



Expert Cloud Consulting

Enhance Optimise & Scale

ASCP GPUonCLOUD Pvt Ltd

“Expert Cloud Consulting” -

SOP | Managed Services: CloudWatch Monitoring for RDS

28.July.2025

version 1.0

Contributed by Tejal Kale

Approved by Akshay Shinde

Expert Cloud Consulting

Office #811, Gera Imperium Rise,

Hinjewadi Phase-II Rd, Pune, India – 411057

“Expert Cloud Consulting”

Managed Services: CloudWatch Monitoring for RDS

1.0 Contents

1.0 Contents	1
2.0 General Information:	2
2.1 Document Purpose.....	2
2.2 Document Revisions.....	2
3.0 Document Overview:.....	3
3.1 Document Workflow :	3
4.0 Steps / Create RDS Instance	4
Modules Structure	5
4.4: Apply Terraform Scripts	5
Initialize and apply the Terraform code:	5
4.5: Cloudwatch Alarms	5
4.5.1: Simulate Load to Trigger Alarms.....	5
4.5.2: Alarm Triggers SNS Topic	6
4.5.3: SNS Triggers Lambda Function	6
4.6: Lambda Function Actions	7





2.0 General Information:

2.1 Document Purpose

This document provides an introduction to **Infrastructure as Code (IaC)**, focusing on automation and management of cloud resources using **Terraform** and **CloudFormation**. It includes hands-on assignments to provision a multi-tier architecture on AWS with Terraform and automate S3-Lambda integrations with CloudFormation. The purpose is to equip users with practical skills in IaC for scalable and repeatable cloud infrastructure deployments.

2.2 Document Revisions

Date	Version	Contributor(s)	Approver(s)	Section(s)	Change(s)
27/Dec/2024	1.0	Tejal Kale	Akshay Shinde	All Sections	New Document Created

3.0 Document Overview:

This solution provides complete monitoring and automatic incident response for a MySQL RDS instance using AWS and the osTicket system. The main goal is to detect issues early, alert the right people, and automatically create support tickets—without needing any manual work.

First, we create a MySQL database on Amazon RDS. Then, using Terraform, we set up CloudWatch alarms for important metrics like:

- CPUUtilization – to catch high CPU usage
- FreeableMemory – to watch for low memory
- FreeStorageSpace – to track low disk space
- DatabaseConnections – to find connection problems
- RDS Events – to detect things like failovers or reboots

Each alarm is connected to an SNS topic. When a threshold is crossed, the alarm sends a message to this topic. The SNS topic triggers a Lambda function that:

- Sends an alert email to the system admins
- Automatically creates a support ticket in osTicket using its API, with details about the issue like the instance ID, the metric that caused the alert, and the time it happened

To test the setup, we intentionally put load on the RDS instance to trigger these alarms. This confirms that the system works end-to-end—from detecting a problem, to sending alerts, to creating tickets.

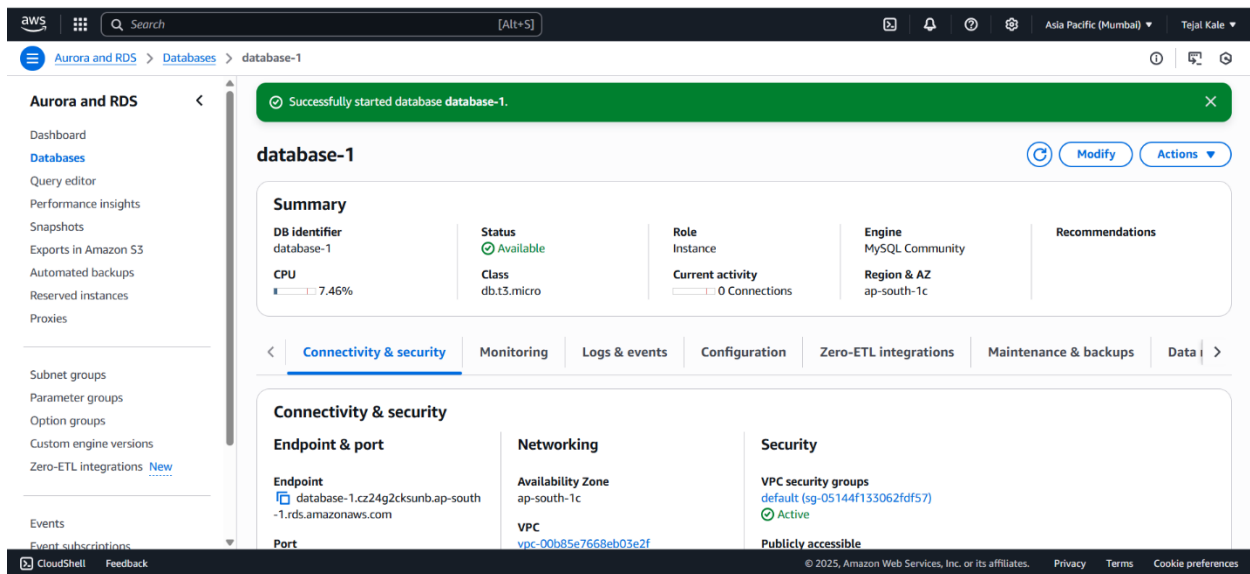
This setup helps find and fix problems faster, which improves the availability, performance, and reliability of your database system.

3.1 Document Workflow :



4.0 Steps / Create RDS Instance

A MySQL RDS database was created manually using the AWS Console. Monitoring was turned on so that we can track important performance data like CPU usage, memory, and storage space. Basic monitoring gives updates every 5 minutes. If enhanced monitoring is also enabled, it provides more detailed information (like memory and disk activity) every minute. This setup helps us keep an eye on the database's health and quickly catch any problems.



4.2 Create CloudWatch Alarm Terraform Modules

This Terraform project automates the creation of Amazon CloudWatch alarms for monitoring Amazon RDS instances. It covers module :

- **cpu_alarm** :Creates CloudWatch alarms for RDS instances that trigger when average CPUUtilization exceeds a defined threshold over a set evaluation period, helping detect high CPU load.
- **memory_alarm** : Creates CloudWatch alarms for RDS instances that trigger when available memory (FreeableMemory) falls below a threshold, indicating high memory usage.
- **storage_alarm** : Creates CloudWatch alarms for RDS instances that trigger when available storage (FreeStorageSpace) drops below a defined threshold, indicating low disk space.
- **db_connection** : Creates CloudWatch alarms that trigger when the number of database connections drops to zero, indicating possible connectivity or availability issues.
- **rds_events** : Monitors RDS events such as reboots, failovers, and maintenance actions using CloudWatch Event Rules and targets for real-time alerting.

Modules Structure

```
rds_alarms/
├─ main.tf
├─ terraform.tfvars
├─ variables.tf
├─ rds_ids.json
├─ terraform.tfstate
├─ modules/
│   ├─ cpu_alarm/
│   ├─ memory_alarm/
│   ├─ storage_alarm/
│   ├─ db_connection_alarms/
│   └─ rds_events/
```

4.4: Apply Terraform Scripts

Initialize and apply the Terraform code:

terraform init

terraform apply

4.5: Cloudwatch Alarms

After applying the Terraform script, the following CloudWatch alarms are automatically created for the specified RDS instances:

The screenshot above shows multiple CloudWatch alarms provisioned via Terraform for metrics like CPU utilization, memory usage, storage space, database connections, and RDS event monitoring.

Each alarm is in OK state by default and transitions to ALARM when the configured threshold is breached.

Alarms (4)

☐ Hide Auto Scaling alarms

Clear selection

Create composite alarm

Actions

Create alarm

Alarm state: Any

Alarm type: Any

Actions status: Any

< 1 >

<input type="checkbox"/>	Name	State	Last state update (Local)	Conditions	Actions
<input type="checkbox"/>	RDS-database-1-Low-FreeableMemory	OK	2025-07-25 16:18:24	FreeableMemory < 50000000 for 1 datapoints within 5 minutes	Actions enabled Warning
<input type="checkbox"/>	RDS-database-1-Low-FreeStorage	OK	2025-07-25 16:18:09	FreeStorageSpace < 10000000000 for 1 datapoints within 2 minutes	Actions enabled Warning
<input type="checkbox"/>	RDS-database-1-DB-connect	OK	2025-07-25 16:17:48	DatabaseConnections > 2 for 1 datapoints within 1 minute	Actions enabled Warning
<input type="checkbox"/>	RDS-database-1-CPU-High	OK	2025-07-25 16:17:39	CPUUtilization > 40 for 1 datapoints within 2 minutes	Actions enabled Warning

4.5.1: Simulate Load to Trigger Alarms

To test the alarm setup, we connected to the RDS instance using MySQL Workbench and ran heavy queries to create load on the database. This increased CPU, memory, and disk usage. We then monitored the metrics in CloudWatch to see the alarms get triggered.

4.5.2: Alarm Triggers SNS Topic

When the RDS metrics (like high CPU, low memory, or low storage) cross the set threshold, the corresponding CloudWatch alarm changes its status to **ALARM**. As soon as this happens, the alarm automatically sends a notification to an **SNS (Simple Notification Service) topic**. This topic acts as a messaging hub and forwards the alert to all its subscribers—such as Lambda functions, email addresses, or other systems—for further action. This ensures that the issue is detected and reported instantly.

The screenshot displays the Amazon SNS console for the topic 'rds-cross-ac'. The 'Details' tab shows the topic name, ARN, and type. The 'Subscriptions' tab lists two subscribers: a Lambda function and an email address, both with a 'Confirmed' status.

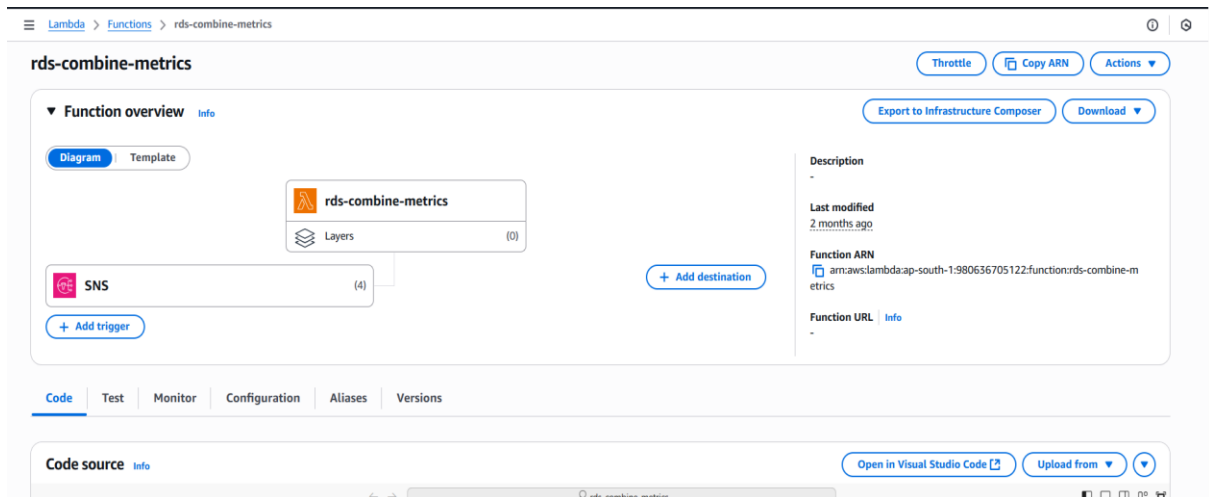
Below the console, an email notification from 'AWS Notifications' is shown. The subject is 'ALARM: "RDS-database-latest-CPU-High" in Asia Pacific (Mumbai)'. The email body states that the alarm has entered the ALARM state due to a threshold crossing. It includes a link to view the alarm in the AWS Management Console and a list of alarm details.

Alarm Details:

- Name: RDS-database-latest-CPU-High
- Description: RDS instance database-latest CPU usage too high
- State Change: OK -> ALARM
- Reason for State Change: Threshold Crossed: 1 datapoint [51.883882605281386 (22/05/25 10:47:00)] was greater than the threshold (40.0).
- Timestamp: Thursday 22 May, 2025 10:50:56 UTC
- AWS Account: 357820945061
- Alarm Arn: arn:aws:cloudwatch:ap-south-1:357820945061:alarm:RDS-database-latest-CPU-High

4.5.3: SNS Triggers Lambda Function

When the alarm sends a message to the SNS topic, the **SNS topic automatically triggers a Lambda function** that is subscribed to it. This means the Lambda function runs as soon as the alert is sent. The function then takes further action—like sending an email or creating a support ticket—based on the alarm details.



4.6: Lambda Function Actions

Sends a POST request to the osTicket API, which automatically creates a new support ticket. The request includes important alarm details like the RDS instance name, metric breached, threshold, and time of the alert.

Assigned To:	— Unassigned —	Help Topic:	None
SLA Plan:	Default SLA	Last Message:	5/22/25 17:33
Due Date:	5/23/25 11:33	Last Response:	

Ticket Thread (1)
Tasks

Chandrakant Borle posted 5/22/25 17:33 🚨 RDS Alert - (Ac Name: Tejal_k Ac No: 357820945061)

🚨 RDS CPU Utilization Alert

- Alarm: RDS-database-latest-CPU-High
- Region: Asia Pacific (Mumbai)
- Time: 22/05/2025 12:03:09 UTC
- DB Instance: database-latest
- CPU Usage: Threshold Crossed: [52.68 > threshold (40.0)]

Created by **Chandrakant Borle** 5/22/25 17:33