To create a simple **BookMyShow-like website** in PHP where users can view movies and **book tickets**, you can modify your current code to:

1. Replace users with movies and bookings.
2. Allow users to **view movies**, **book tickets**, and **view/delete bookings**.

**🔧 1. Database Tables Suggestion**

-- movies table

CREATE TABLE movies (

id INT AUTO\_INCREMENT PRIMARY KEY,

title VARCHAR(255),

show\_time DATETIME

);

-- bookings table

CREATE TABLE bookings (

id INT AUTO\_INCREMENT PRIMARY KEY,

movie\_id INT,

customer\_name VARCHAR(100),

tickets INT,

FOREIGN KEY (movie\_id) REFERENCES movies(id)

);

**📄 index.php – List Movies and Book Tickets**

<?php

include 'db\_connect.php';

if (!isset($conn) || $conn->connect\_error) {

die("Database connection failed.");

}

$result = $conn->query("SELECT \* FROM movies");

if (!$result) {

die("Query failed: " . $conn->error);

}

?>

<!DOCTYPE html>

<html>

<head>

<title>BookMyShow – Movies</title>

</head>

<body>

<h2>Available Movies</h2>

<table border="1">

<tr><th>ID</th><th>Title</th><th>Show Time</th><th>Action</th></tr>

<?php while ($row = $result->fetch\_assoc()): ?>

<tr>

<td><?= htmlspecialchars($row['id']) ?></td>

<td><?= htmlspecialchars($row['title']) ?></td>

<td><?= htmlspecialchars($row['show\_time']) ?></td>

<td>

<a href="book.php?movie\_id=<?= $row['id'] ?>">Book Ticket</a>

</td>

</tr>

<?php endwhile; ?>

</table>

<br>

<a href="bookings.php">View Bookings</a>

</body>

</html>

<?php $conn->close(); ?>

**📄 book.php – Book a Ticket**

<?php

include 'db\_connect.php';

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

$movie\_id = $\_POST['movie\_id'];

$customer\_name = $\_POST['customer\_name'];

$tickets = $\_POST['tickets'];

$stmt = $conn->prepare("INSERT INTO bookings (movie\_id, customer\_name, tickets) VALUES (?, ?, ?)");

$stmt->bind\_param("isi", $movie\_id, $customer\_name, $tickets);

$stmt->execute();

$stmt->close();

header("Location: bookings.php");

exit;

}

$movie\_id = $\_GET['movie\_id'] ?? 0;

$result = $conn->query("SELECT \* FROM movies WHERE id = $movie\_id");

$movie = $result->fetch\_assoc();

?>

<!DOCTYPE html>

<html>

<head><title>Book Ticket</title></head>

<body>

<h2>Book Ticket for <?= htmlspecialchars($movie['title']) ?></h2>

<form method="post">

<input type="hidden" name="movie\_id" value="<?= $movie['id'] ?>">

Name: <input type="text" name="customer\_name" required><br><br>

Number of Tickets: <input type="number" name="tickets" required min="1"><br><br>

<input type="submit" value="Book Now">

</form>

<br>

<a href="index.php">Back to Movies</a>

</body>

</html>

**📄 bookings.php – View All Bookings**

<?php

include 'db\_connect.php';

$query = "

SELECT bookings.id, movies.title, bookings.customer\_name, bookings.tickets

FROM bookings

JOIN movies ON bookings.movie\_id = movies.id

";

$result = $conn->query($query);

?>

<!DOCTYPE html>

<html>

<head><title>All Bookings</title></head>

<body>

<h2>Bookings</h2>

<table border="1">

<tr><th>ID</th><th>Movie</th><th>Customer</th><th>Tickets</th><th>Action</th></tr>

<?php while ($row = $result->fetch\_assoc()): ?>

<tr>

<td><?= $row['id'] ?></td>

<td><?= htmlspecialchars($row['title']) ?></td>

<td><?= htmlspecialchars($row['customer\_name']) ?></td>

<td><?= $row['tickets'] ?></td>

<td><a href="delete\_booking.php?id=<?= $row['id'] ?>" onclick="return confirm('Cancel this booking?')">Delete</a></td>

</tr>

<?php endwhile; ?>

</table>

<br>

<a href="index.php">Back to Movies</a>

</body>

</html>

**📄 delete\_booking.php – Delete a Booking**

<?php

include 'db\_connect.php';

$id = $\_GET['id'] ?? 0;

$conn->query("DELETE FROM bookings WHERE id = $id");

header("Location: bookings.php");

?>

**✅ db\_connect.php Example**

<?php

$host = 'localhost';

$user = 'root';

$password = '';

$dbname = 'bookmyshow';

$conn = new mysqli($host, $user, $password, $dbname);

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

?>

Would you like me to help with:

* HTML/CSS styling?
* Admin interface to **add movies**?
* Export to PDF or email confirmation?

Let me know!

Here’s a simple Dockerfile to containerize your **PHP + MySQL-based BookMyShow-like ticket booking app**.

**✅ Dockerfile**

# Use official PHP image with Apache

FROM php:8.1-apache

# Install mysqli extension and other dependencies

RUN docker-php-ext-install mysqli

# Enable Apache mod\_rewrite (optional, but useful)

RUN a2enmod rewrite

# Copy project files to Apache document root

COPY . /var/www/html/

# Set permissions

RUN chown -R www-data:www-data /var/www/html

# Expose port 80

EXPOSE 80

# Start Apache in the foreground

CMD ["apache2-foreground"]

**🧪 Project Directory Structure**

Make sure your structure looks like:

bookmyshow/

│

├── Dockerfile

├── db\_connect.php

├── index.php

├── book.php

├── bookings.php

├── delete\_booking.php

└── ... (other files)

**🛠️ How to Build and Run**

1. **Build the image**:
2. docker build -t bookmyshow-php .
3. **Run the container**:
4. docker run -d -p 8080:80 --name bookmyshow bookmyshow-php

Then open your browser at:  
👉 http://localhost:8080

**⚠️ MySQL Integration**

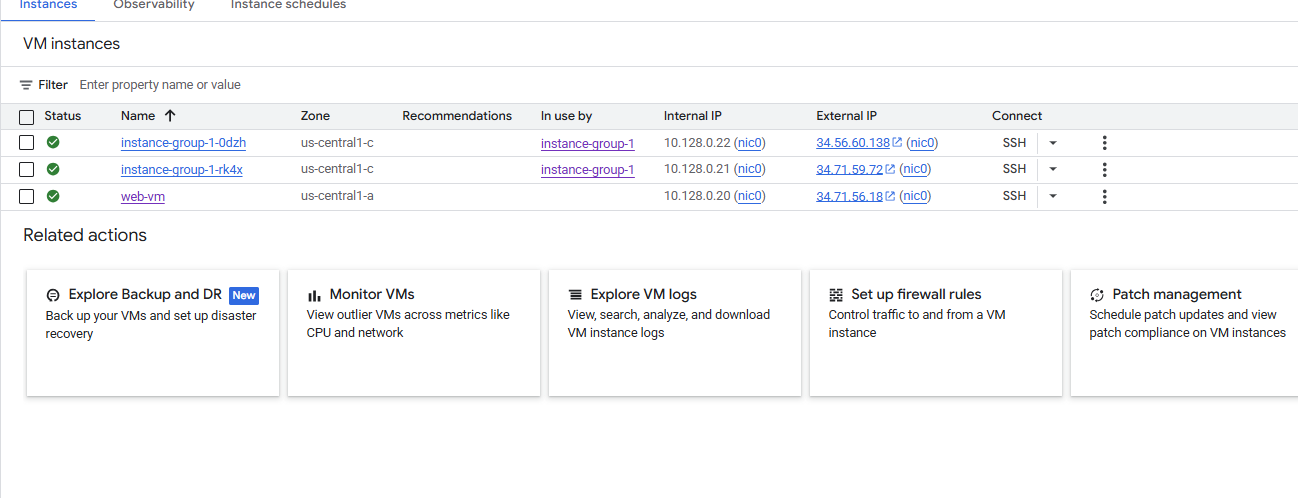
If you're using **MySQL**, you'll also need a container for MySQL. Use docker-compose.yml to connect both containers easily.

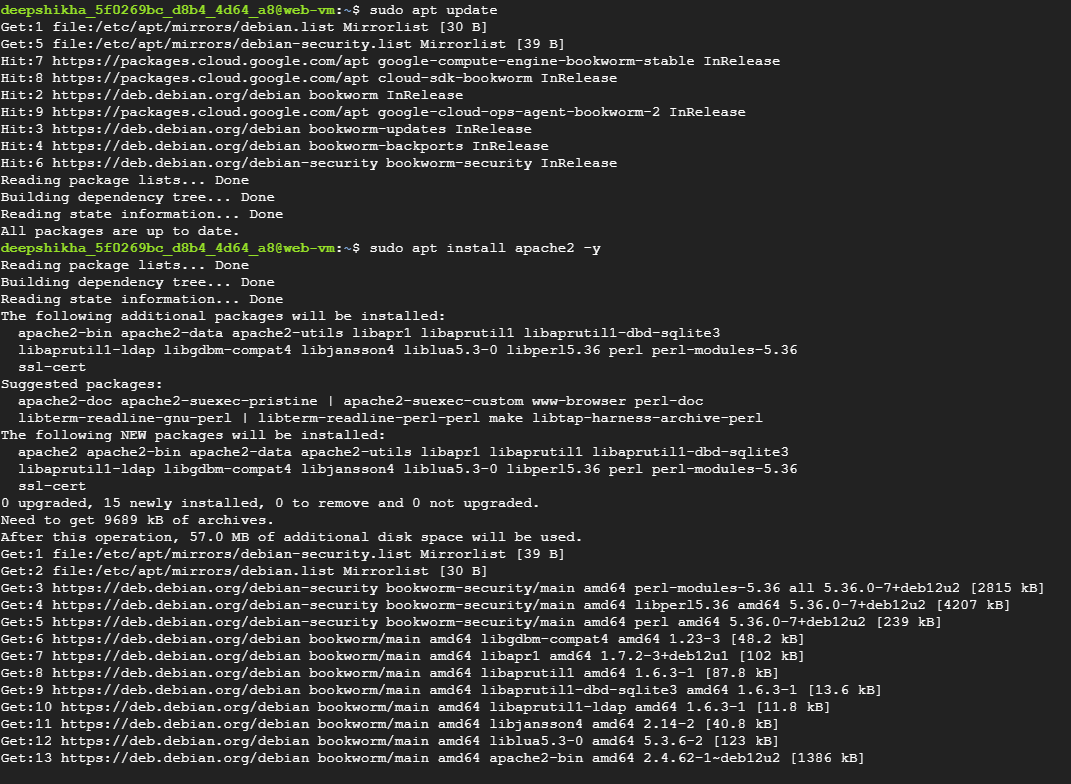
Do you want the docker-compose.yml as well for MySQL + PHP setup?

**Task 4 -** **Use Cloud CDN for low-latency delivery**

**Install a Web Server**

* + SSH into the VM → Run:

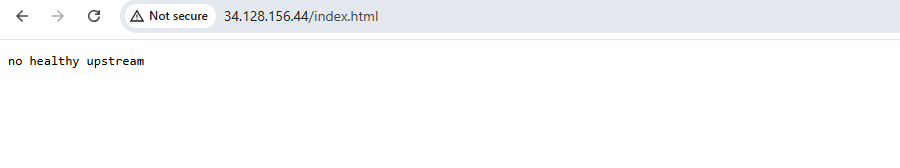




Output

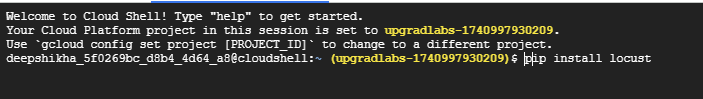


Test the CDN

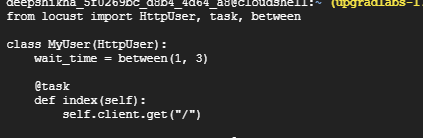


**Task 5 - Simulate extreme loads using Google Cloud Performance Testing Tools**

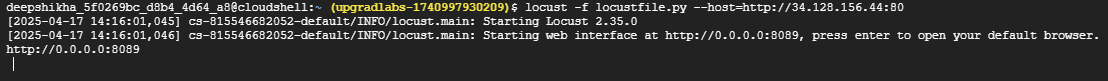
Option 1: Using Cloud Shell + Locust



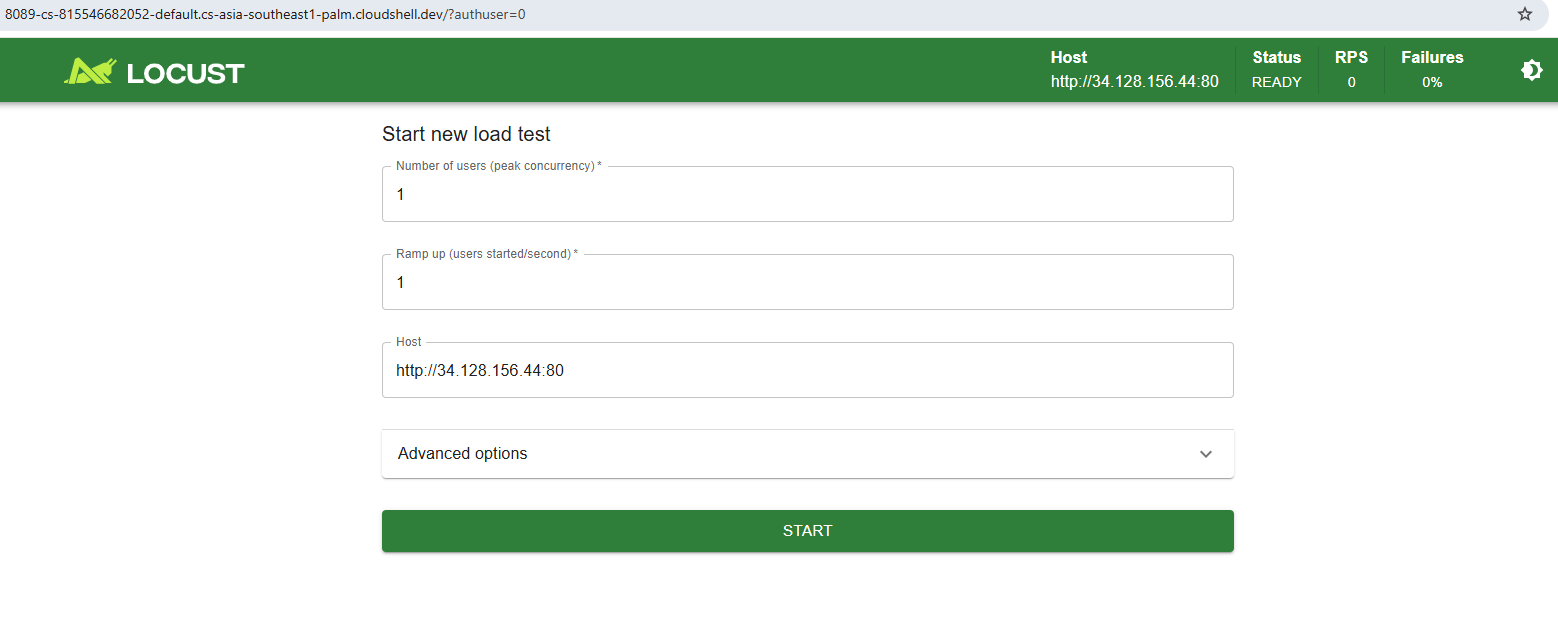
Create locustfile.py



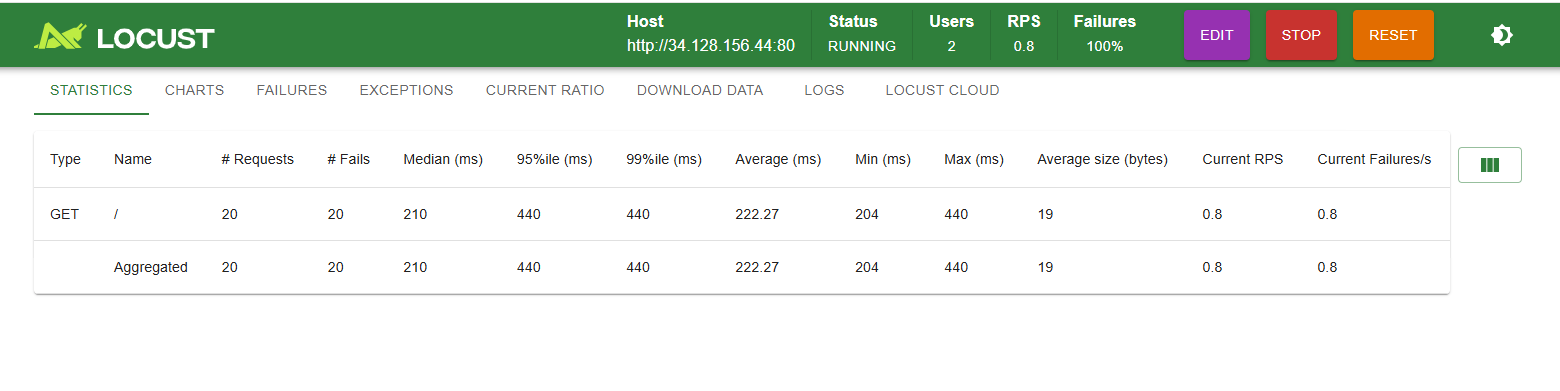
locust -f locustfile.py --host=http:// 34.128.156.44:80



Open URL given in Cloud Shell (usually port 8089)

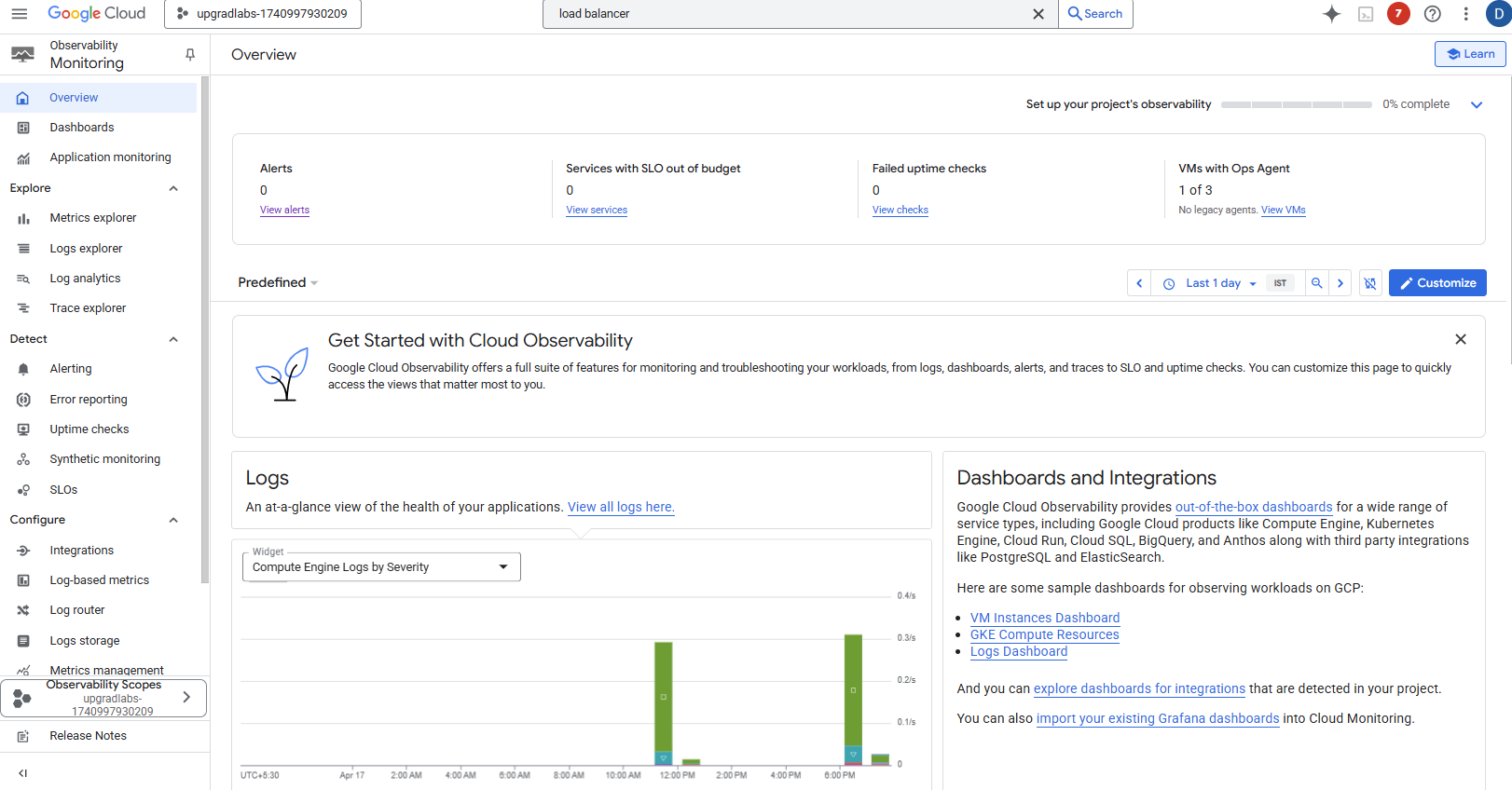


Start test with 10–100 users and monitor results

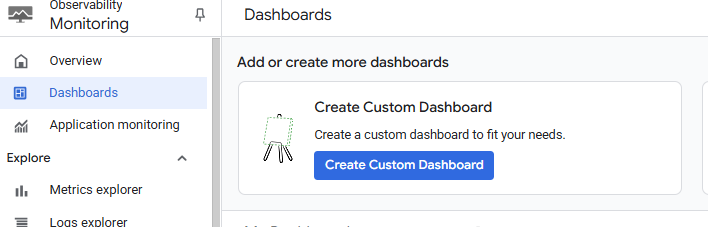


**Task 6 - Evaluate performance metrics and adjust scaling parameters**

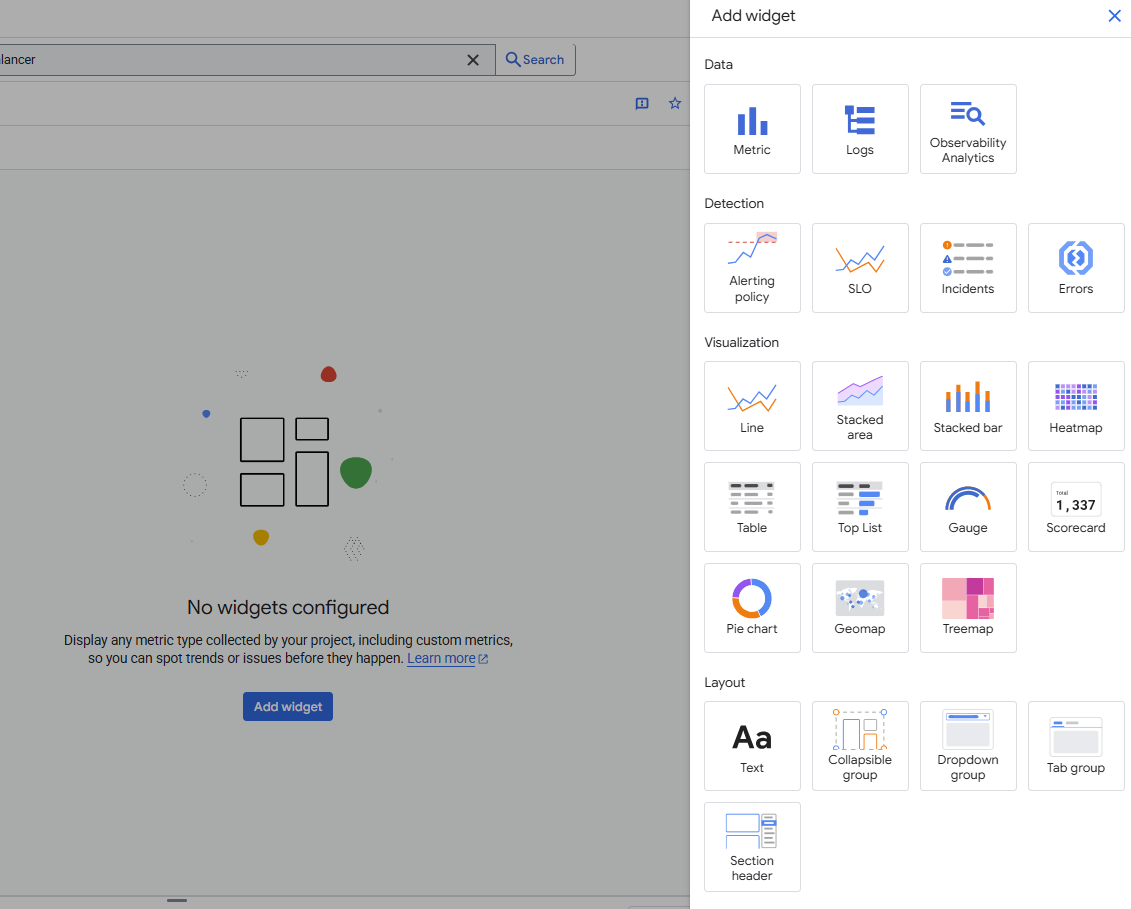
Go to Monitoring Dashboard

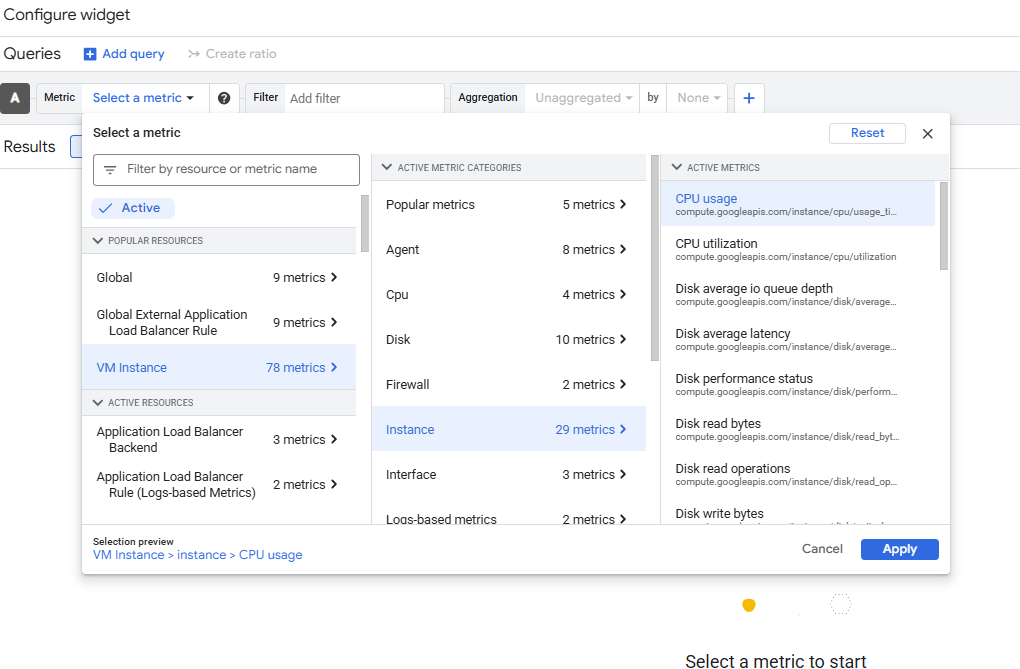
****

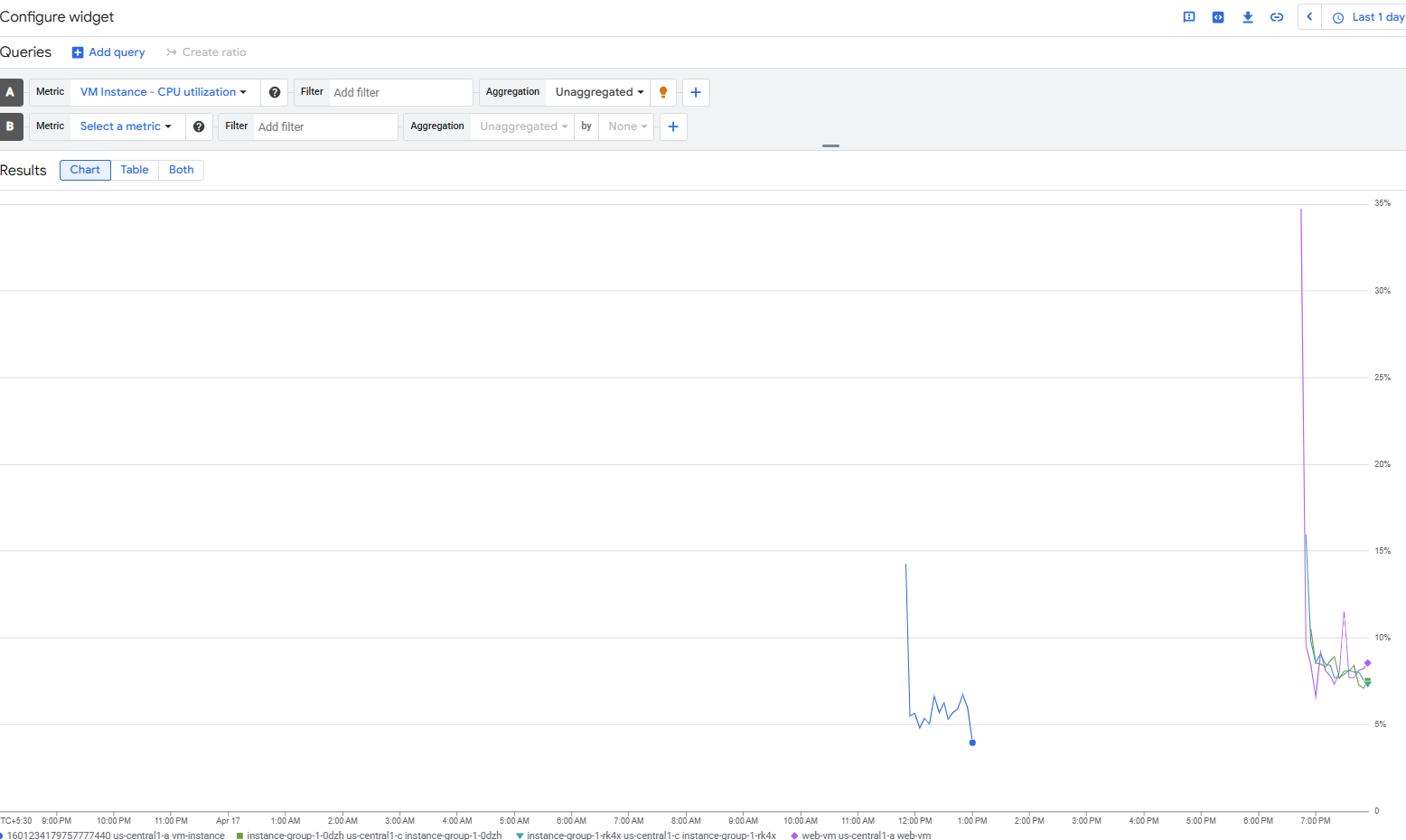
Create a Custom Dashboard



Select line option







Create alert policy

