, with 'TG-1' selected. Below the table, the 'Targets' tab is active, showing a list of registered targets. The targets are EC2 instances with IDs like i-0e6823432bf55befb, i-014530a16e8d01680, and i-005f5d8412c0b780e. Each target has a health status of 'Healthy' or 'Unused'.

Name	ARN	Port	Protocol	Target type	Load balancer	VPC ID
TG-1	arn:aws:elasticloadbalancing...	80	HTTP	Instance	ALB	vpc-0624d6c47ec27f87b

Instance ID	Name	Port	Zone	Health status	Health statu...	Admini...	Overri...	Launch...	Anomaly
i-0e6823432bf55befb		80	ap-south-1a(a...	Healthy	-	No override	No overri...	July 28, 2...	Norm...
i-014530a16e8d01680		80	ap-south-1b(a...	Unused	Target is in a...	No override	No overri...	July 28, 2...	Norm...
i-005f5d8412c0b780e		80	ap-south-1a(a...	Healthy	-	No override	No overri...	July 28, 2...	Norm...

12. Now we need to upgrade MYSQL version from 8.0.41 to 8.0.42 without any downtime so we are using BLUE-GREEN-DEPLOYMENT where endpoint will remain same

The screenshot shows the 'Create blue/green deployment' wizard in the AWS Management Console for RDS. The wizard is divided into several sections: 'Engine configurations', 'Instance configuration', and 'RDS Optimized Writes'. In the 'Engine configurations' section, the 'Blue instance configuration' is set to 'Blue' with engine version 8.0.41. The 'Green instance configuration' is set to 'Green' with engine version MySQL 8.0.42. The 'Blue DB parameter group' is set to 'default.mysql8.0'. In the 'Instance configuration' section, the 'Blue DB instance class' is set to 'db.t3.micro'. The 'RDS Optimized Writes' section has a checkbox for 'Enable Optimized Writes for green database' which is currently unchecked.

Engine configurations

Blue instance configuration **Blue**
Blue engine version 8.0.41
Blue DB parameter group default.mysql8.0

Green instance configuration **Green**
Choose the engine version for green databases. MySQL 8.0.42
Choose the DB parameter group for green databases. default.mysql8.0

Instance configuration

Blue DB instance class **Blue**
db.t3.micro

RDS Optimized Writes

☐ Enable Optimized Writes for green database
Amazon RDS Optimized Writes provide up to 2x improvement in write transaction throughput.

Green DB instance class **Green**
☐ Compute optimized classes (includes c classes)

User Data Script ApacheTarget groups | EC2 | ap-south-1Aurora and RDS | ap-south-1Nikhil

ap-south-1.console.aws.amazon.com/rds/home?region=ap-south-1#create-bgdinstance-id=db-inst-primary;prevRoute=%23databases%3A

Search[Alt+S]

Asia Pacific (Mumbai)iamUserNikhil @ nikhilcloud

Aurora and RDSDatabasesCreate blue/green deployment

review and commr

Connection overview

Blue/green deployment name
bg-deployment-1

Database identifier Blue
db-inst-primary

Replication

Database identifier Green
db-inst-primary-green

Step 1: Configure blue/green deployment

Edit

Blue/green deployment name

Blue/green deployment name
bg-deployment-1

Configuration details

	Blue database Blue	Green database Green
DB instance ID	db-inst-primary	db-inst-primary-green
Engine	MySQL Community	MySQL Community
Engine version	8.0.41	8.0.42 *
DB Parameter group	default.mysql8.0	default.mysql8.0

CloudShellFeedback

© 2025, Amazon Web Services, Inc. or its affiliates. PrivacyTermsCookie preferences

Type here to search

22:00
28-07-2025

User Data Script ApacheTarget groups | EC2 | ap-south-1Databases | Aurora and RDS | ap-south-1Nikhil

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

Search[Alt+S]

Asia Pacific (Mumbai)iamUserNikhil @ nikhilcloud

Aurora and RSDatabases

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

Event subscriptions

Recommendations 0

Certificate update

Successfully created blue/green deployment bg-deployment-1 and green database db-inst-primary-green-42daxu

View details

How was your experience creating a Blue/Green Deployment? Provide feedback

Databases (3)

Group resources

Modify

Actions

Create database

Filter by databases

DB identifier	Status	Role	Engine	Region & AZ	Size	Recommendati
bg-deployment-1	Available	Blue/gree...	-	-	-	-
db-inst-primary	Available	Primary	MySQL Com...	ap-south-1a	db.t3.micro	
db-inst-primary-green-42daxu	Available	Primary	MySQL Com...	ap-south-1a	db.t3.micro	

CloudShellFeedback

© 2025, Amazon Web Services, Inc. or its affiliates. PrivacyTermsCookie preferences

Type here to search

22:19
28-07-2025

Hello world!

July 28, 2025