International Journal of Aquatic Science
ISSN: 2008-8019
Vol 12, Issue 02, 2021

Building Scalable And Reliable Saas Solutions With Stateless Architecture Towards Implementing Doorstep Groceries Application

Kalyani A[1], Jothi Ram S[2]

[1]Assistant Professor, Department of Computer Applications ,PSG College of Technology Coimbatore, India
[2]PG Student, Department of Computer Applications ,PSG College of Technology Coimbatore, India

Email: [1]akk.mca@psgtech.ac.in, [2]ramukuttytheiva@gmail.com

Abstract: The Doorstep Groceries application uses stateless architecture by using a Representational State Transfer (REST) API to produce stateful behavior by augmenting HTTP to make stateless applications. In any stateful applications state is to be maintained to minimize the connection to the DB or server. The data handled in any stateless service is typically transitory and a separate back-end service is needed to store the state information. Every organisation adopts containers and tends to begin with stateless containers as it is easier to adapt to this new type of architecture. Any available compute resources such as EC2 instances can service any request because stateless applications can scale horizontally.
Highly reliable applications must set their Recovery Time Objectives (RTO) and Recovery Point Objectives (RPO) to suit the needs of their business. The Doorstep Groceries need to be available all the time to support customers at any time of the day. Stores should also not lose their customers' orders and customers should not lose their groceries. This requires The Doorstep Groceries to have a very low RTOs and RPOs.
The Doorstep Groceries will run on AWS EC2 cloud. Implementation of several application recovery automation and performance test automations ensure that The Doorstep Groceries will run as expected even under higher than average load and the application failures can be detected and recovered immediately.

Keywords – Stateless architecture, REST API, containers, EC2, Cloud, Automation, Instances

1. INTRODUCTION

E-commerce is all about selling and buying goods and services online. In the past couple of decades, eCommerce has seen an exponential growth that has truly changed the nature of many businesses. After seeing how COVID-19 has crippled the supply chain, whether day-to-day essential products or the usual luxuries typically bought online such as the popular electronics, The Doorstep Groceries will focus on the gaps in the physical and online businesses exposed by COVID-19. The Doorstep groceries will focus on grocery shopping where it will make the biggest difference as this gap is the widest. The Doorstep groceries aimed to bridge the gap and disparity in the grocery shopping experience for the

International Journal of Aquatic Science
ISSN: 2008-8019
Vol 12, Issue 02, 2021

rich and poor by building an eCommerce application that will help the society be ready when the next pandemic, natural disaster or any calamity of that scale strikes again.

Automating AWS simplifies common deployment and maintenance tasks of Amazon EC2 (Elastic Cloud Compute). Automation enables us to configure and manage instances. Automation simplifies the tasks like changing/managing the states of one or more

EARLY PHASES OF ONLINE SYSTEM

The application is developed to focus on small grocery stores in rural areas in India. These stores use very little technology to interact with customers though some may have basic inventory management and billing software. Most orders are placed in person and then picked up right then and there. Very few stores offer door delivery or online shopping let alone modern payments such as Google pay or Credit cards. Some of the existing Point of Sale (POS) softwares are slowly adding online shopping features but they are primarily built for bigger departmental stores, typically expensive to procure, complex to operate and not meant for simple use cases of the small grocery stores.

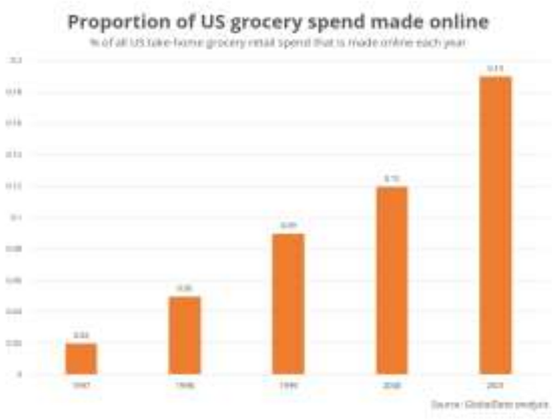


Fig: 1.1 Proportion Of Us Grocery Spent Made Online

PROPOSED WORK

The proposed work is to apply the use of stateless architecture in developing a mobile application as a platform for people to do online grocery shopping. The Doorstep Groceries intends to do this by totally providing cloud native applications and offering it as Software as a solution(SaaS). The store will be owned and operated by the same owners with the same staff and their customers will remain their own. The idea is to make grocery shopping easy via online during any pandemic situation where people are not allowed to go out and buy groceries and also the governments will be putting up restrictions.

TECHNOLOGY OVERVIEW

In designing and implementing the system, the design architecture are into three phases (i.e. The front-end, middle-tier and back-end).
This cross platform application system is designed to run specifically on any device.

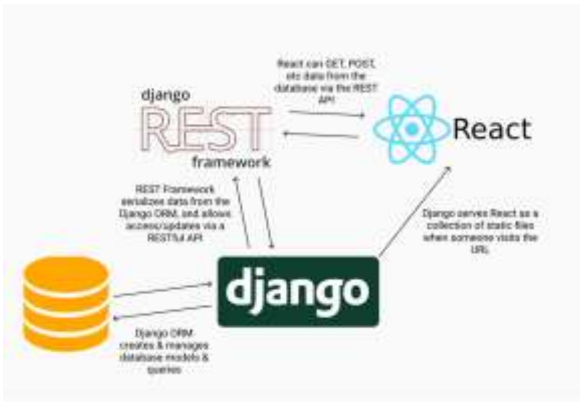

django
REST
framework
React can GET, POST, etc data from the database via the REST API
React
REST Framework serializes data from the Django ORM, and allows access/updates via a RESTful API
Django serves React as a collection of static files when someone visits the URL
django
Django ORM creates & manages database models & queries

Fig: Working Of Used Technology

A REST API is a standardized way to provide data to other applications. Typically, an API is a window into a database. The API backend handles querying the database and formatting the response. The response will be a static one, usually in JSON format of the resources that are requested. API helps the applications communicate with one another or even within themselves. Here API allows React [7] to consume information from the database.
The DJANGO REST framework makes the serialization easy. The models for the databases are defined using Python which helps to write raw SQL, for most part of the DJANGO object relational mapping handles all the database migrations and queries. Serialization is the process of converting a model to JSON. Using a serializer the JSON representation of the model can be specified on what fields it must present. The serializer will return as JSON representation so the API can parse them.

The rendering of data in application is done by passing the database queryset into serializer so that it is converted to JSON and rendered. The REST Framework router makes sure our requests end up at the right resource dynamically. A router works with a viewset to dynamically route requests.

The state management is done with Redux which is a state container that behaves consistently across client, server, and native environments. State management is essential to communicate and share data across components.

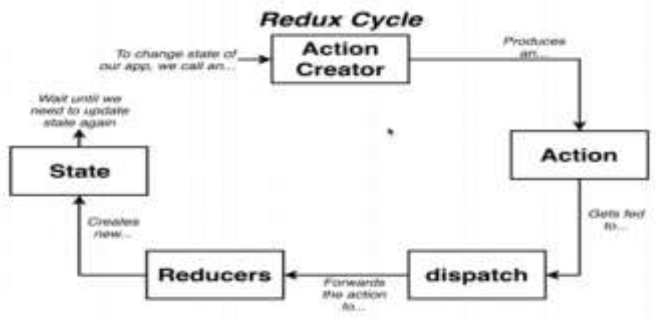

Redux Cycle
To change state of our app, we call an...
Action Creator
Produces an...
Action
Gets fed to...
dispatch
Forwards the action to...
Reducers
Creates new...
State
Wait until we need to update state again

Fig: Redux In React Framework

The way Redux works is there is a central store that holds the entire state of the application. Each component can access the stored state without having to send down props from one

component to another. There are three building parts: actions, store and reducers. The only way the data can be sent from the application to Redux store is by using actions. Reducers are pure functions that take the current state of an application, perform an action, and return a new state. These states are stored as objects, and they specify how the state of an application changes in response to an action sent to the store.
The Store holds the applications state. The state stored can be accessed and it can be updated, registered or unregistered.

IMPLEMENTATION

The modules involved in the system are as follows:

STORE Module:

In this module available stores in the franchise along with the description as shown in fig 5.1 will be displayed for the customer to select preferred store to shop.

CATEGORY Module:

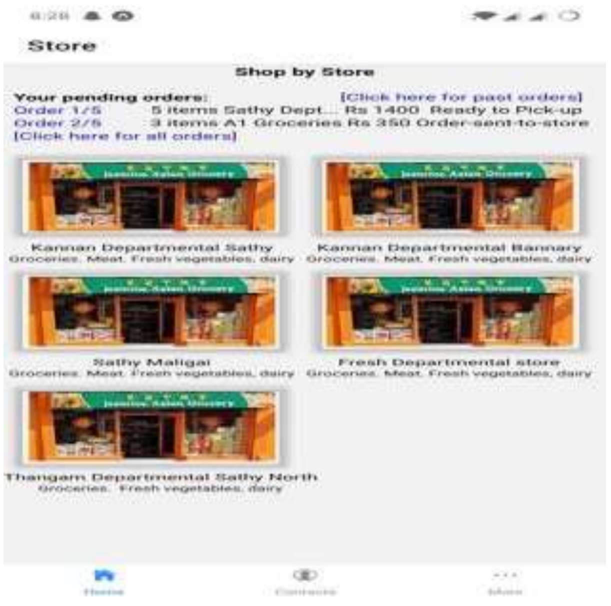
The product items available in the selected store as shown in fig 5.2 will be categorised into different categories and displayed to select the preferred category.

PRODUCT Module:

The products available in the selected category will be shown along with the details of the product (fig 5.3) and the customer can click the add to cart button to add the product to cart.

CART Module:

Cart screen shown in fig 5.4 shows the added items in the cart where the customer can modify the added items. Once the cart is populated, the customers are ready to check out their order by submitting. At any point in time before the submission, they can modify what is in the cart directly or go back to the store, category and product list screens to modify their order.


Store

Shop by Store

Your pending orders: [Click here for past orders]
Order 1/5 5 items Sathy Dept... Rs 1400 Ready to Pick-up
Order 2/5 3 items A1 Groceries Rs 350 Order-sent-to-store
[Click here for all orders]

Kannan Departmental Sathy
Groceries. Meat. Fresh vegetables, dairy

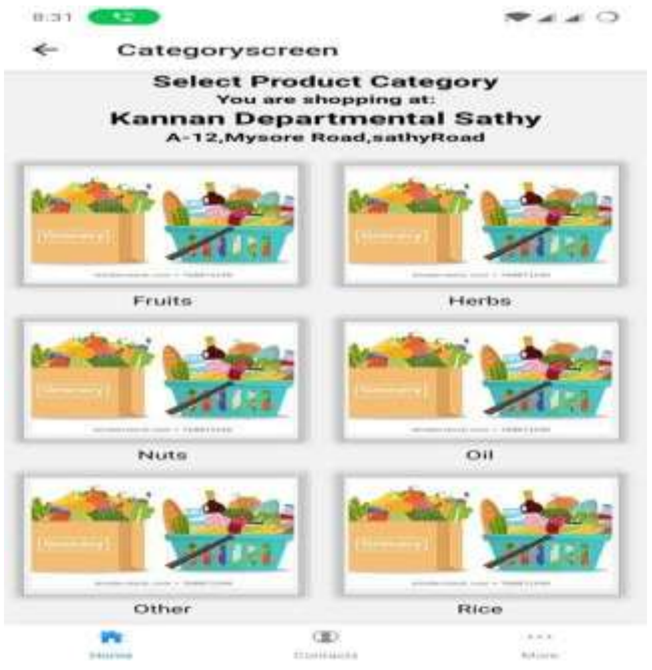
Kannan Departmental Bannary
Groceries. Meat. Fresh vegetables, dairy

Sathy Maligai
Groceries. Meat. Fresh vegetables, dairy

Fresh Departmental store
Groceries. Meat. Fresh vegetables, dairy

Thangam Departmental Sathy North
Groceries. Fresh vegetables, dairy

Fig: 5.1 Store Details



8:31

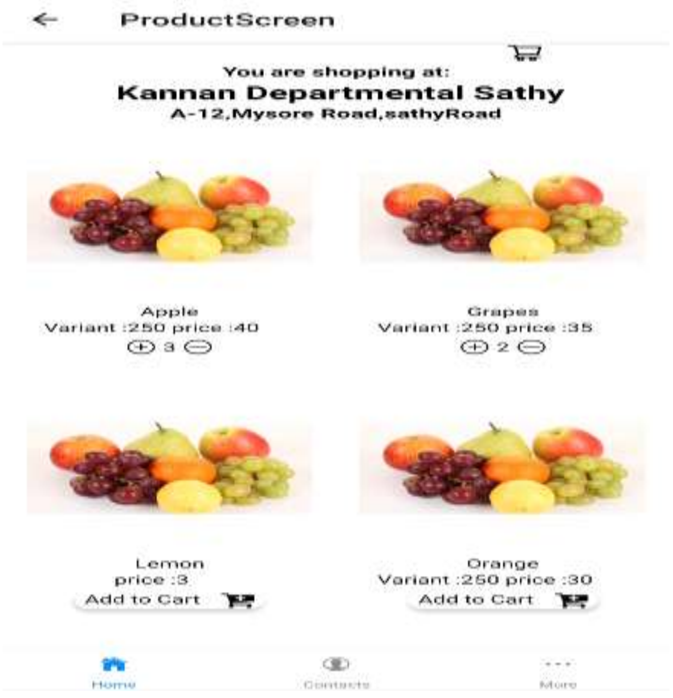
Categoryscreen

Select Product Category
You are shopping at:
Kannan Departmental Sathy
A-12,Mysore Road,sathyRoad

Fruits
Herbs
Nuts
Oil
Other
Rice

Home
Contacts
More

Fig: 5.2 Category Details



ProductScreen

You are shopping at:
Kannan Departmental Sathy
A-12,Mysore Road,sathyRoad

Apple
Variant :250 price :40
⊕ 3 ⊖

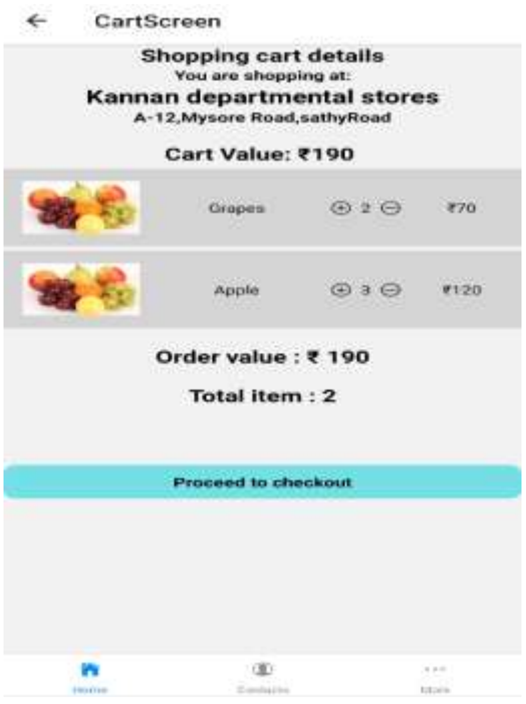
Grapes
Variant :250 price :35
⊕ 2 ⊖

Lemon
price :3
Add to Cart

Orange
Variant :250 price :30
Add to Cart

Home
Contacts
More

Fig: 5.3 Product Details


CartScreen

Shopping cart details
You are shopping at:
Kannan departmental stores
A-12,Mysore Road,sathyRoad

Cart Value: ₹190

	Grapes	2	₹70
	Apple	3	₹120

Order value : ₹ 190
Total item : 2

Proceed to checkout

fig: 5.4 cart details

ACHIEVING SCALABILITY WITH STATELESS ARCHITECTURE

The major purpose of using stateless architecture [2] is that the client's application state will never be stored on the server, but passed around from the client to every place that needs it. The statelessness helps in deploying it to multiple servers by scaling the APIs to millions of concurrent users. Since there is no session related dependency any server can handle any request. Furthermore the stateless application will reduce the complexity of programming. Incoming requests are received, processed, and forgotten. The Doorstep Groceries App uses stateless architecture because as it is an ecommerce application there will be thousands of concurrent users and various requests can be handled by different servers, and the server does not need to keep track of state between requests.

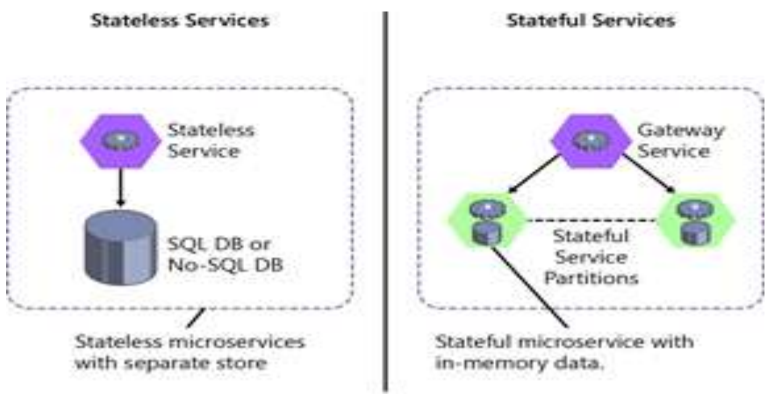

Stateless Services
Stateful Services
Stateless Service
SQL DB or No-SQL DB
Stateless microservices with separate store
Gateway Service
Stateful Service Partitions
Stateful microservice with in-memory data.

Fig:6.1 Stateless And Stateful Services

As the application is developed as a Software as a Service (SaaS), [1] establishing a stateless architecture [3] is a critical step in the evolution of SaaS technology. Users of SaaS technology can expect superior performance and consistency, as well as a lower cost benefit, due to the stateless architecture.

ACHIEVING RTO WITH APPLICATION MANAGEMENT AUTOMATION

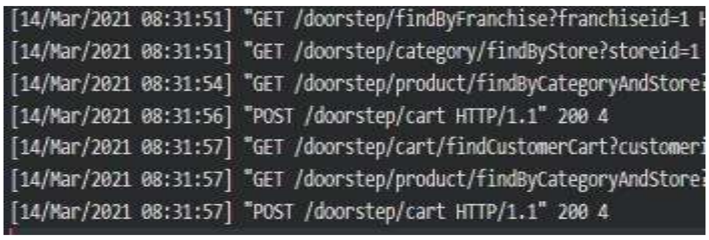
Management of the cloud infrastructure [9] is an important and difficult task as the environment has a huge number of resources and maintaining, monitoring and cleansing of resources in each and every region will take up a lot of human effort and time. The work environment consists of a number of instances and volumes made by many developers but it is necessary to keep a track on which user is making the resources. The automation is to monitor the EC2 resources and provide a tag to know who has created the instance. The rapid growth of industries resulting in a huge environment to be maintained, automated strategies are necessary for management of resources and will also help in Cost optimization of the Industry.

Utilizing a Stateless Architecture with a REST framework [8] ensures that each transaction is broken into small pieces and saved in the application database instantly so any loss of data due to application issues is reduced to a few seconds to a minute at worst. So a low RPO is inherently achieved. Additionally database backups are automated to further help avoid data loss. Achieving a low RTO [4,5] requires further work in terms of continuous application monitoring and recovery capabilities of the application.

AWS helps to use automation so that the product can build faster and more efficiently. Automation of manual operations or processes, including as deployments, development and test workflows, container management, and configuration management, can be made simple and cost-effective using AWS. But the cost spent on AWS for automation can be reduced by developing the automation process by writing automation scripts. Automation scripts help the developer to start the instance or stop the instance with ease. Further automation helps to maintain the log. It also helps to clone the Git repo.

Scripting is a simple way to automate scheduled tasks. Automating the AWS with scripting provides, starts and stops the EC2 instances during times when apps and services are not needed. Scripting can help to manage the elastic computing utility bill, thus saving the time and money by conserving computing resources and automating repetitive scheduled tasks.

Maintaining a log file of the server responses helps to identify any error if it occurs and keeps on having records dynamically. If there is any error occurring in the REST API method the log will maintain the records for future analyzing. Visualizing will present the response data in ways that help to make sense of it with the help of Kibana.

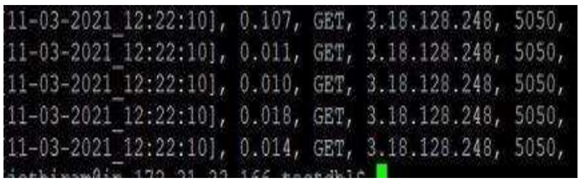


```
[14/Mar/2021 08:31:51] "GET /doorstep/findByFranchise?franchiseid=1 H
[14/Mar/2021 08:31:51] "GET /doorstep/category/findByStore?storeid=1
[14/Mar/2021 08:31:54] "GET /doorstep/product/findByCategoryAndStore
[14/Mar/2021 08:31:56] "POST /doorstep/cart HTTP/1.1" 200 4
[14/Mar/2021 08:31:57] "GET /doorstep/cart/findCustomerCart?customeri
[14/Mar/2021 08:31:57] "GET /doorstep/product/findByCategoryAndStore
[14/Mar/2021 08:31:57] "POST /doorstep/cart HTTP/1.1" 200 4
```

Fig: 7.1 Automation Of Aws

TEST AUTOMATION

Automated testing will be a huge boon as it allows developers to run the test suite right after the implementation of a new feature. Automation testing ensures that testing happens continuously and gives faster feedback, quick release turnaround without consuming more effort compared to manual testing. Automating the testing process will not only optimize effort, schedule and budget but also deliver higher quality products since it provides additional guaranteed test coverage without the risk of human error.



```
11-03-2021_12:22:10], 0.107, GET, 3.18.128.248, 5050,
11-03-2021_12:22:10], 0.011, GET, 3.18.128.248, 5050,
11-03-2021_12:22:10], 0.010, GET, 3.18.128.248, 5050,
11-03-2021_12:22:10], 0.018, GET, 3.18.128.248, 5050,
11-03-2021_12:22:10], 0.014, GET, 3.18.128.248, 5050,
```

Fig: 8.1 Testing Automation Script

2. CONCLUSION

The Doorstep Groceries App will be very helpful during any pandemic situation where people may face difficulties to buy daily groceries. Additionally, it can improve the way of life everyday even during normal times. The next generation of SaaS is stateless architecture. The automation consolidates task and process automation into a single solution for more efficient application deployment and without buying from AWS it is a best cost saving mechanism. As the future enhancement to the app the customer can choose different products from different stores and can be placed in a single order.

ACKNOWLEDGEMENT

Advice and support given by Mr. Vinayak Arumugam, Software Architect, is indeed a great help in the successful completion of this work. Our special thanks to Jasmine ecommerce Solution Inc., for providing all the support.

3. REFERENCES

[1] Bauer, E. “Cloud automation and economic efficiency”. IEEE Cloud Comput. 2018.
[2] Gareth Dwyer, “Stateless vs Stateful Architecture: Why Stateless Won?” 2020 https://virtasant.com/blog/stateful-vs-stateless-architecture-why-stateless-won

[3] Hugh McKee, “Cloud Native Applications: Stateful or Stateless Services?” 2020 https://thenewstack.io/cloud-native-applications-stateless-or-stateful-services/
[4] IBM Services, “RTO (Recovery Time Objective)” 2019 https://www.ibm.com/services/business-continuity/rto
[5] IBM Services, “RPO (Recovery Point Objective)” 2019 https://www.ibm.com/services/business-continuity/rpo
[6] Jeff Amerine, “Why the next generation of SaaS is ‘stateless’!” 2020 https://www.nuverge.com/blog/stateless
[7] React native documentation 2020 https://reactnative.dev/docs/getting-started

[8] R. T. Fielding, R. N. Taylor, J. R. Erenkrantz,
M. M. Gorlick, J. Whitehead, R. Khare, and P. Oreizy Reflections on the REST Architectural Style and "Principled Design of the Modern Web Architecture" 2017.

[9] Pahwa, P., Taneja, S. and Jain, S. "Design of a Multidimensional Model Using Object Oriented Features in UML", IARS' International Research Journal. Vic. Australia, 1(1) 2011. doi: 10.51611/iars.irj.v1i1.2011.6.

[10] Simeen Sheikh, Ganesan Suganya and Premalatha Mariappan, "Automated Resource Management on AWS Cloud Platform," IEEE Trans. on Software Engineering, pp. 133–147, 2019.