Full-Stack Cloud Application – Deployed with AWS EC2, RDS, Cognito, CloudFront, WAF & CloudWatch

🔷 Project Overview:

This project demonstrates a secure, scalable full-stack application deployed using key AWS services. The app includes a user sign-up/sign-in system, a data submission form, and real-time monitoring — all hosted on the cloud.

👇 Key Features Implemented:

✅ 1. Cloud-hosted Backend

A Node.js server was deployed on Amazon EC2, publicly accessible via browser.

Backend connected to Amazon RDS (MySQL) to store form data.

✅ 2. User Authentication with Cognito

Integrated Amazon Cognito to handle secure user login and signup.

Hosted login page using Cognito’s hosted UI.

After successful login, users are redirected back to the app with a personalized welcome and user data.

✅ 3. Form Submission + Data Persistence

Authenticated users are shown a form.

Data entered is securely saved into the RDS database.

✅ 4. Content Delivery with CloudFront

App is distributed globally via Amazon CloudFront CDN, increasing speed and performance.

✅ 5. Security with AWS WAF

Added AWS Web Application Firewall (WAF) to protect against malicious traffic and fake URL attacks.

Verified by simulating blocked access attempts.

✅ 6. Real-Time Monitoring with CloudWatch

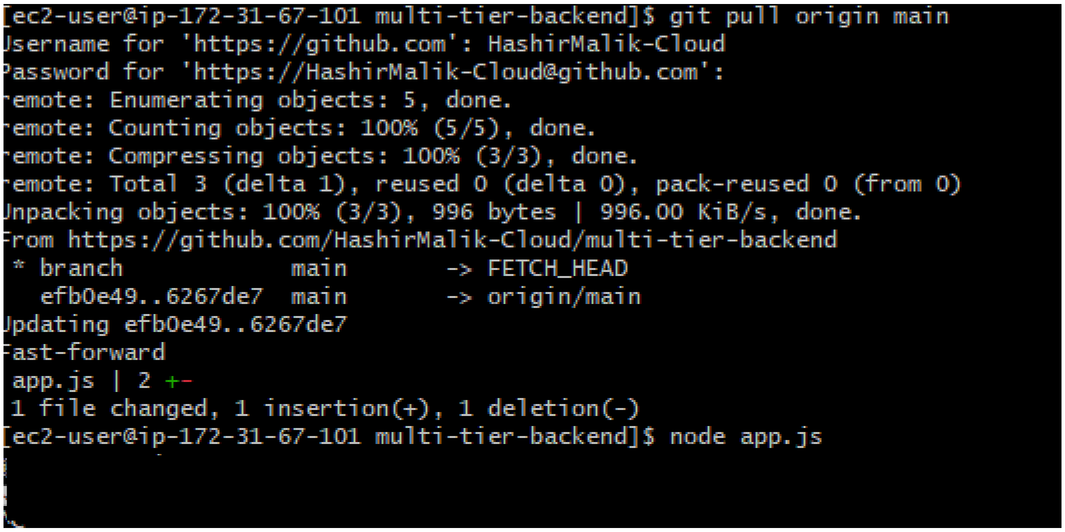
Installed and configured CloudWatch Agent on EC2 instance.

Custom logs and system-level logs are streamed in real-time.

Helpful for performance tracking and issue debugging.

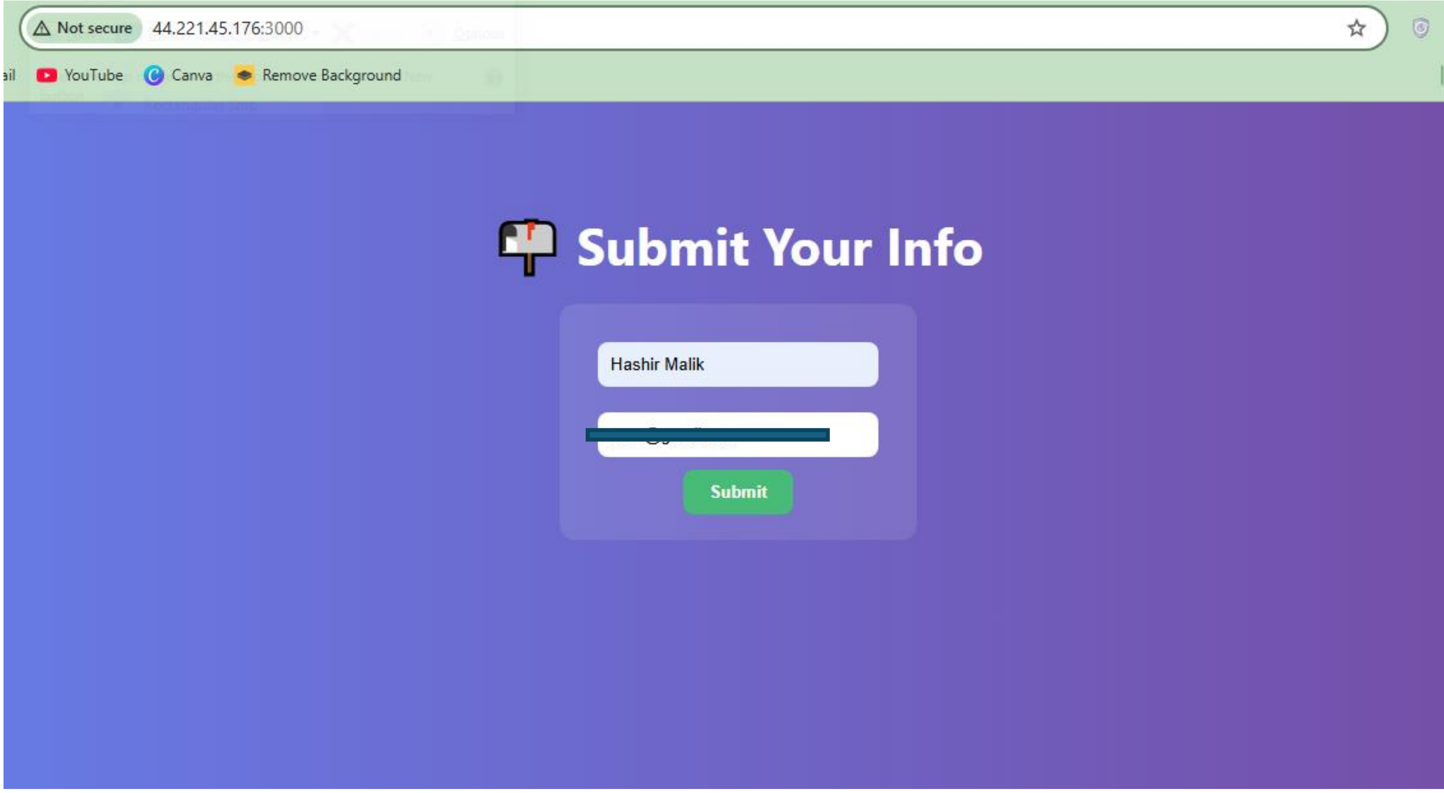
📷 Screenshots (Coming Below)

1. ✅ Pulled code from GitHub



```
[ec2-user@ip-172-31-67-101 multi-tier-backend]$ git pull origin main
Username for 'https://github.com': HashirMalik-Cloud
Password for 'https://HashirMalik-Cloud@github.com':
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 996 bytes | 996.00 KiB/s, done.
From https://github.com/HashirMalik-Cloud/multi-tier-backend
 * branch            main       -> FETCH_HEAD
   efb0e49..6267de7  main       -> origin/main
Updating efb0e49..6267de7
Fast-forward
 app.js | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
[ec2-user@ip-172-31-67-101 multi-tier-backend]$ node app.js
```

2. ✅After running my app, tested it in browser (Form Submission)

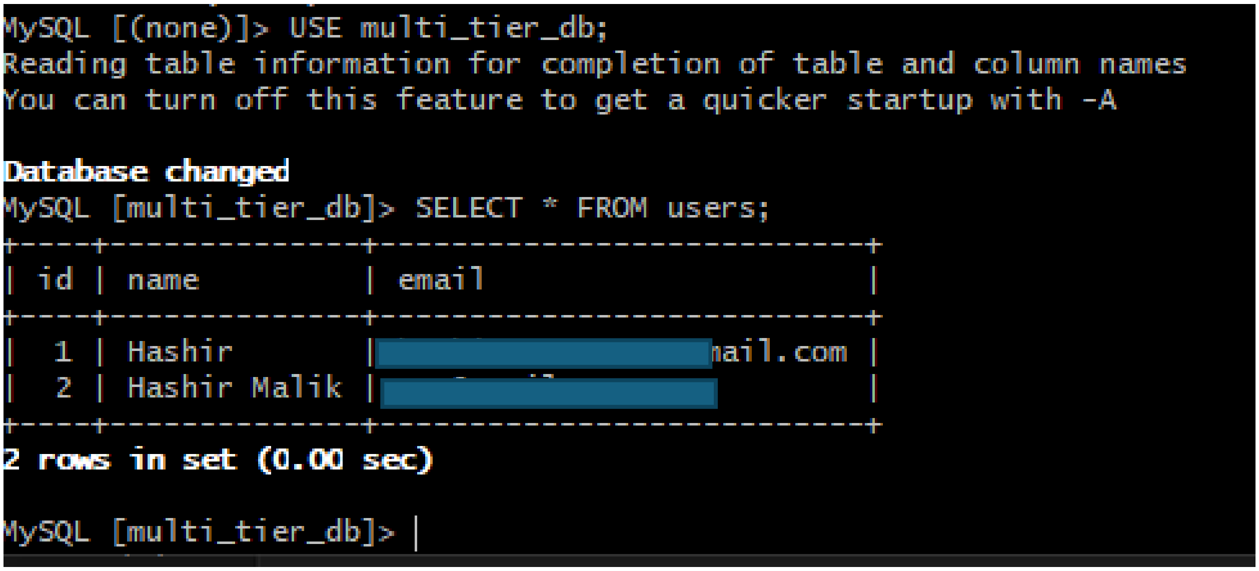


Submit Your Info

Hashir Malik

[redacted]

3. ☑ Data stored in Database

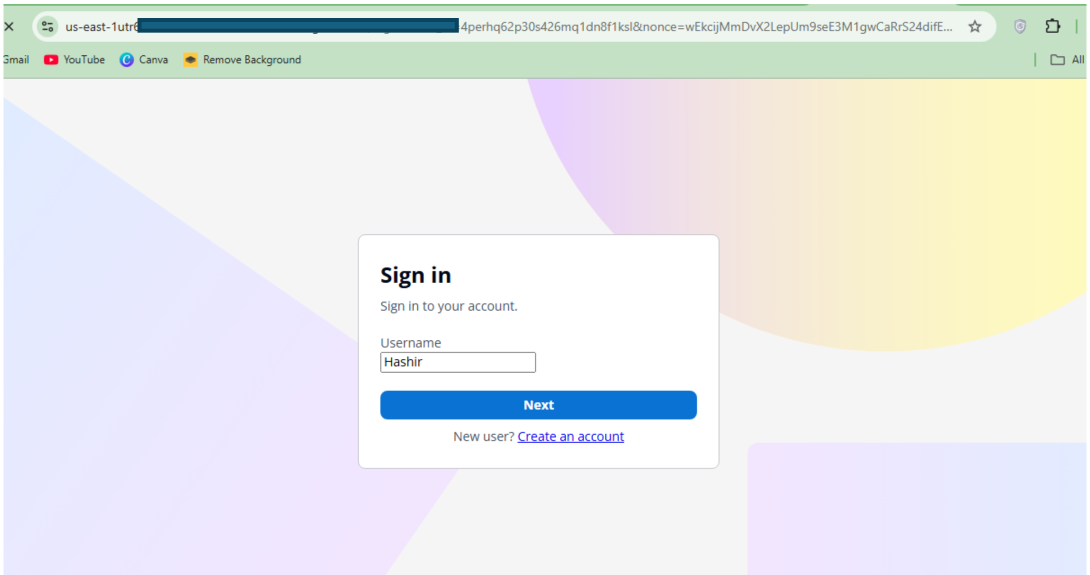


```
MySQL [(none)]> USE multi_tier_db;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MySQL [multi_tier_db]> SELECT * FROM users;
+----+---------------+----------------------------+
| id | name          | email                      |
+----+---------------+----------------------------+
|  1 | Hashir        |[redacted]ail.com |
|  2 | Hashir Malik  | [redacted]                 |
+----+---------------+----------------------------+
2 rows in set (0.00 sec)

MySQL [multi_tier_db]> |
```

4. ☑ Sign-In page (Cognito)


us-east-1utr[redacted]4perhq62p30s426mq1dn8f1ksl&nonce=wEkcijMmDvX2LepUm9seE3M1gwCaRrS24difE...
Gmail
YouTube
Canva
Remove Background
All

Sign in

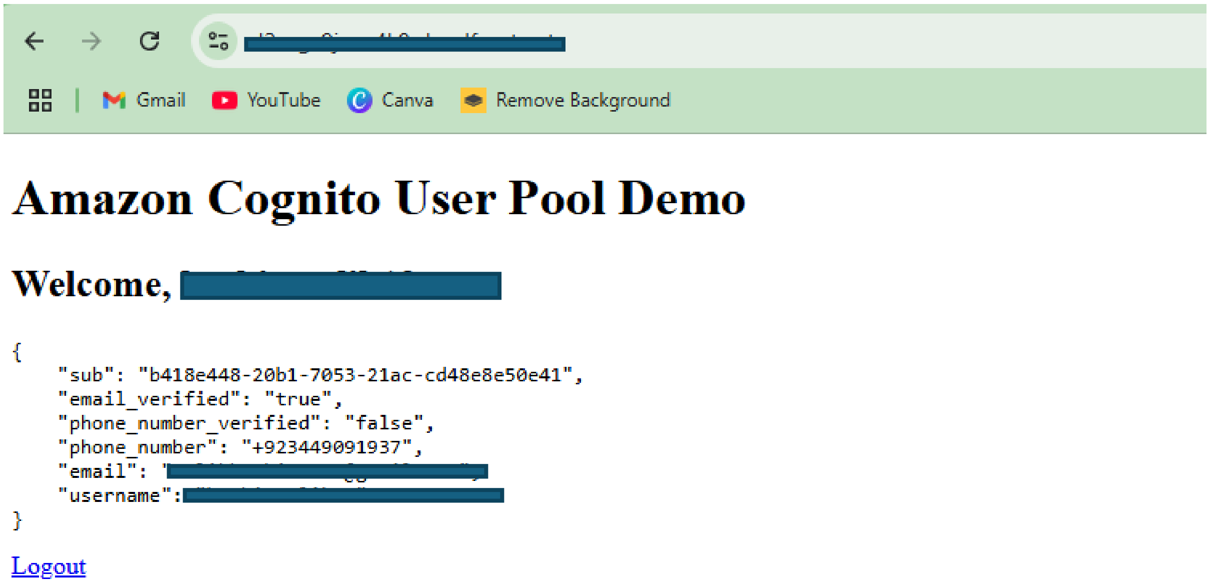
Sign in to your account.

Username: Hashir

Next

New user? Create an account

5. ☑ Signed-In successfully



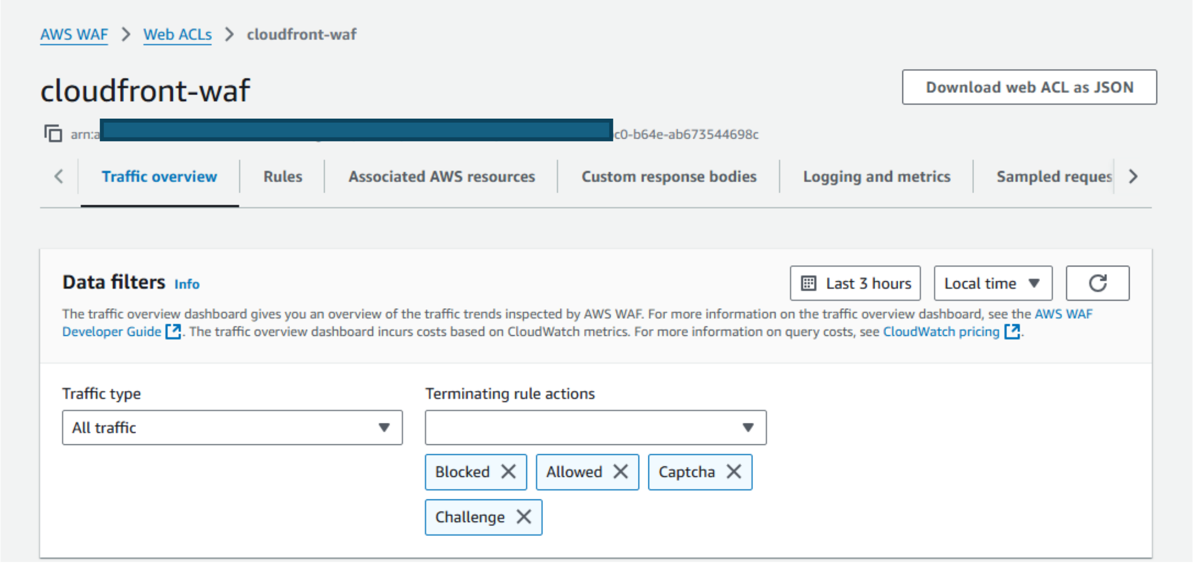
Amazon Cognito User Pool Demo

Welcome, [redacted]

```
{
    "sub": "b418e448-20b1-7053-21ac-cd48e8e50e41",
    "email_verified": "true",
    "phone_number_verified": "false",
    "phone_number": "+923449091937",
    "email": [redacted]
    "username":[redacted]
}
```

Logout

6. ☑ Security with AWS WAF



cloudfront-waf

arn:a[redacted]c0-b64e-ab673544698c

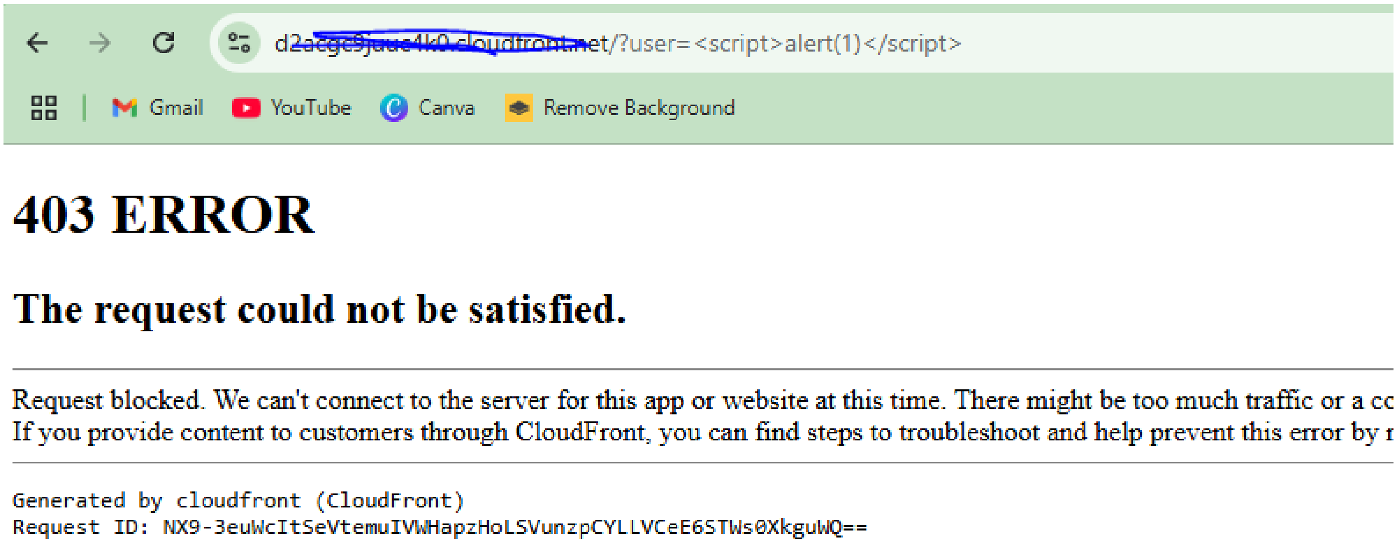
Data filters Info

The traffic overview dashboard gives you an overview of the traffic trends inspected by AWS WAF. For more information on the traffic overview dashboard, see the AWS WAF Developer Guide. The traffic overview dashboard incurs costs based on CloudWatch metrics. For more information on query costs, see CloudWatch pricing.

Traffic type: All traffic

Terminating rule actions: Blocked, Allowed, Captcha, Challenge

7. ☑ Blocked Unauthorized Request (WAF)



d[redacted]et/?user=<script>alert(1)</script>

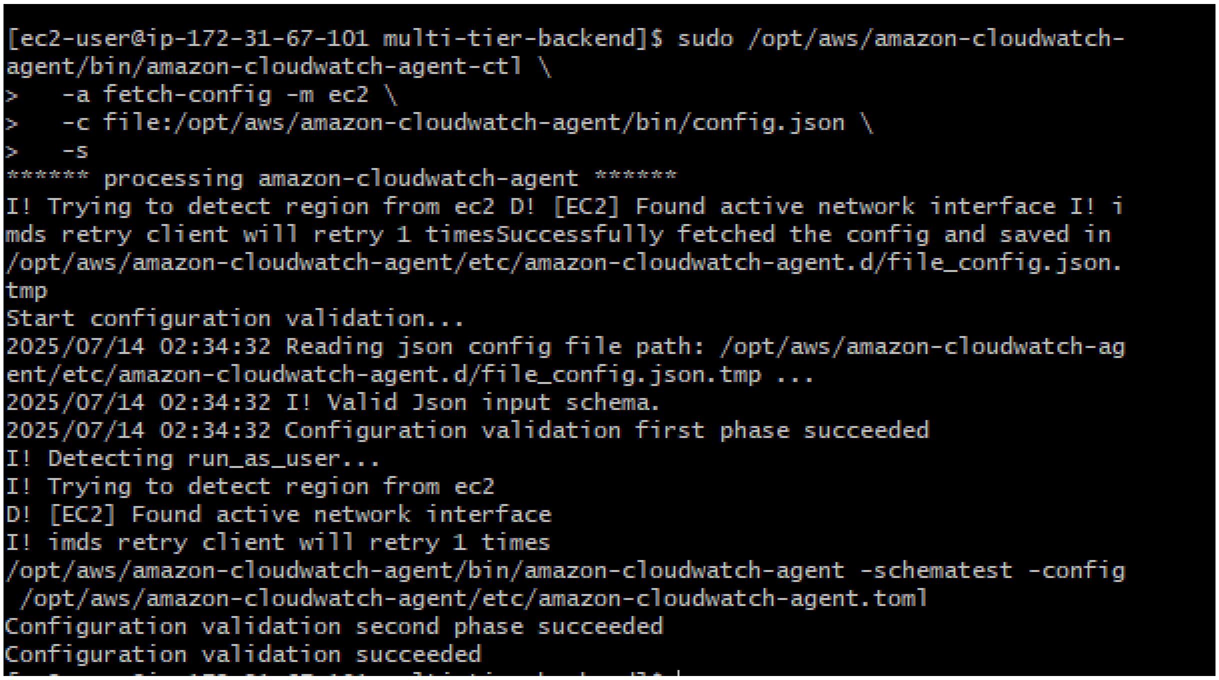
403 ERROR

The request could not be satisfied.

Request blocked. We can't connect to the server for this app or website at this time. There might be too much traffic or a cc
If you provide content to customers through CloudFront, you can find steps to troubleshoot and help prevent this error by r

```
Generated by cloudfront (CloudFront)
Request ID: NX9-3euWcItSeVtemuIVWHapzHoLSVunzpCYLLVCeE6STWs0XkguWQ==
```

8. ☑ Logs in CloudWatch



```
[ec2-user@ip-172-31-67-101 multi-tier-backend]$ sudo /opt/aws/amazon-cloudwatch-
agent/bin/amazon-cloudwatch-agent-ctl \
>   -a fetch-config -m ec2 \
>   -c file:/opt/aws/amazon-cloudwatch-agent/bin/config.json \
>   -s
****** processing amazon-cloudwatch-agent ******
I! Trying to detect region from ec2 D! [EC2] Found active network interface I! i
mds retry client will retry 1 timesSuccessfully fetched the config and saved in
/opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.d/file_config.json.
tmp
Start configuration validation...
2025/07/14 02:34:32 Reading json config file path: /opt/aws/amazon-cloudwatch-ag
ent/etc/amazon-cloudwatch-agent.d/file_config.json.tmp ...
2025/07/14 02:34:32 I! Valid Json input schema.
2025/07/14 02:34:32 Configuration validation first phase succeeded
I! Detecting run_as_user...
I! Trying to detect region from ec2
D! [EC2] Found active network interface
I! imds retry client will retry 1 times
/opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent -schematest -config
 /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.toml
Configuration validation second phase succeeded
Configuration validation succeeded
[ec2-user@ip-172-31-67-101 multi-tier-backend]$ |
```

💡 Why This Project Matters

This project represents the modern cloud architecture — integrating authentication, database, global distribution, security, and monitoring — all in one. It’s a perfect fit for:

Startups launching secure apps

Enterprises building scalable infrastructure

Clients needing login systems, form handling, secure storage, and cloud-based deployment