**Assignment2 Group B**

Author: Jitendra Yadav

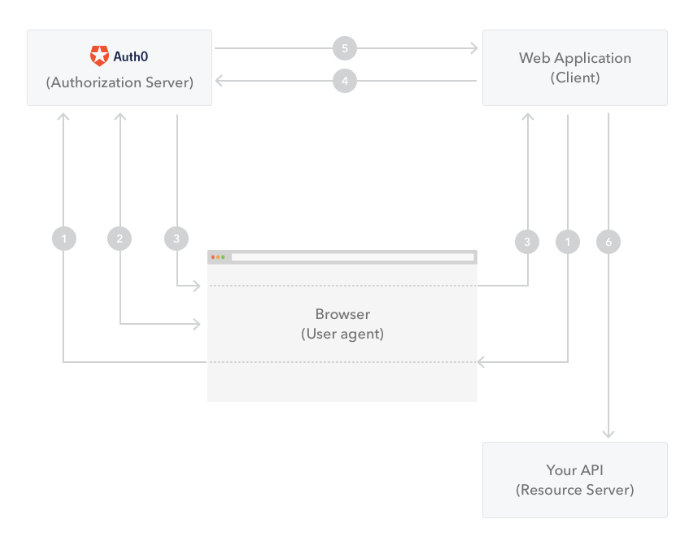
Individual contribution details for design and development of ShipNFix application.

**ShipNFix application core components& ecosystem**

* authentication-as-a-service Auth0
* NSQ real-time distributed messaging
* DynamoDB
* Golang web application
* Amazon ECS
* Docker
* Docker repo in AWS
* OpsWorks
* NSQ monitoring web interface
* ShipNFix billing API exposed using API gateway (will be accessible with final assignment submission)
* ELB (currently disabled)
* AWS region – US-EAST-1

**Authentication as a service implementation in ShipNFix application**

**ShipNFix**



**Application Architecture**

For assignment2 **ShipNFix** billing, Amazon kinesis & API gateway features are not available for testing, these features can be tested with final assignment

AWS cloud

ECS image repo

ECS cluster

NSQ cluster

Auth0

Browser

ShipNFix application

DynamoDB

Amazon API gateway

ShipNFix billing API

DynamoDB

AWS lambda

(API service for billing)

Amazon Kinesis

ShipNFix monitoring services

S3Buckets

**NSQ real-time distributed messaging system in application**

Consumer

ShipNFix topic

All tasks channel

Consumer

Queued tasks channel

Consumer

User Channel

ShipNFix uses NSQ real-time distributed messaging for implementing the business logic need by ShipNFix application developed in Golnag programming language.

* Business logic which uses NSQ is publishing a task
* Setting tasks in the queue
* Accepting tasks from the queue by the user who are subscribed to ShipNFix and want to earn money using ShipNFix platform.
* Starting/completing the task

Public url for **ShipNFix** application

<https://shipnfix.space>

<http://shipnfix.space>

(Non-SSL url would not be available as soon as SSL certs are provisioned)

Testing steps for ShipNFix application

* Login to **ShipNFix**
* Publish task and verify that task published is available in list of task
* Accept task
* Go to my tasks link
* Start task from list of task available in my tasks and the status of task should change
* Complete the task and status of task should change to completed
* All these information is captured in NSQ and DynamoDB and billing feature would be available in final assignment where publisher and task and doer of task can see their money spent and money earner based on the tasks published or completed by them.

**References**

**[Create and Manage a Nonrelational Database with Amazon DynamoDB](https://nsq.io/)**

**[Real-time distributed messaging platform with NSQ](https://nsq.io/)**

**[B](https://aws.amazon.com/getting-started/projects/build-log-analytics-solution/?trk=gs_card)****[uild a Log Analytics Solution](https://aws.amazon.com/getting-started/projects/build-log-analytics-solution/?trk=gs_card)**

[**Launch a LAMP stack web app**](https://aws.amazon.com/getting-started/projects/launch-lamp-web-app/?trk=gs_card)

[**Single sign-on implementation with Auth0**](https://auth0.com/)

**[A Go web development eco-system, designed to make the life of a Go web developer easier](https://aws.amazon.com/getting-started/projects/deploy-elastic-hpc-cluster/?trk=gs_card" \l "aws-element-modal-11b13988-1da8-45da-b546-32ff300c1706)**

**[Deploying an elastic HPC cluster](https://aws.amazon.com/getting-started/projects/deploy-elastic-hpc-cluster/?trk=gs_card" \l "aws-element-modal-11b13988-1da8-45da-b546-32ff300c1706)**