Contents

[**Objective:** 1](#_Toc32420101)

[**Project description:** 1](#_Toc32420102)

[**Smart contracts:** 1](#_Toc32420103)

[**Implementation:** 1](#_Toc32420104)

[**Git Hub Repo:** 2](#_Toc32420105)

[**Trello Board:** 2](#_Toc32420106)

[**Technology stack:** 4](#_Toc32420107)

[**Azure DevOps CI/CD** 4](#_Toc32420108)

[**Asset Management** 10](#_Toc32420109)

[**Employer Screen** 15](#_Toc32420110)

**Objective:**

This document defines the transfer of all relevant information and artefacts produced during SIT764. With this document, a new member should be able to identify all key aspects and artefacts of the project and have access to key systems or configurations.

**Project description:**

Blockchain is a decentralized system that exists between all permitted parties. Here, it is not needed to pay intermediaries and it saves time and conflict**.** In the previous trimester, we created the employee contract which holds the functionality of creating an account contract which holds the superfunds/money for the user as an employee. Blockchain records transactions on distributed ledgers accessible to all. In this trimester, we created the employer contract which has the functionality of adding or deleting an employee account, make payment for the employee by setting percentage and timer for payouts for employees and also view the list of employees linked with the employer contract. Moreover, the employer can check funds in the employer account. The contract and the system should comply with applicable regulatory norms. A single Signon is used for both the identities, employer and the employee account, where new attributes are used to identify the different contracts.

**Smart contracts:** Smart contracts allow the performance of credible transactions without the need of third parties. It is a computer protocol intended to digitally facilitate, verify, or enforce the negotiation or performance of a contract. We will use contributions of employer in our smart contract system because most of the assets are digital assets.

**Implementation:** Our developers took these factors into consideration, implementation, segregation of duties and authorization on transactions that are completed in our smart contract.

|  |  |  |
| --- | --- | --- |
| **Feature** | **State** | **Notes** |
| Account creation | Completed | Enables user to create an employer smart contract |
| Single Login | Completed | Provides an interface to help users to login either as an employer or an employee to the designed application. |
| Dashboard | Completed | Helps users to navigate through the features of the application |
| My Profile | Completed | A page which displays the list of the employee accounts, deposits, account summary etc. |

## **Git Hub Repo:**

We have used Hyper Ledger for implementing block-chain for creating a smart contract using solidity and React JS for designing the frontend. The source code could be accessed by the following link:

## **Trello Board:**

A screenshot of a newspaper

Description automatically generated

**List of what was Delivered:**

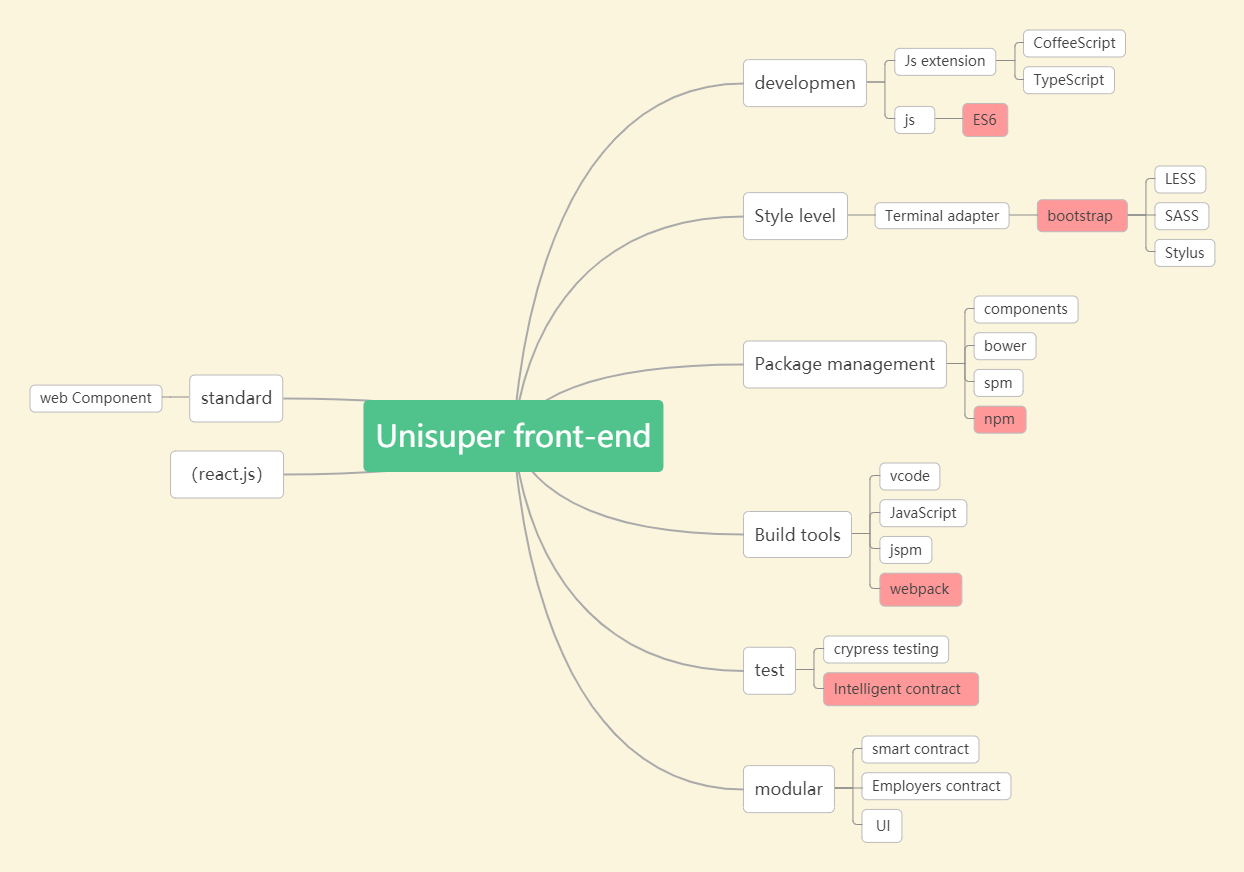
* Employer screen with employer button functionality.
* Research on band Protocol and Implementation of it.
* Assets Management screen with invest (Buy and Sell) functionality.
* Completion of UI/UX design.

**Lessons Learned:**

* Attending Daily Sprint meeting was important because it align the team together and we have focused more towards to goal of project.
* Asking for help from teammates and supervisor when we stuck in task, helps to resolve the issues much faster.
* For the Group task everyone should have come forward to volunteer the task. It should not be like if one of the teammates completed the last group task then it became its job to complete it every time.
* Changing the Role in teams gives the opportunity for everyone to work on different scopes but sometimes this opportunity can become problem to complete the task because changing the role in every Iteration means understanding and upskilling before working on any part of the project, As all the technology were new to us.

**Tech Lesson Learnt:**

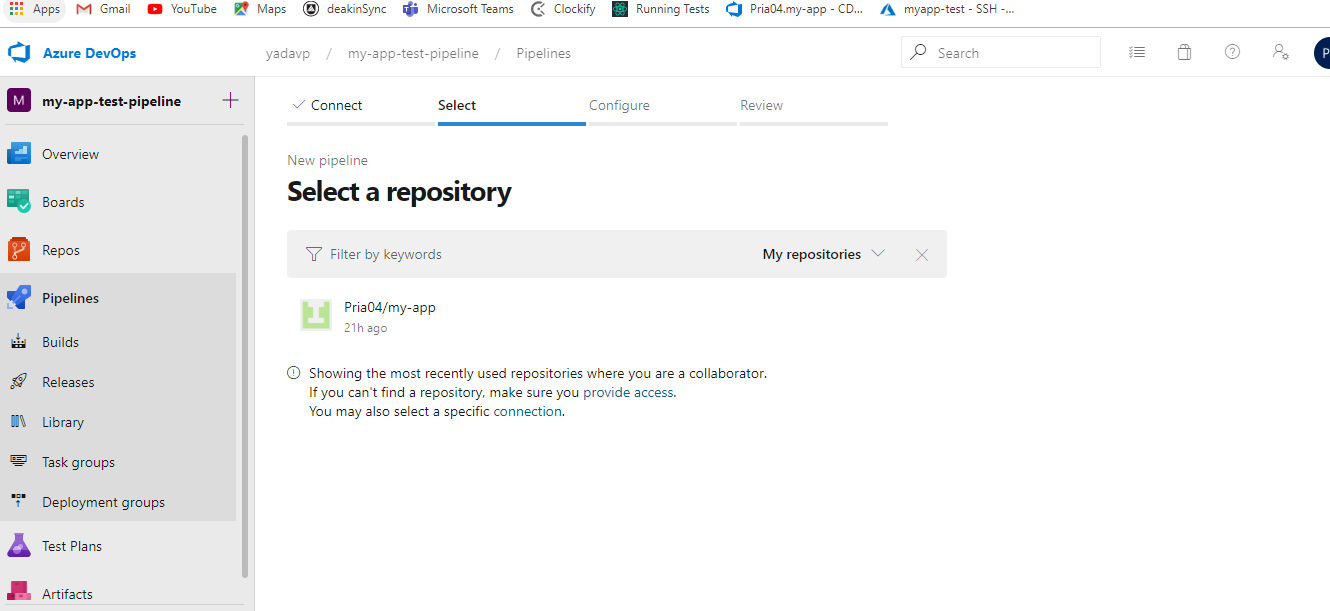
* Learning the Important technology like Azure DevOps, Solidity and React JS was the great opportunity we got in this project.
* Communication between people working on related technologies is vital. A team member working on color scheme or wireframes needs to communicate with the front-end developers. A member working on database schema, transactions need to communicate with members working on the back- end.



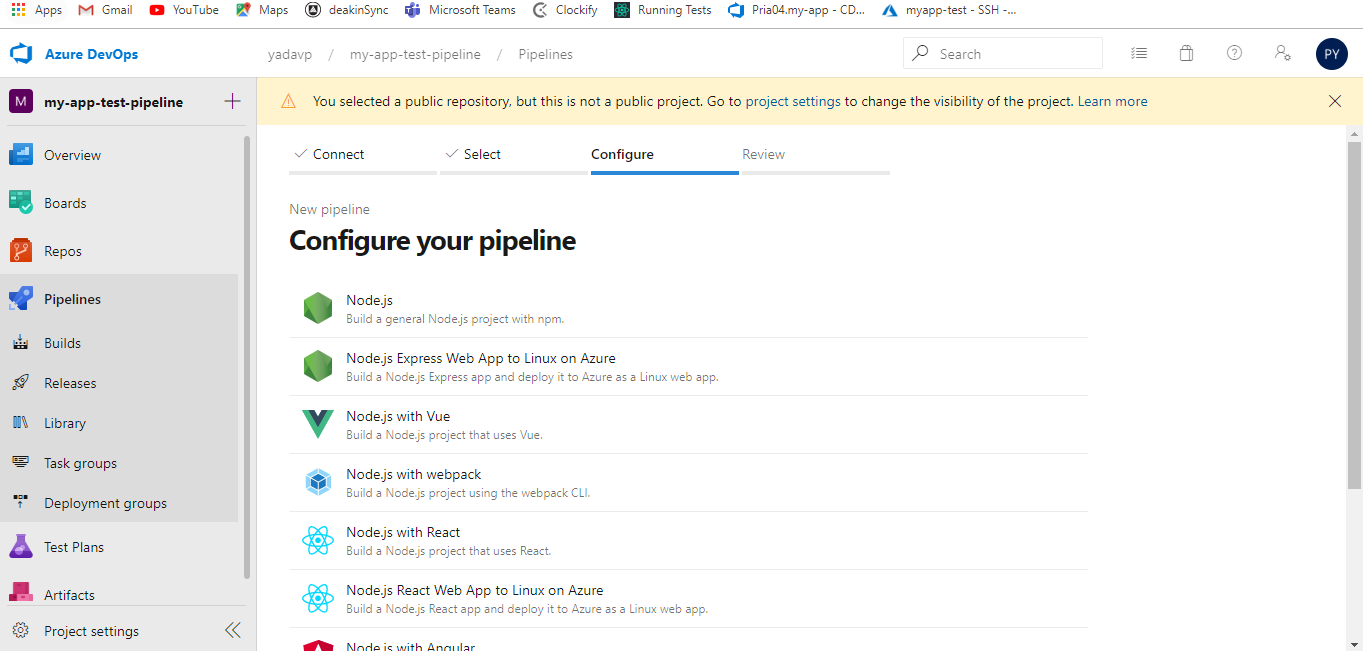
## **Technology stack:**

### **Azure DevOps CI/CD**

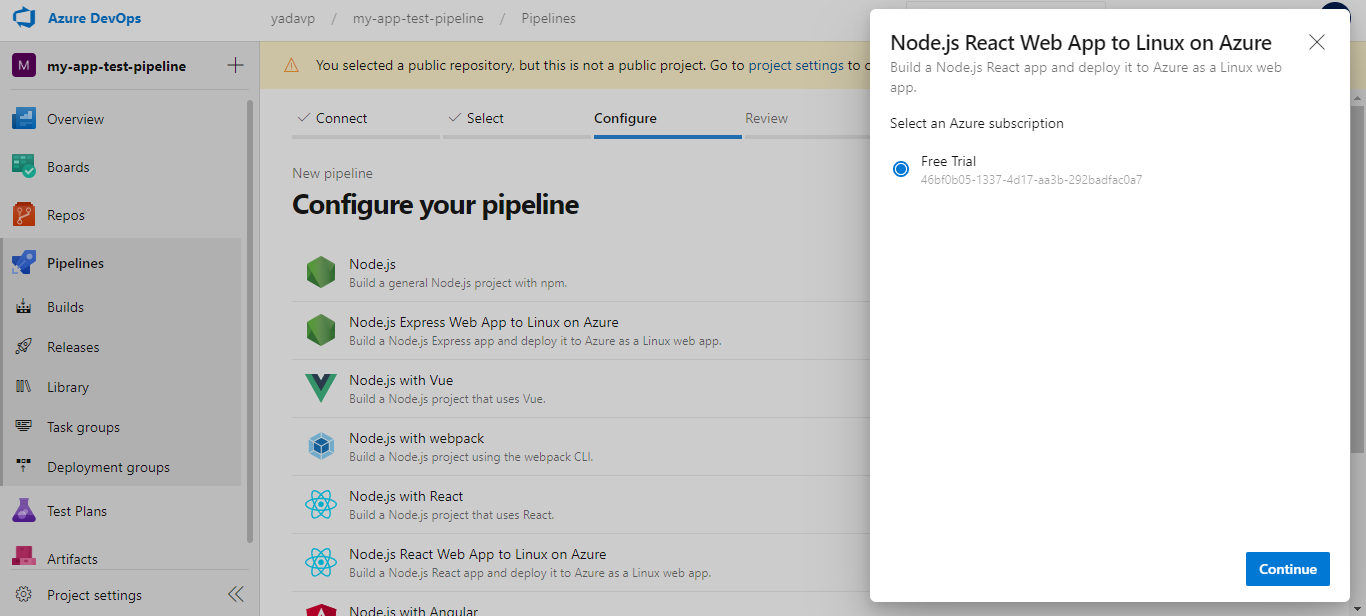
* In our project we needed the cloud environment and Azure DevOps provide fully featured Continuous Integration/ Continuous Delivery Services where we Build, Test and Deploy our code on Azure pipelines.
* Following are the simples steps of creating Azure CI part, in this we deployed a React app using GitHub and then creates a free account on Azure DevOps Pipelines.
* When you click on GitHub, you will see all the private and public repositories of your GitHub, As of now my repository is already linked with the azure so its directly showing my repo.



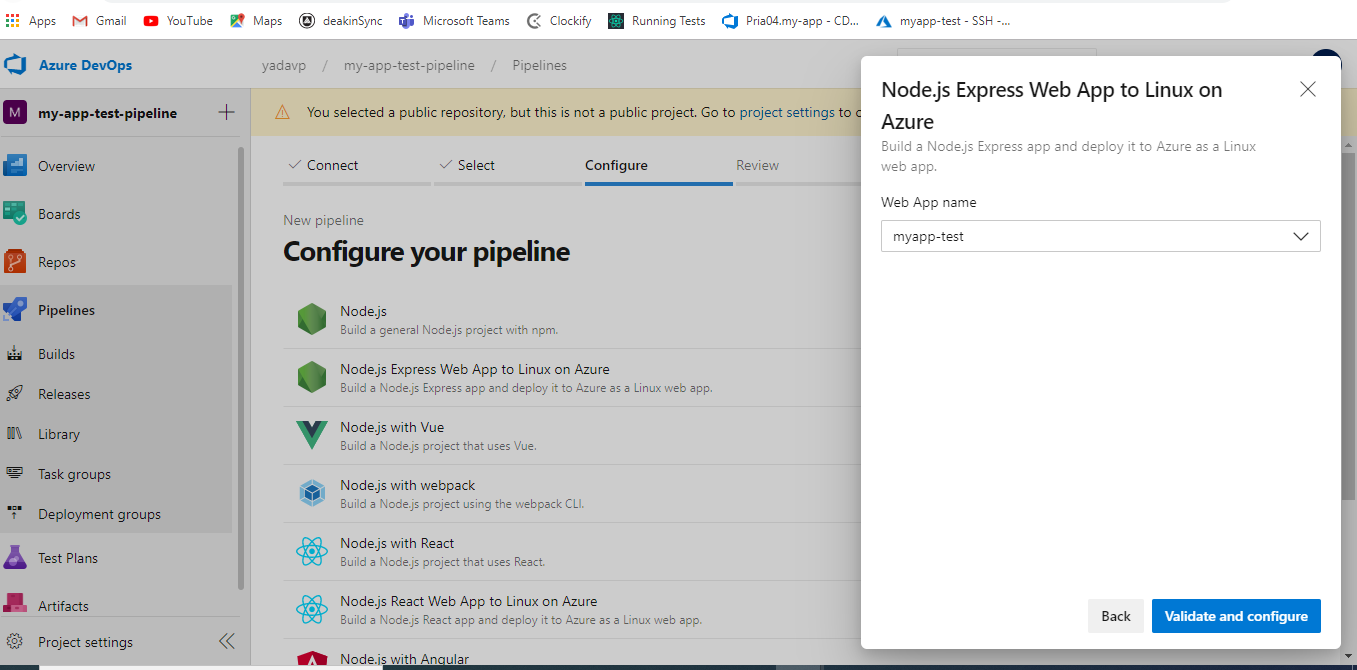
* It will ask for the template and for this project, select **‘Node.js React Web App to Linux on Azure’**



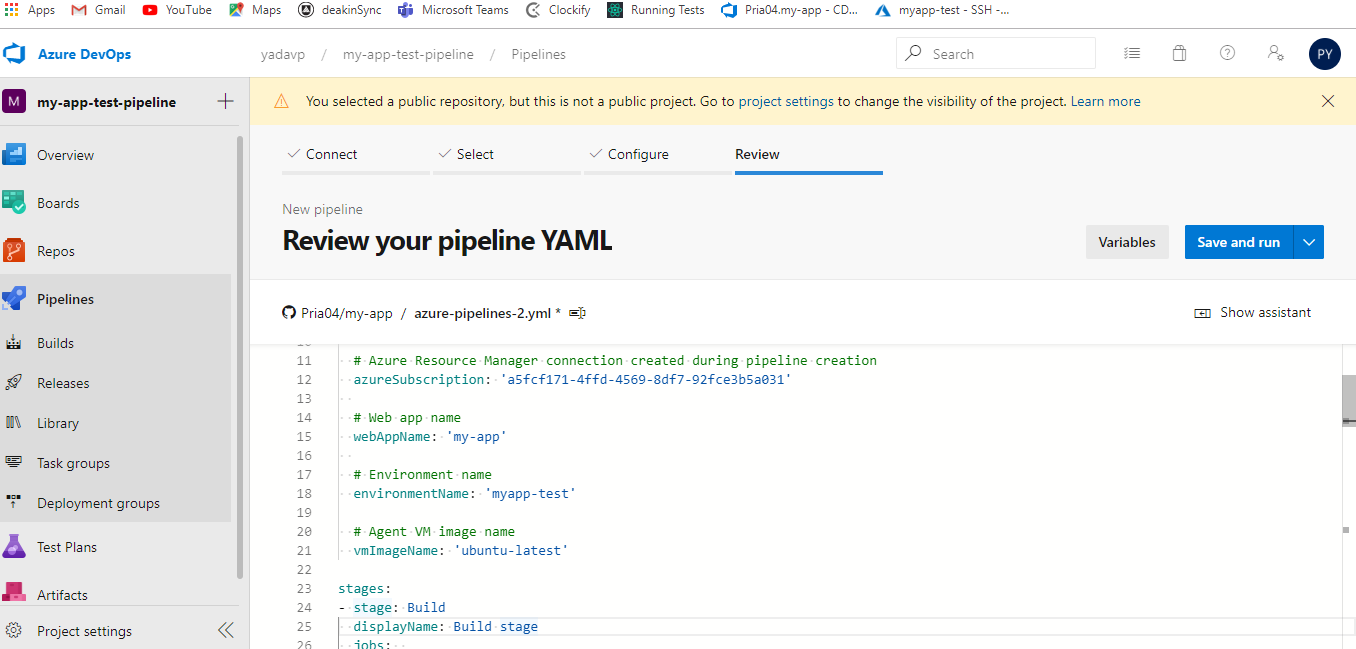
* Select **‘Free Trail’**



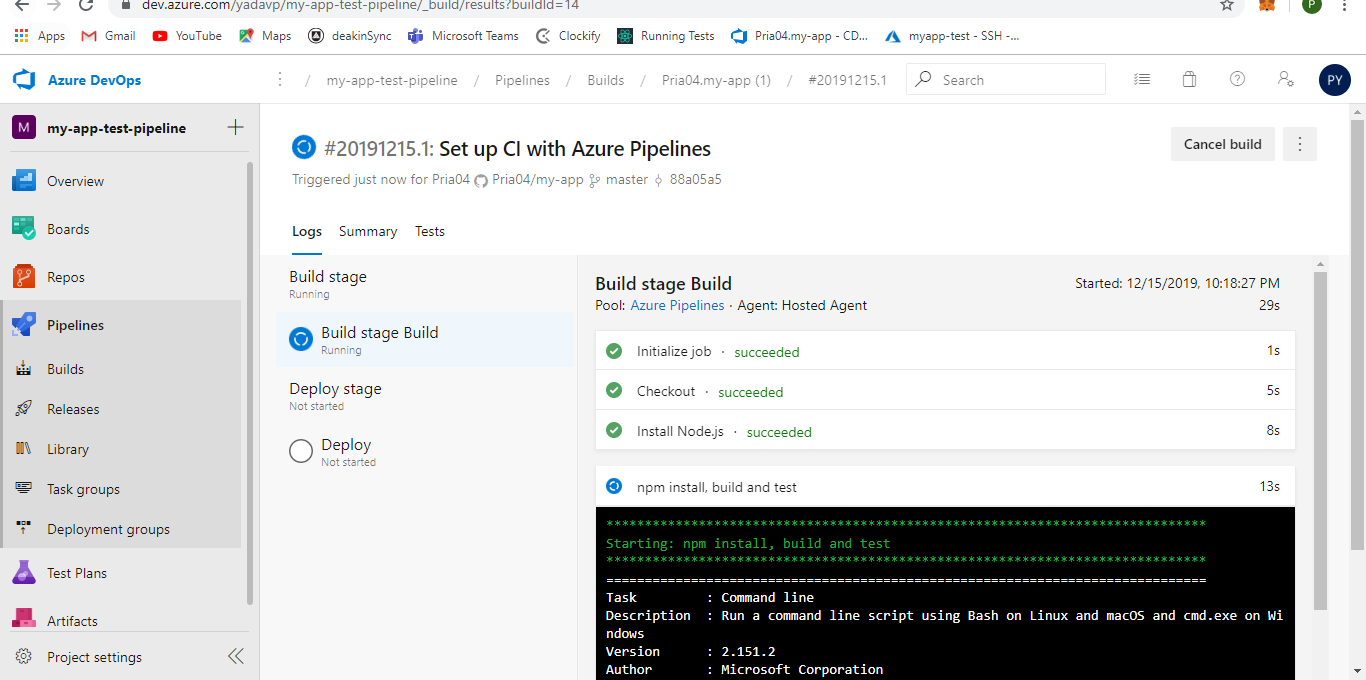
* Write the Web App Name



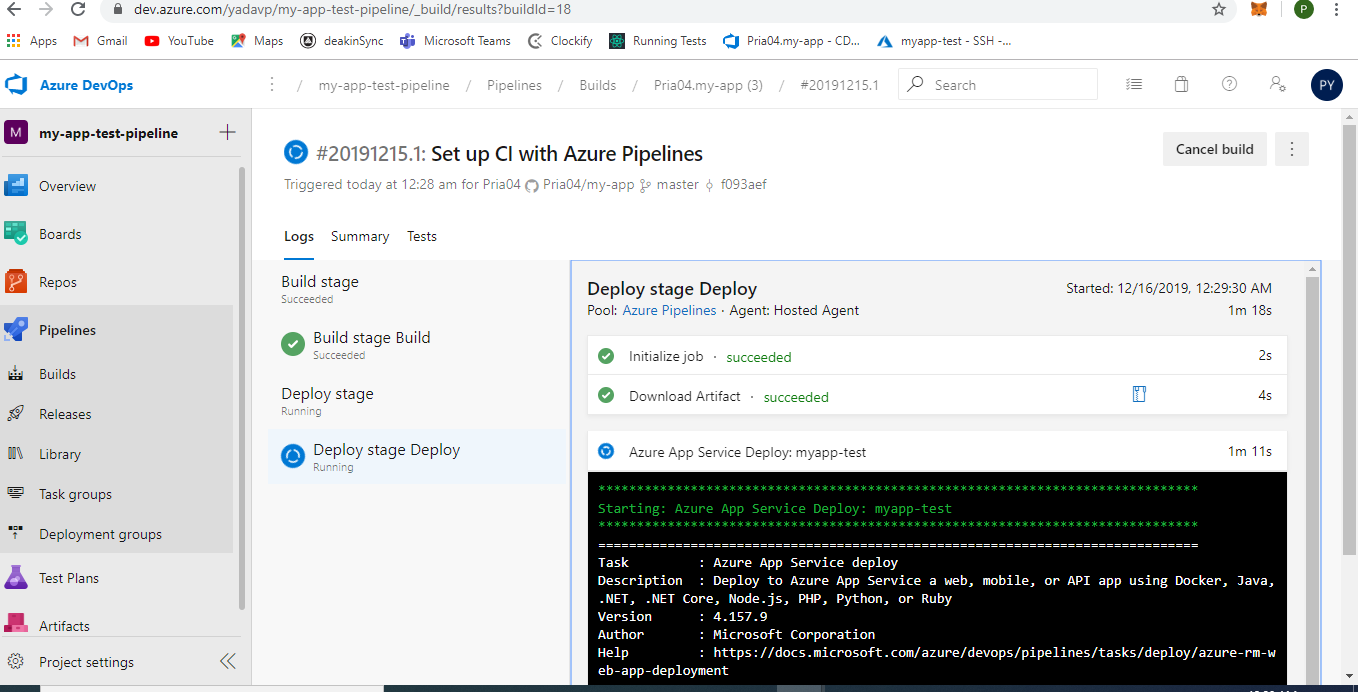
* Azure will create a Yaml file according to the technology used in Repo and then click on **‘Save and run’**



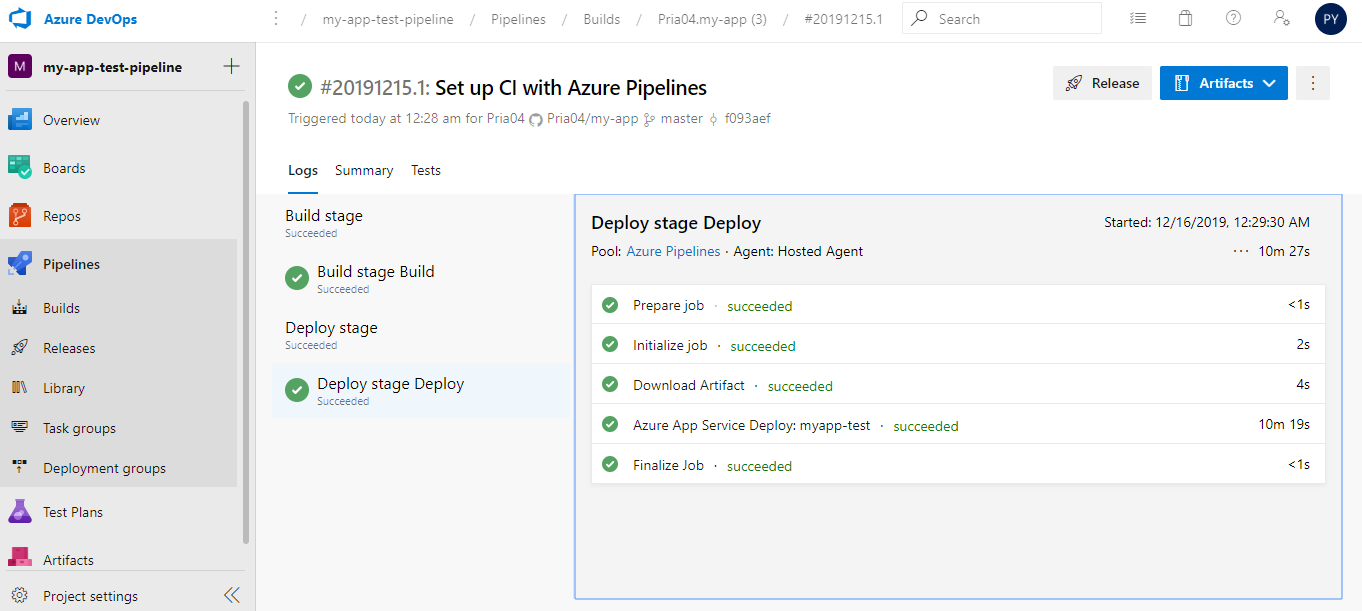
* Now it will run the Yaml code and download the code from GitHub and run the npm.



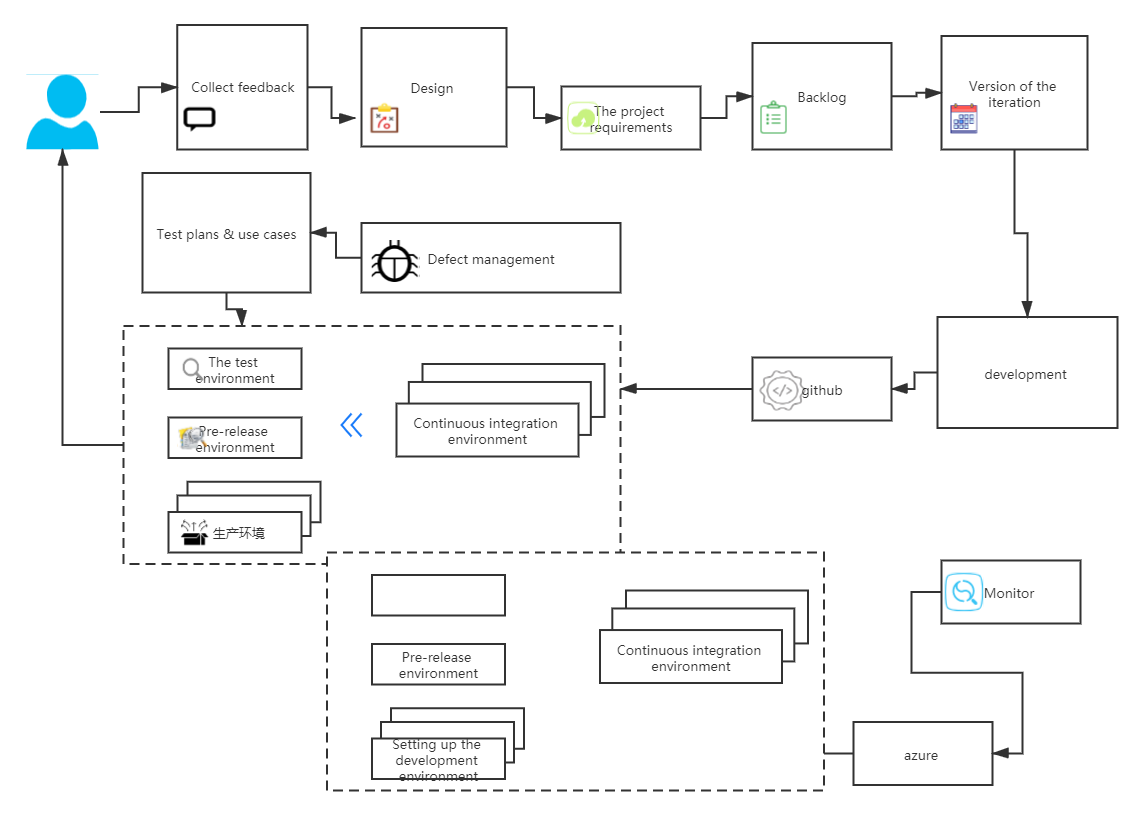
* Now Build Stage is completed, and it moved to Deploy Stage.

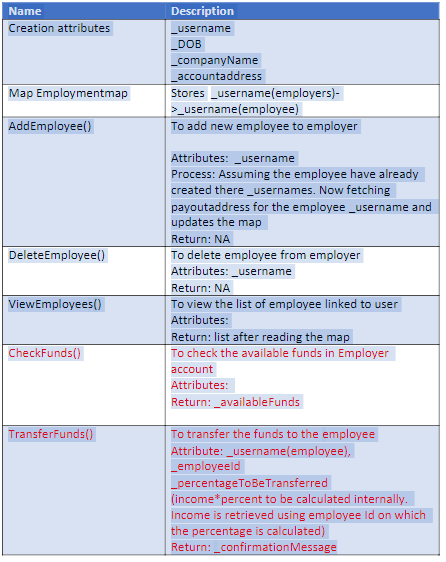


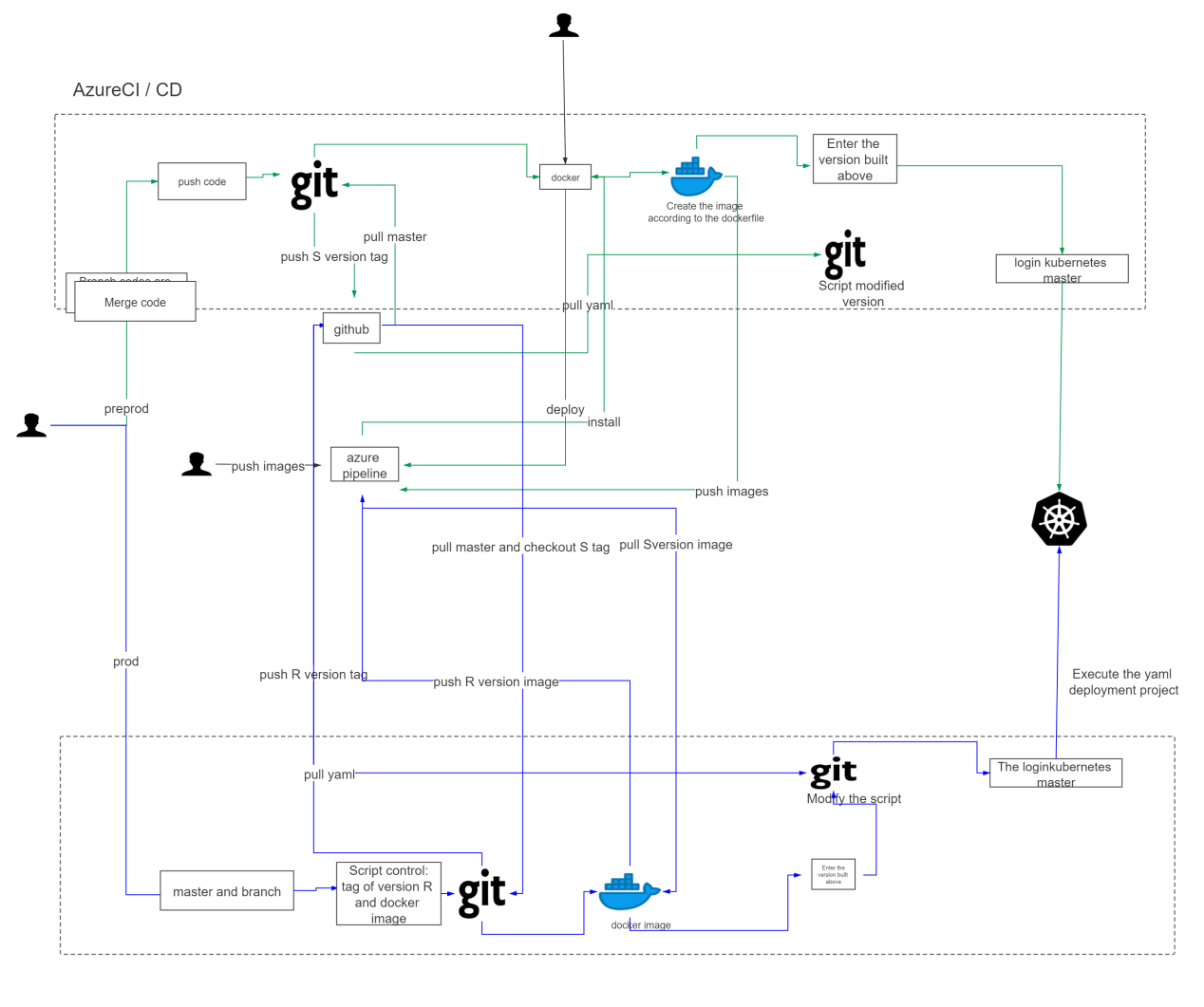
* At the End Deployment if there is no error that means deployment is successful.



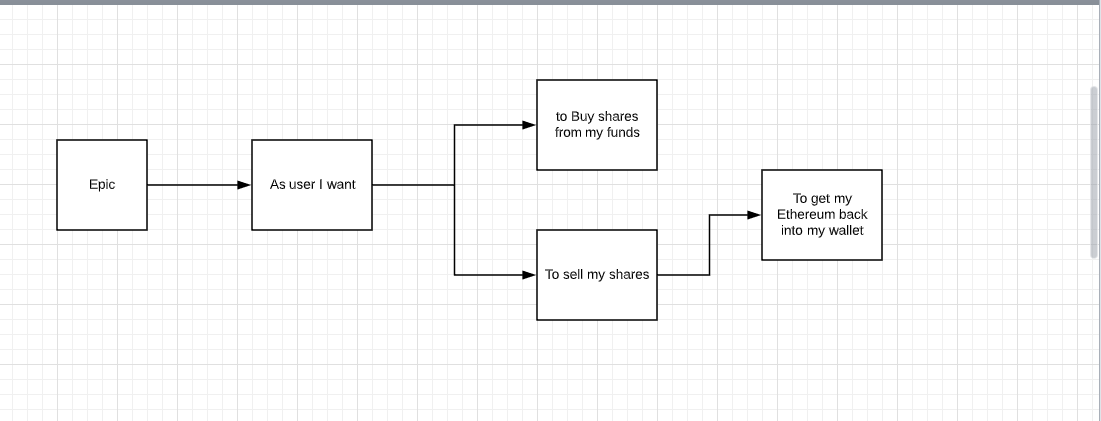
The operation and maintenance team are responsible for using azure's DevOps service to push our project to the cloud server to run. The technologies involved include azure vm, docker, monitoring, Kubernetes, elk architecture.







### **Asset Management**

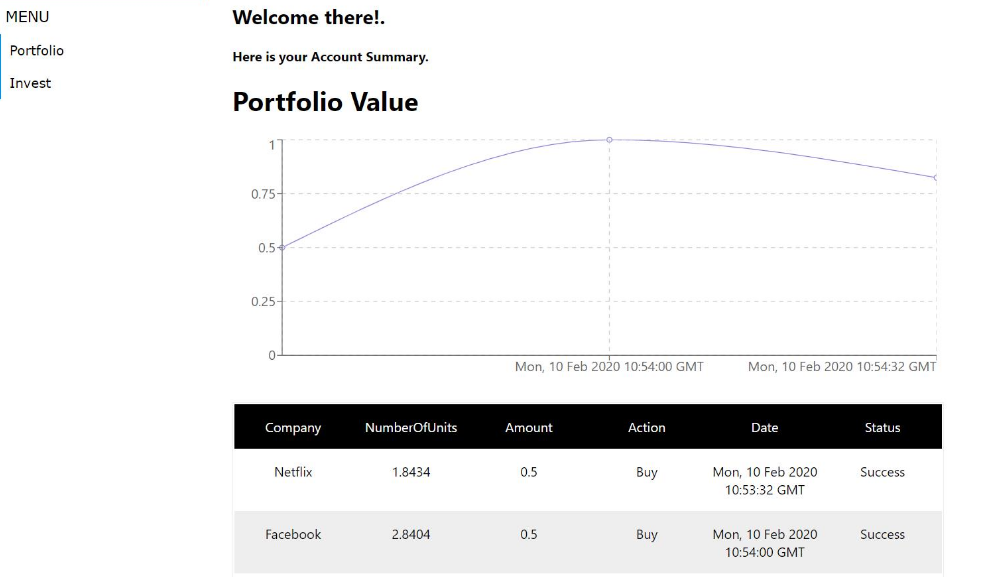


**Band Protocol**

our system tried simulating SMSF on blockchain technology. We have two smart contracts on blockchain where one is manager account and the other is investment account. There will be a single manager account for all users and one investment account for each user. Now that, in our system, we implemented the functionality, where one can buy and sell assets. For this to do, we need the price of the share we want to invest in.

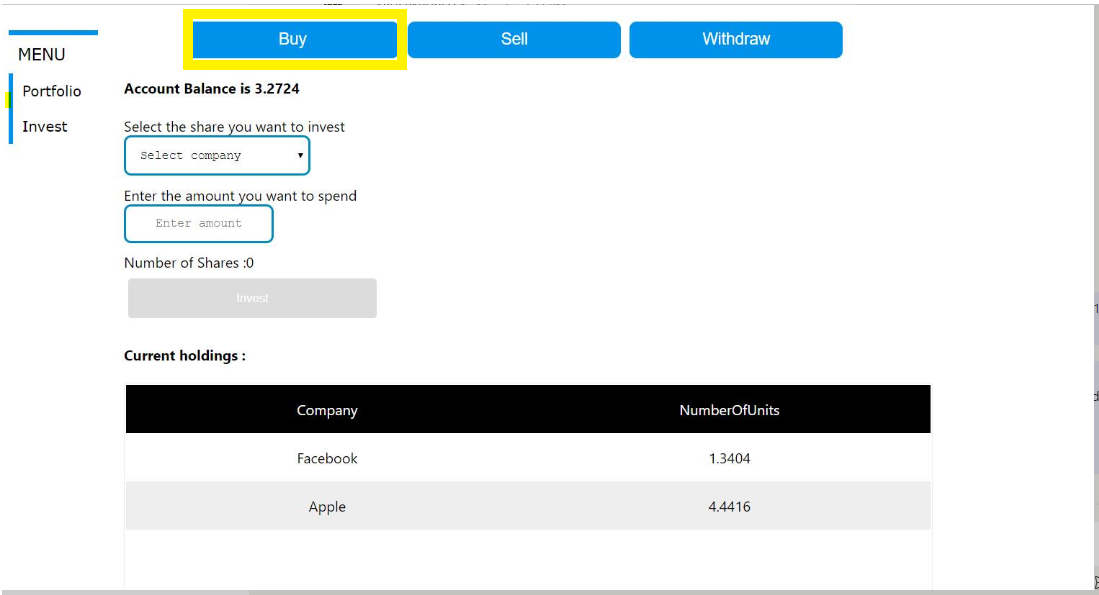
For getting the share price, we cannot directly get the data from any websites as the system is built on blockchain. So, we have used band protocol which connects smart contracts with trusted off chain data providers. For this band protocol to work, there is a band library which needs to be deployed on kovan test net. Kovan test network is a publicly accessible blockchain for Ethereum. As of now it has got only the test network, Ethereum consortium is working on building the main network. This protocol provides a financial data set that has quite a few share prices. For every transaction on kovan network will be through infura(API that provides access over HTTP, websockets to Ethereum networks) that allows only signed transactions to connect to kovan. For this, we have used metamask signature. Using band protocol, we were able to fetch the share price of a company.

**Portfolio Screen:**

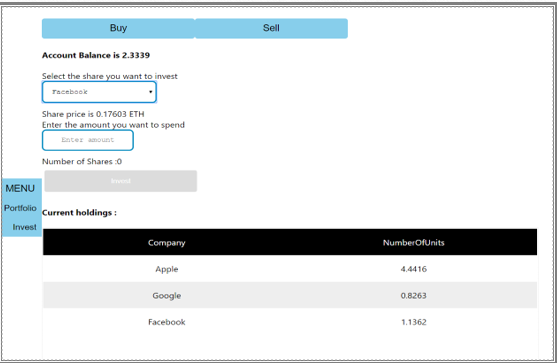


* This is the Portfolio Screen of Asset management where employee can invest their shares by clicking on Invest Button.
* And this portfolio screen that displays the chart of net value of total assets to date. It can show the transaction history.
* If any transaction Success/fails, then it stored in the transaction history.

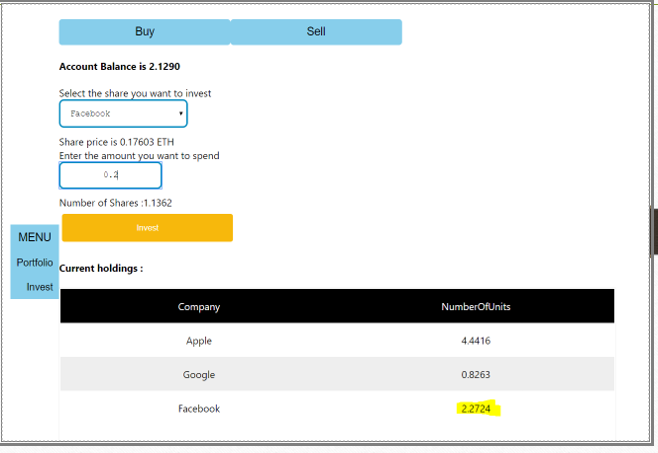
**BUY:**



* In the Invest Screen you can see the Account balance of your wallet.
* To Buy shares employee have to click on Buy Button as shown in the above screen.

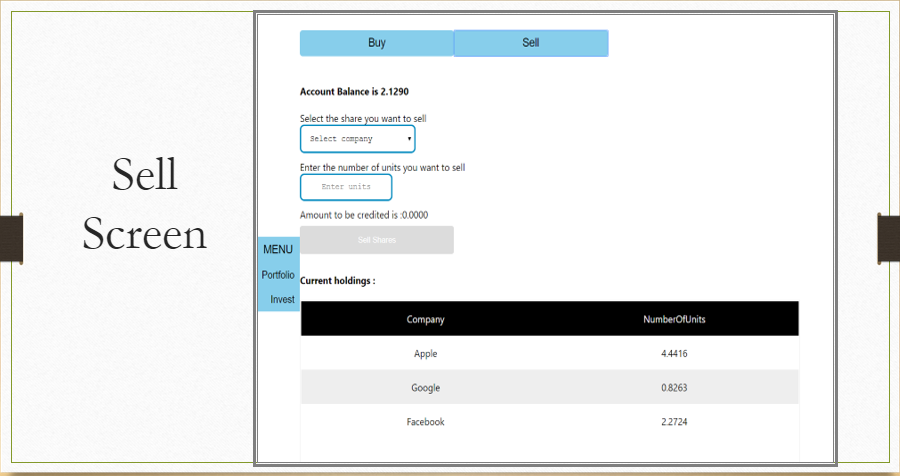


* In the Buy screen you can invest you shares in the companies like Facebook, google etc. and you can see the share prices of that company also.
* In the current holding section, you can see the investments that are already done.
* By this we can invest in the funds.
* And there is a validation check on invest button, when you put the investment amount less then or equal to funds available in your wallet.



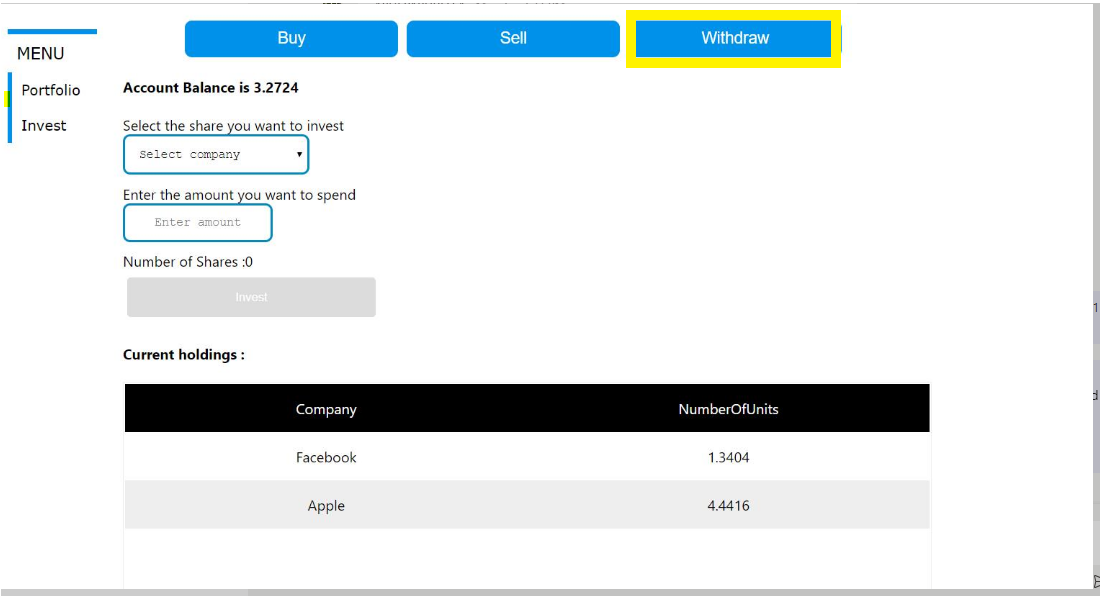
**-**Your portfolio gets updated with the latest transaction and actual holdings can be seen in the invest screen.

**Sell:**



* To sell the shares available in your account you must select the share that you have already purchased.
* And how many units of shares you want to sell and then “sell shares” button enables, and you can sell your shares.
* Once you click on sell, you can see the holdings that were deducted.
* Only after selling the shares we can use the Withdraw functionality.

**WITHDRAW:**



* In withdraw you can get back the funds available on your portfolio screen back to your wallet.
* For that you need to click on the button and enter the amount you want back from the available account fund.
* Then clicking on withdraw button, will send back the amount entered back to the superannuation fund account.

### **Employer Screen**