NEXT-GEN-CLOUD-SOLUTION-WITH-AWS-3-TIER-APPLICATION-OPTIMIZATION

Prerequisites:

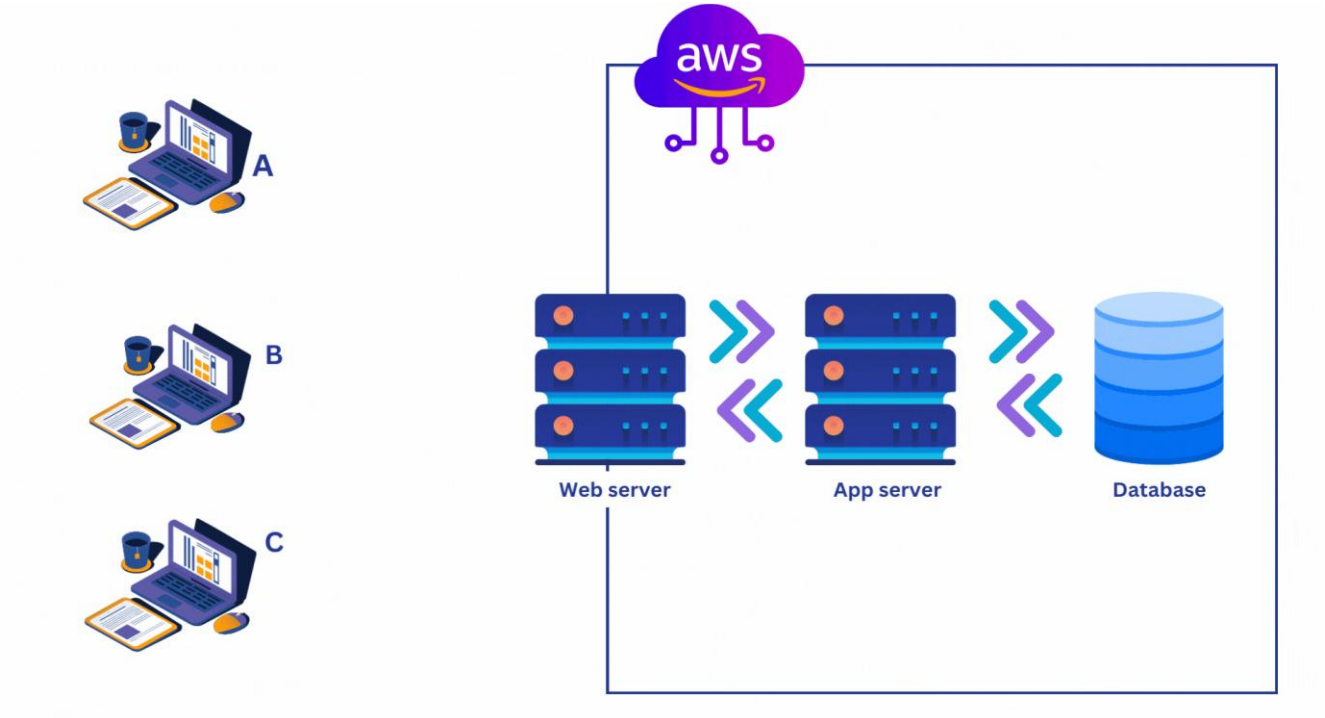
- AWS Account
- Basic knowledge of Linux

List of AWS services:

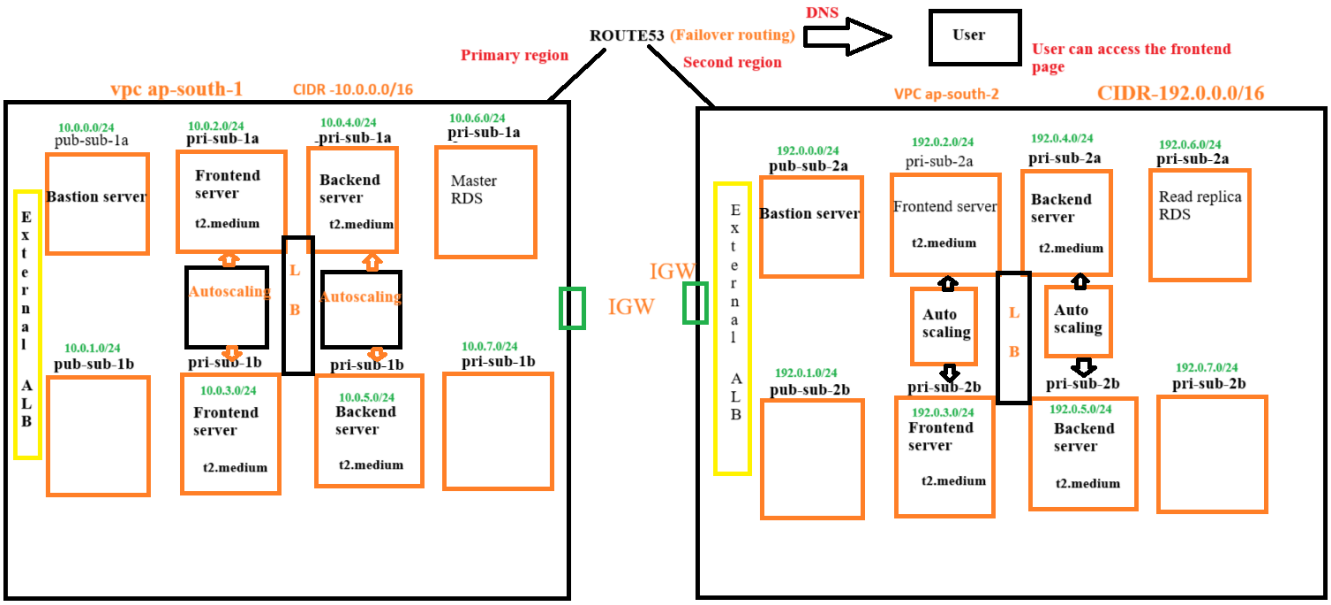
- Amazon VPC
- Amazon EC2
- Amazon Autoscaling
- Amazon RDS
- Amazon CloudWatch
- Amazon Certificate Manager
- Application load balancer
- Lambda
- Python
- EBS
- S3
- IAM role configuration and permission management

Three-tier architecture is a software architecture pattern that separates an application into three layers.

- Presentation layer (handles user interaction)
- Application layer (processes business logic and data processing)
- Data layer (manages data storage and retrieval)



Architecture of the Project



A Step-by-step guide

we are following a **Warm standby Disaster recovery strategy** so we are going to utilize two regions during our deployment. **ap-south-1 mumbai as primary** and **ap-south-2 hyderabad** as secondary region.

1. create vpc in two region and create a 8 subnets for each vpc

   - Two public subnets
   - Two frontend private subnets
   - Two backend private subnets
   - Two database private subnets

2. create rds database in project-vpc

3. Launch bastion host server in public subnet

3. Launch ubuntu server in backend subnets and connect from bastion host server. Follow the steps:

   - sudo apt update -y
   - curl -fsSL https://deb.nodesource.com/setup_18.x | sudo -E bash - &&\
     sudo apt-get install -y nodejs -y
   - sudo apt update -y
   - sudo npm install -g corepack -y
   - corepack enable
   - corepack prepare yarn@stable –activate
   - sudo npm install -g pm2

4. Then clone your git repo

- git clone https://github.com/Ramakrishnaragi/Next-Gen-Cloud-Solution-with-AWS-3-Tier-Application-Optimization.git

- cd backend
- vi .env (edit the .env file in bellow path if u dont have any .env file just create in below path)

```
##### add this mater
DB_HOST=book.rds.com #add here rds endpoint or route53 private dns
DB_USERNAME=admin    #cahnge to nyour rds user name
DB_PASSWORD=""   # change to your rds password
PORT=3306
```

then run below command in backend directory

- npm install
- npm install dotenv
- sudo pm2 start index.js --name "backendApi"

Install MySQL in backend server

- sudo apt install mysql-server -y

In the backend directory use below command to initialize the database

mysql -h book.rds.com -u admin -p<password> < test.sql

Install the cloud watch agent to send the logs to cloud watch.

5. Launch ubuntu server in frontend subnets and connect from bastion host server. Follow the steps:

- sudo apt update -y
- sudo apt install apache2 -y
- curl -fsSL https://deb.nodesource.com/setup_18.x | sudo -E bash - &&\
  sudo apt-get install -y nodejs -y
- sudo apt update -y
- sudo npm install -g corepack -y

#Core modules are built-in libraries that come with the Node.js runtime. They provide essential functionality for various tasks like file handling, HTTP requests, working with stream

- corepack enable
- corepack prepare yarn@stable --activate

#Yarn is a package manager for JavaScript projects, similar to npm (Node Package Manager). It helps developers manage dependencies (libraries and packages) required for their projects. Yarn ensures fast, reliable, and secure dependency management, making it popular among developers.

- sudo npm install -g pm2

then clone your git repo

- git clone https://github.com/Ramakrishnaragi/Next-Gen-Cloud-Solution-with-AWS-3-Tier-Application-Optimization.git
- cd client
- vi src/pages/config.js

// // const API_BASE_URL = "http://localhost:8800";

const API_BASE_URL = "http://ramcloud.shop";

// export default API_BASE_URL;

// const API_BASE_URL = process.env.REACT_APP_API_BASE_URL || "http://backend";

// export default API_BASE_URL;

// const API_BASE_URL = "REACT_APP_API_BASE_URL_PLACEHOLDER";

export default API_BASE_URL;

then go to client directory

run below commands

- npm install
- npm run build
- sudo cp -r build/* /var/www/html

Install the cloud watch agent to send the logs to cloud watch. Here write python code in lambda and integrate event bridge, when the logs are reached to cloud watch immediately the event bridge will be triggered and executes the lambda and move the logs files to s3 for cost optimization.

Create a Target groups for frontend and backend servers

Create a frontend or external load balancer and configured to target groups and same backend configure also like front-end

create a health check In route53 for primary region

create a route53 public hosted zone and create a cname record take the frontend load balancer url and select the health check and choose the primary region and add second region(#note: don’t add healthcheck for this secondary region) and create the records

Now hit the DNS it will show the front end response

Add the information also

Challenges:

1. Maintaining High Availability
2. Monitoring & Logging
3. Security at Scale
4. Challenge in Automation
5. Connections b/w frontend server, backend server and rds

Goals:

High Availability & Scalability

Security & Compliance

Cost Optimization

Disaster Recovery & Fault Tolerance

##

Finally access this DNS---> ramcloud.shop