

Asset Onboarding Framework

END

~~MID~~TERM PS2 REPORT

PREPARED BY

Aditya Mathur (2018btechcse102)

COMPANY NAME

IBM, Bangalore

FACULTY GUIDE

Mr. Gaurav Raj

Submitted On: - 13-05-22



Department of Computer Science Engineering,
Institute of Engineering and Technology (IET) JK
Lakshmipat University Jaipur

May 2022

CERTIFICATE.....	3
ACKNOWLEDGEMENTS	4
ABSTRACT	5
CHAPTER 1: FIRST MONTH'S WORK.....	6
JAVASCRIPT :	6
REACT:.....	7
TECHNOLOGY SANDBOX:	7
GITHUB:	8
CHAPTER 2: PROJECT WORK.....	9
Module 1 Asset Onboarding Checklist:	9
The Module 2: Didact builder.....	14
Module 3 Container Image Builder:.....	16
Module 4 GitHub Action:.....	18
DOCUMENTATION AND FINISHING:.....	20
PROJECT DEMO:.....	20



July 06, 2022

IBM India Private Limited

Manyata Embassy Business Park,
G2 Block, Nagwara Outer Ring Road,
Bangalore – 560045, India.

Tel : 91-80-49139999

<http://www-07.ibm.com/in/careers/>

Dear Mathur, Aditya,

Congratulations!

We would like to take this opportunity to commend you on the successful completion of your project.

This letter certifies that you have completed your internship with IBM India Private Limited. This also confirms your employment in an intern capacity from January 08, 2022 to July 07, 2022 under the guidance of Charu Nawani.

Project title – Asset Onboarding Framework

Project Details – Onboarding Framework for Sandbox: Asset Onboarding is a project to make the process of adding an asset to technology sandbox easier and automated

Internship Serial Number: AVV9PE744

Signed By – IBM Authorized Signatory
Director – Talent Acquisition ISA

CERTIFICATE

This is to certify that the END TERM project work submitted by Aditya Mathur (2018btechcse102) towards the partial fulfilment of the requirements for the degree of Bachelor of Technology in Computer Science Engineering of JK Lakshmipat University Jaipur is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for END TERM examination.



(Santosh Kumar)

fr Mr. Gaurav Raj

Faculty supervisor

Department of Computer

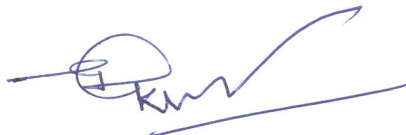
Science Engineering

Institute of Engineering &

Technology

JK Lakshmipat University

Jaipur



Dr. Gireesh Kaushik

Department Coordinator

Department of Computer

Science Engineering

Institute of Engineering &

Technology

JK Lakshmipat University

Jaipur

Date of Submission: - 13-05-2022

ACKNOWLEDGEMENTS

I would like to thank my Manager Charu Nawani, Project Managers Amol Dhondse and Manoj Jhagirdar for giving me a chance to work under their guidance and supervision. Gaurav sir for his help and supervision. My parents for motivating me. It would not have been possible without the guidance and help of all the people mentioned. A hearty thanks to all



Aditya Mathur

ABSTRACT

The report revolves around the work done by me during my 6 month internship at IBM, ISL Bangalore. It includes the project I worked on, the technologies I learned and used and also includes my experience during the time spent in the internship.

The first month included learning basics of Java Script, Node JS and Full Stack development using JS. I also learned about Privacy and Security by completing a couple courses provided and learned the Importance of Abstraction of data to protect from any threats. After the first 2 weeks I received my project and mentors, The project I am working on is the Asset onboarding framework for IBM technology Sandbox.

The project is divided into 4 modules which are, Asset Onboarding Checklist, Didact Builder, Container Image Builder, and Admin and User Dashboard.

The technologies used in every module is different from each other in some manner but the major programming language is Java Script.

The project is almost done with all modules completed, Combining the modules is the part still under progress to be done by the end of month may

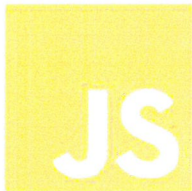
CHAPTER 1: FIRST MONTH'S WORK

The first week was basic training and introduction of the various teams at IBM with their team leaders discussing their team's role at IBM ISL and what projects and things they work on. I was also asked to do 2 courses on data privacy and importance of data security.

The second week was when I was assigned a project manager and was asked to contact them regarding further process. The manager gave us a brief about what technology will we be using in the project.

My Project Manager Amol Dhondse, asked me to get an idea of React and JavaScript as they were the main technologies and Programming Language to be used in the project.

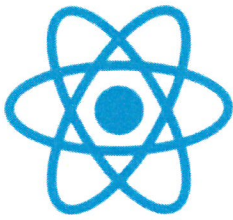
JavaScript :



JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

JavaScript was first known as **LiveScript**, but Netscape changed its name to JavaScript, possibly because of the excitement being generated by Java. JavaScript made its first appearance in Netscape 2.0 in 1995 with the name **LiveScript**. The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and other web browsers.

React:



React is a JavaScript-based UI development library. Facebook and an open-source developer community run it. Although React is a library rather than a language, it is widely used in web development. The library first appeared in May 2013 and is now one of the most commonly used frontend libraries for web

development.

React offers various extensions for entire application architectural support, such as Flux and React Native, beyond mere UI.

After the first 2 weeks I got the project, Asset Onboarding Framework for IBM's Digital Technology Sandbox. I was introduced to what a technology sandbox is and how it works. Also the details of the project and how will we go about it.

I explored around the different IBM platforms including IBM Cloud and IBM Sandbox, attended a few testing sessions for the Technology sandbox and experienced the sandbox first hand. I also started to learn about working in a team using GitHub to work on the same project and maintaining the updated versions across the team.

Technology Sandbox:



A sandbox is an isolated testing environment that enables users to run programs or open files without affecting the application, system or platform on which they run.

Software developers use sandboxes to test new programming code. Cybersecurity professionals use sandboxes to test potentially malicious software. Without sandboxing, software or applications could have potentially unlimited access to all the user data and system resources on a network.

GitHub:

GitHub is a Git repository hosting service, but it adds many of its own features. While Git is a command line tool, GitHub provides a Web-based graphical interface. It also provides access control and several



collaboration features, such as a wikis and basic task management tools for every project.

The flagship functionality of GitHub is “forking” – copying a repository from one user’s account to another. This enables you to take a project that you don’t have write access to and modify it under your own account. If you make changes you’d like to share, you can send a notification called a “pull request” to the original owner. That user can then, with a click of a button, merge the changes found in your repo with the original repo.

These three features – fork, pull request and merge – are what make GitHub so powerful.

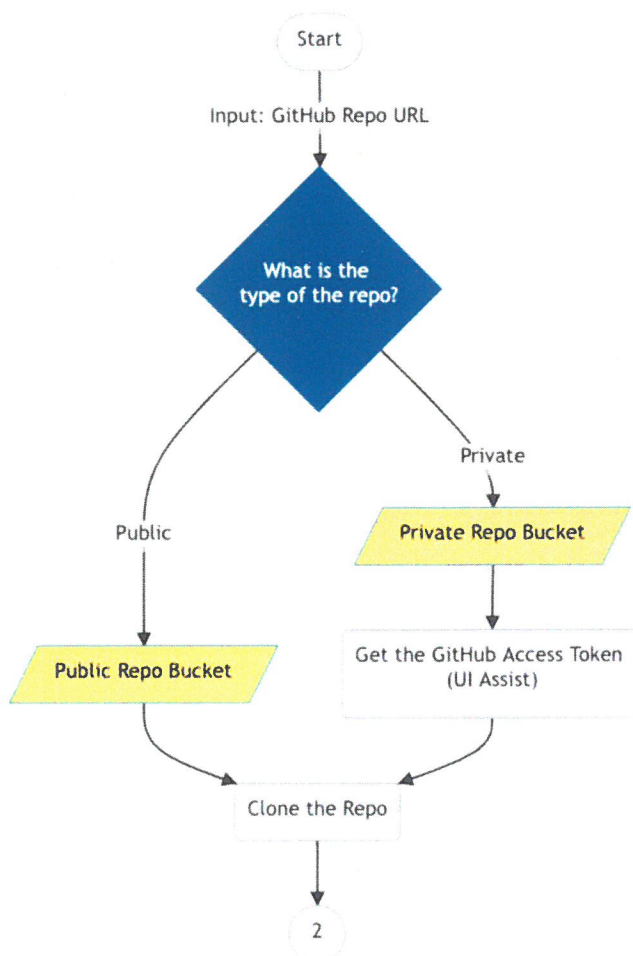
CHAPTER 2: PROJECT WORK

In the second month I started working on the actual project starting from module 1, which is the Asset Onboarding Checklist.

Module 1 Asset Onboarding Checklist:

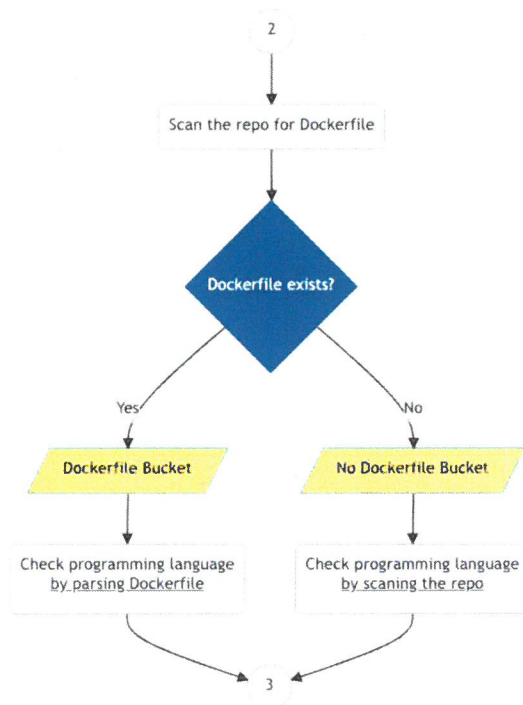
This Module required a number of APIs to check whether an asset which is already uploaded on GitHub can be onboarded or not.

I learned how to create APIs using React and Node js and starting working on the checklist



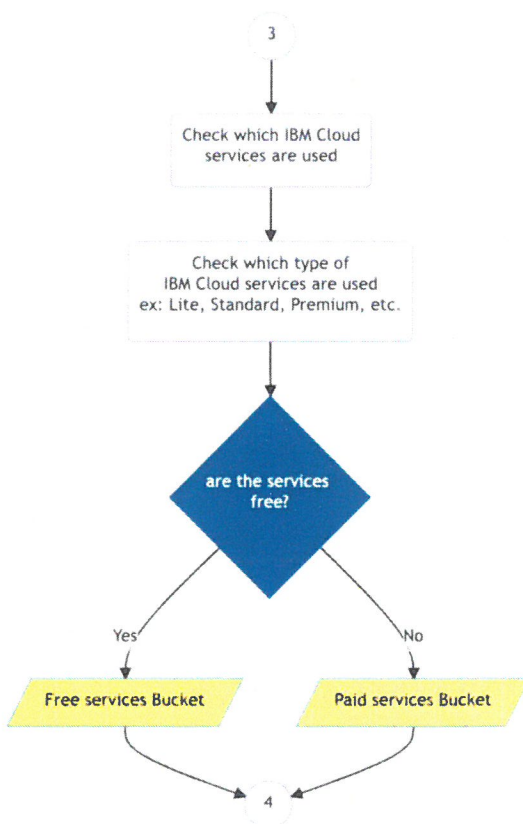
The First API was to check Visibility of the Repository. If it was a public repository we would move to the next step which is cloning but if it was a private one we would ask for a GitHub Access Token to access the Repo and move to the next step.

The next step was to clone the Repo so that further details can be checked. The cloning of the repo was done using the concept of shell script and calling them using certain APIs.

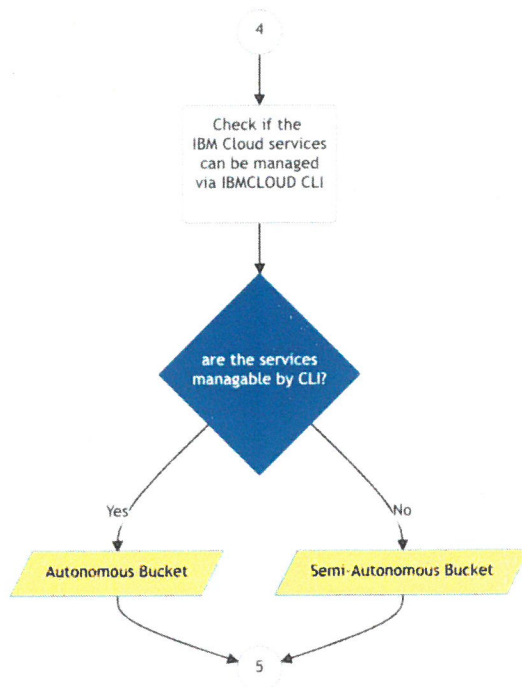


The Second API is to check existence of a Dockerfile. If it exist, parse through the Dockerfile to look for the programming language of the asset.

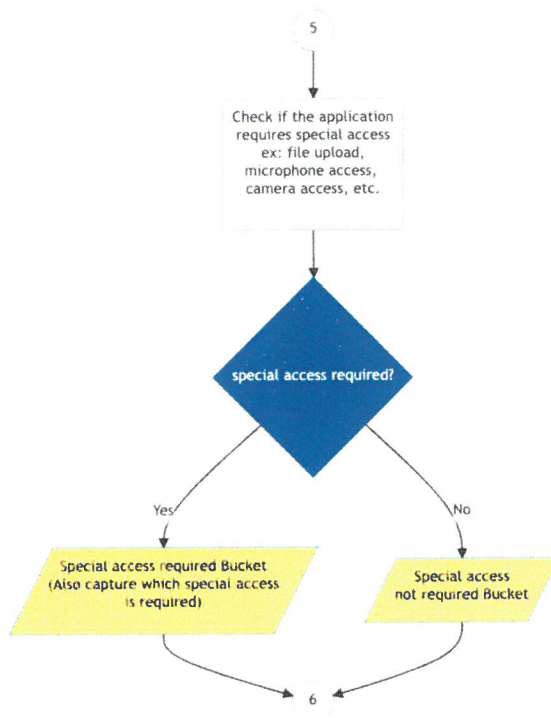
If the File Does not exist parse through the repo to look for language.



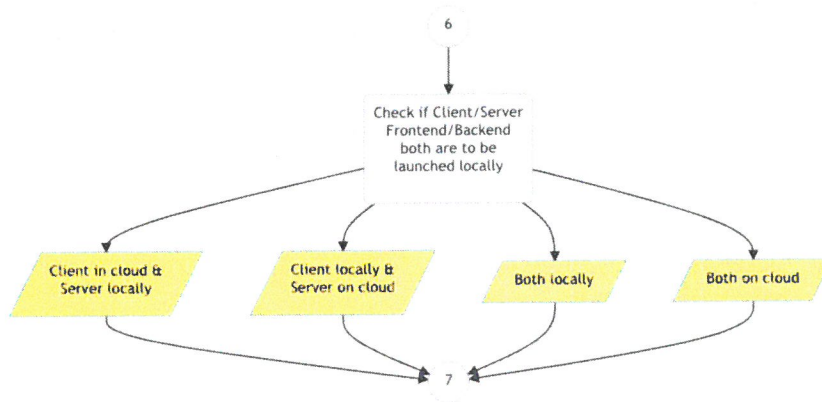
The Third API is to check what IBM Cloud services is the asset using and see if they are free or not and put them in their respective buckets.



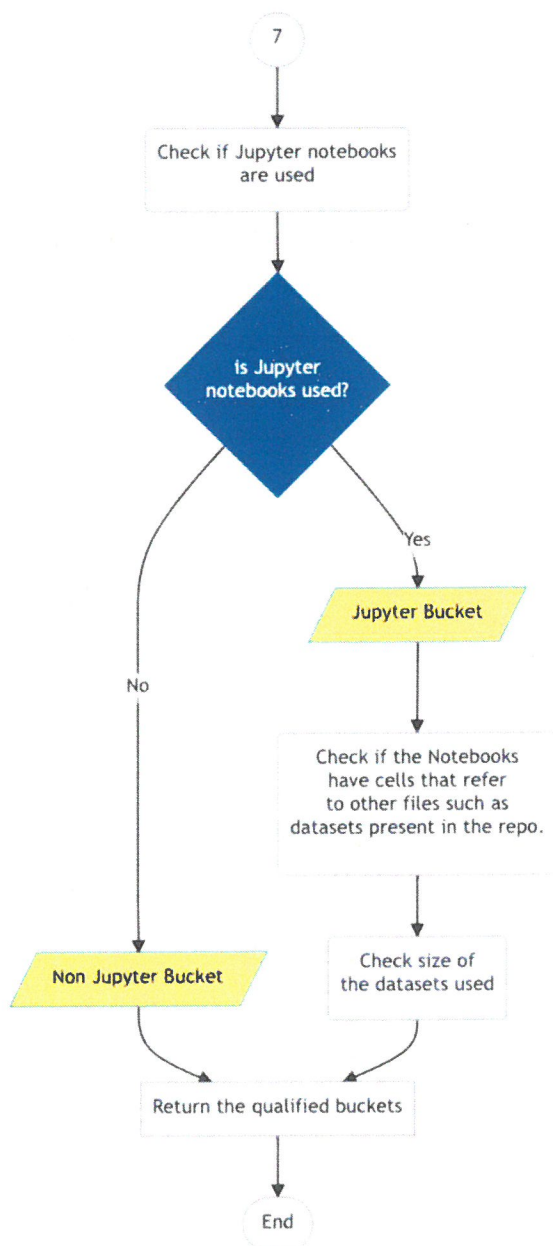
The Fourth API was merged with the 3rd API where it also looks at the services whether they can be managed using IBM CLOUD CLI or not and put in the respective bucket.



The fifth step was taken as a manual input from the user using the front end and categorised based on the user response.



Similar to the step 5, step 6 was also taken as a manual feedback from the user.



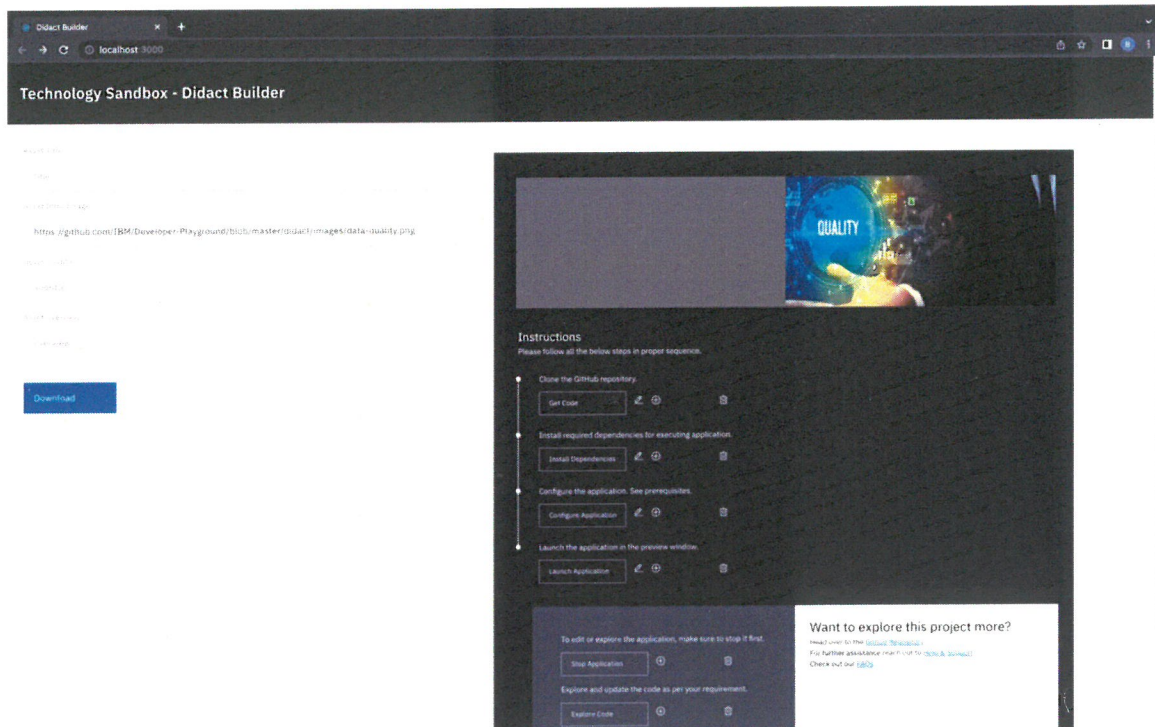
The Last API is to check if the asset uses Jupyter notebook, if it does then check if any of the notebooks have reference to any dataset and then look for the size of the dataset used. Flag if the size is over the allowed limit and classify to the qualified bucket

After creating the APIs and testing it on the local host, the APIs were pushed to the IBM Kubernetes Cluster for deployment by using a `deploy.yaml` file. The APIs after pushing were exposed on the internet and can be called from any remote desktop and used in the front end.

After the deployment of the APIs the frontend was designed and coded which took an input of GitHub URL as an input from user and called all the APIs at a call of a button, and displayed all the fetched data from the repository for user to review and also took the manual input mentioned above. And then post review send the details to module 2 which is the Didact Builder.

The Module 2: Didact builder

A Didact is an interactive readme.md file for the asset, which makes it easy and convenient for the user to run and test the asset.



The Didact Builder creates these interactive didact interface and add it to the Code Patterns section of the Sandbox. So the developer who is onboarding the asset can customize the didact by using functionalities like drag and drop for images text and add or delete commands which will be run in the Sandbox environment.

The developer can add all the commands required to prepare and run their asset, which will create individual buttons for the instructions for the users. This make testing for the user convenient as they don't need to read any detailed instructions or copy paste commands to command prompt.

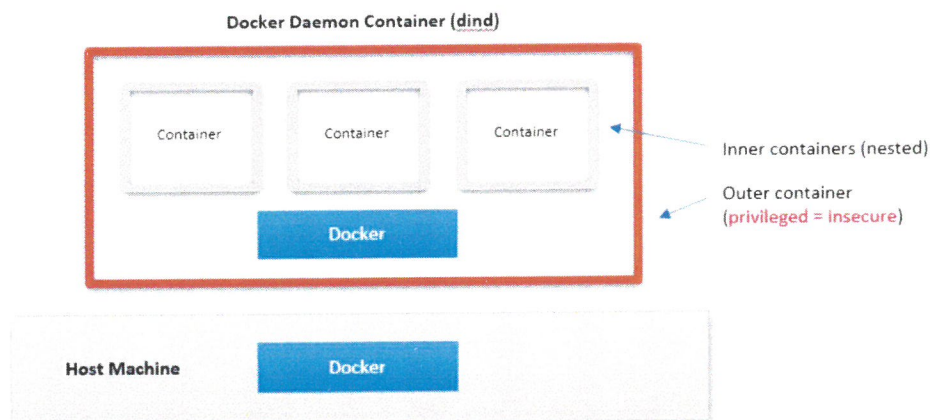
The didact builder has the basic functionality of creating CTAs which are call to actions. The user can add, delete and reorder these CTAs. Clicking on each CTA, 1 or more commands can be added into each

CTA, These will directly run on the terminal of the environment where the didact is running.

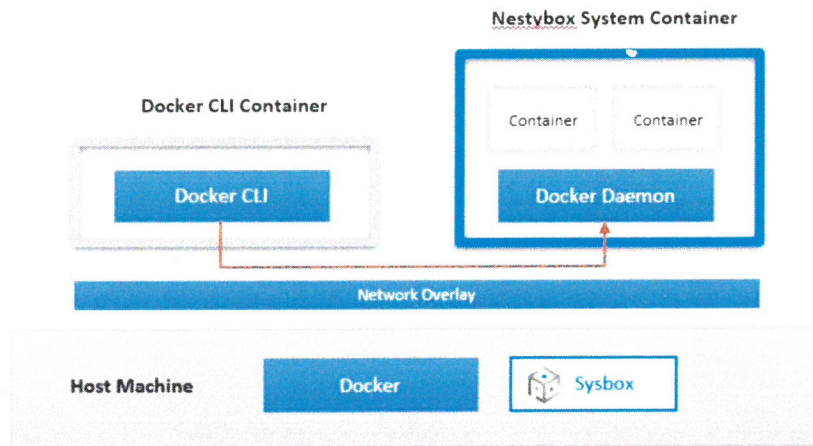
Module 3 Container Image Builder:

This module incorporates work which was mostly novel. The module clones the GitHub repository and looks for docker file, and uses that docker file to create a docker image for the user and then runs it. Now since the working of the module will take place inside IBM Technology Sandbox, which is a cloud environment, Docker daemon can't be used inside the sandbox, coz it causes 2 significant issues, the user would gain root access in the environment and also the workload for the sandbox would increase a lot since building and running docker container for large application or assets is a heavy task.

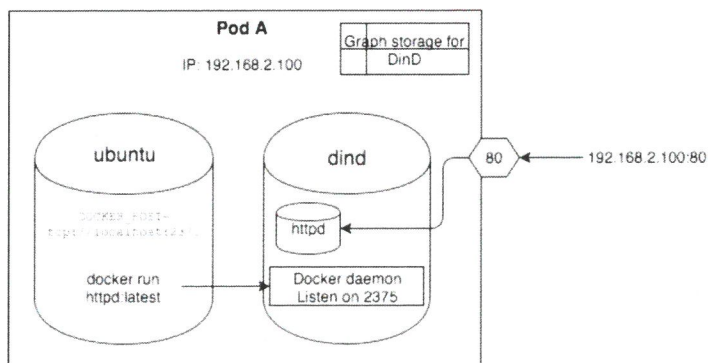
This caused a lot of issues so the approach of using DIND was thought of.



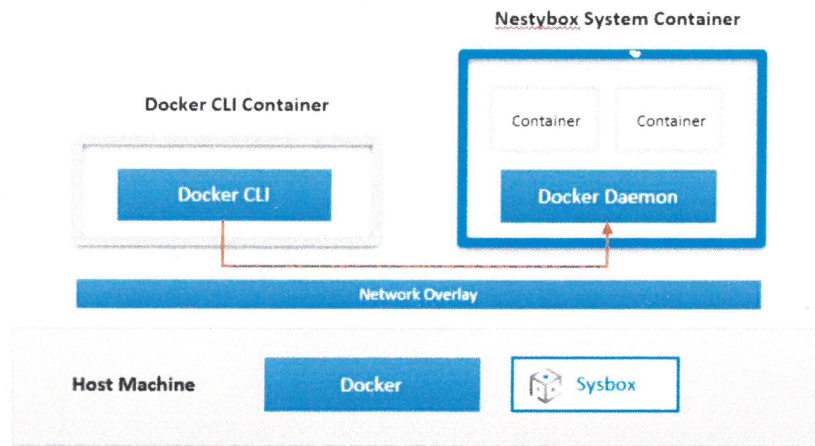
The initial attempt was to use Nestybox's Sysbox to get docker running inside a Kubernetes cluster. The approach was not feasible for the project since sysbox was not supported by IBM Cloud Kubernetes Cluster and the pods created was stuck in a pending state.



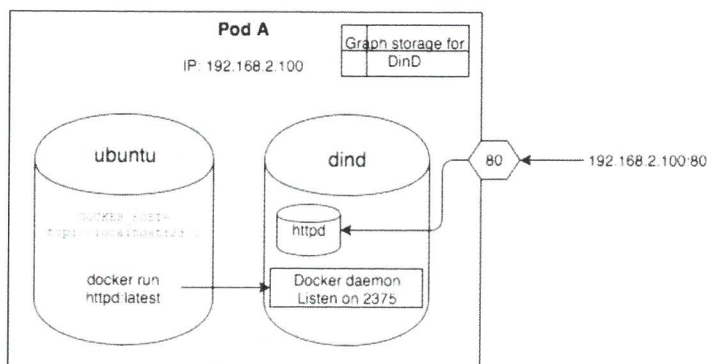
After this several other approaches were tried but none worked, so a custom POD which included IBM Cloud, IKS, Python and Docker was created. The pod was then used to connect to and use as a remote connection. This had several restrictions and complications as since it is a pod it cannot be created into a service and exposed to a IP address so instead, after connecting to the IKS using IBM Cloud login the Docker Daemon port was exposed to the local host of the Sandbox and then localhost port was used as host for running docker commands of creating and running the asset.



The initial attempt was to use Nestybox's Sysbox to get docker running inside a Kubernetes cluster. The approach was not feasible for the project since sysbox was not supported by IBM Cloud Kubernetes Cluster and the pods created was stuck in a pending state.



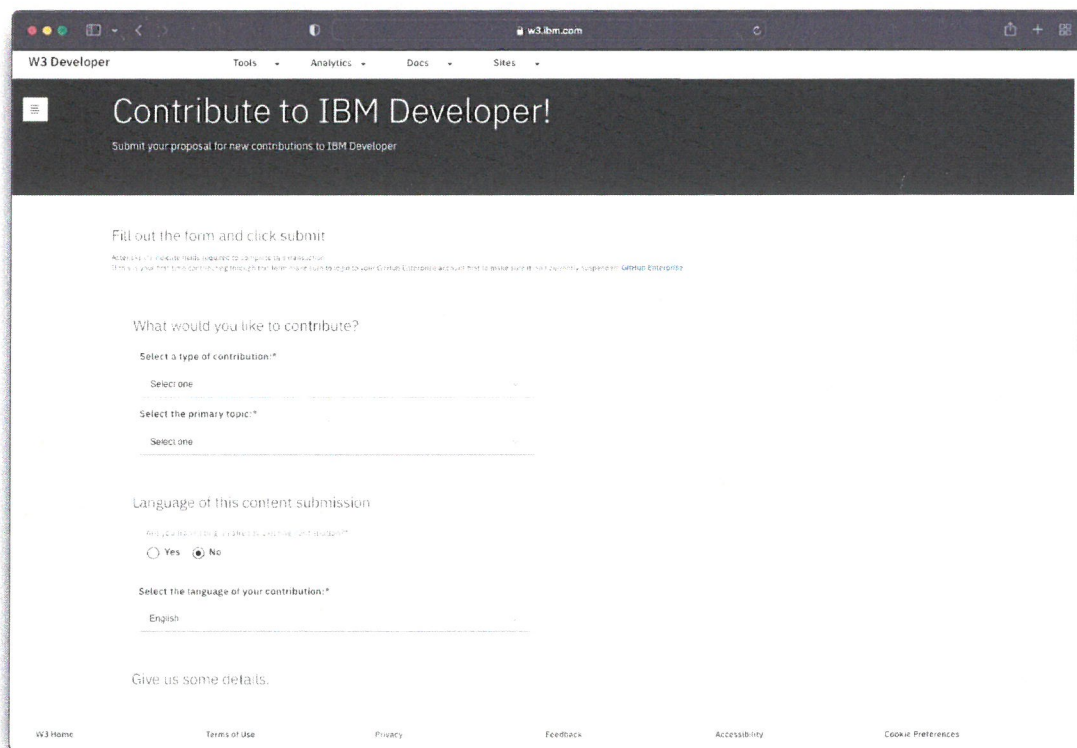
After this several other approaches were tried but none worked, so a custom POD which included IBM Cloud, IKS, Python and Docker was created. The pod was then used to connect to and use as a remote connection. This had several restrictions and complications as since it is a pod it cannot be created into a service and exposed to a IP address so instead, after connecting to the IKS using IBM Cloud login the Docker Daemon port was exposed to the local host of the Sandbox and then localhost port was used as host for running docker commands of creating and running the asset.



Module 4 GitHub Action:

This module which is a second approach for completing the above forementioned steps. It uses GitHub issues to take the GitHub URL as input and runs the feasibility checklist from there. Here a ProBot is used which checks for the issue created by the user and based on the input responses. It will either look for a GitHub repo and if it's not present it will provide the user with a boilerplate template to start working on making a repository for the project.

The working starts from a form which the user fill and then that will create a GitHub issue with the details filled to make it convenient.

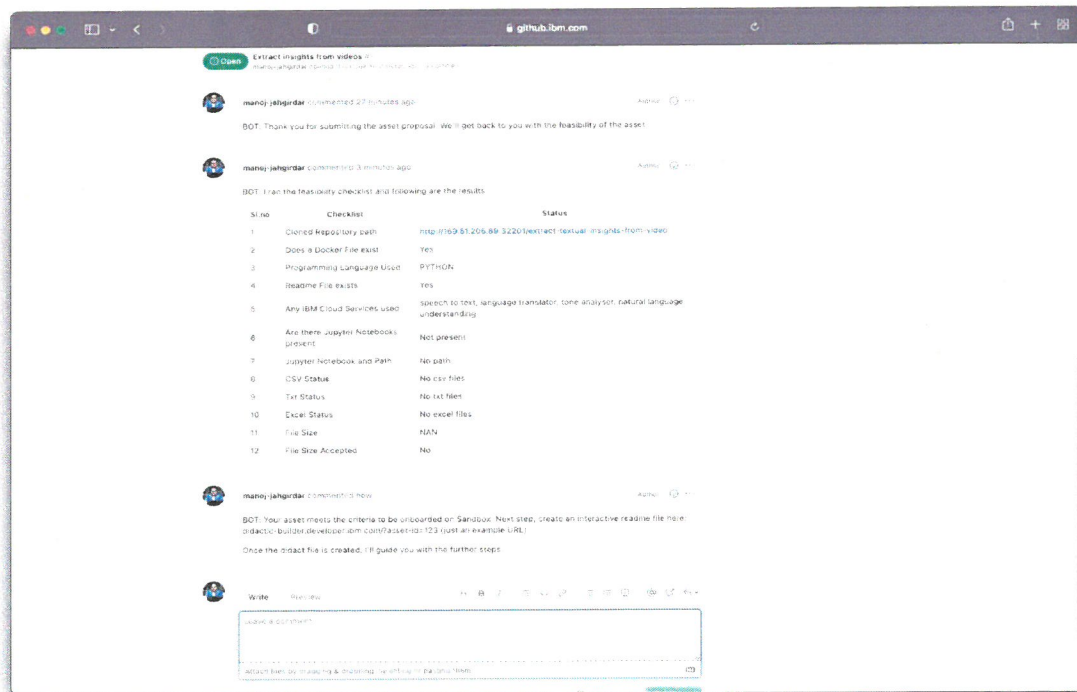
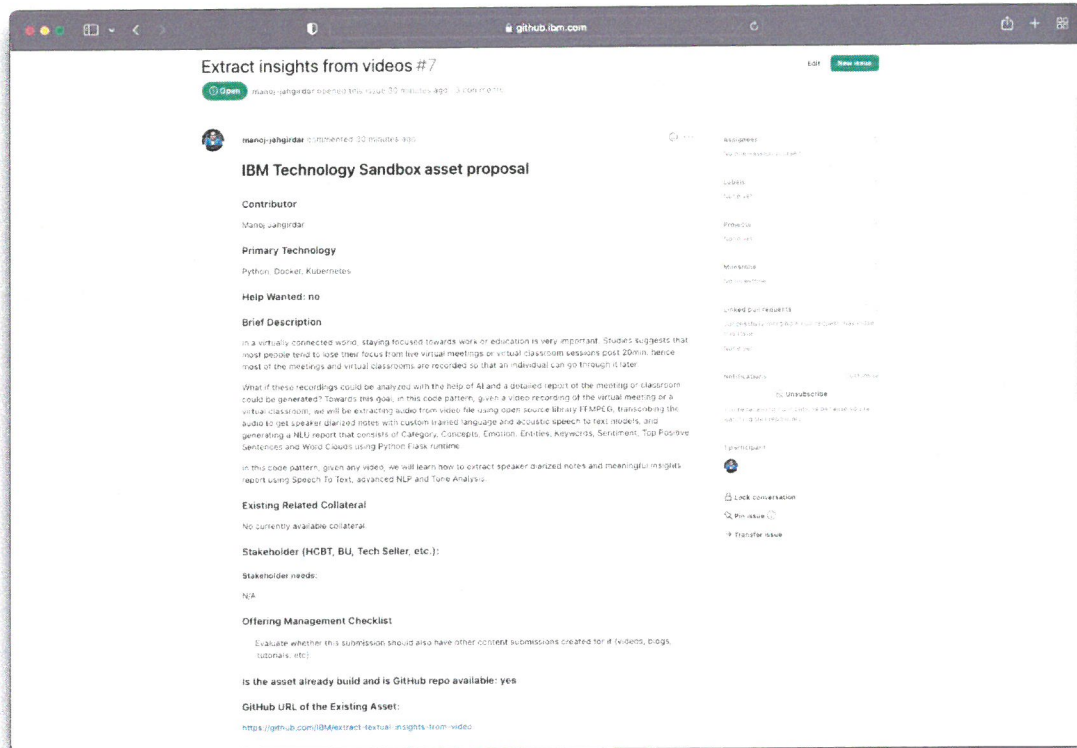


The screenshot shows a web browser window with the URL w3.ibm.com. The page is titled "W3 Developer" and "Contribute to IBM Developer!". Below the title, it says "Submit your proposal for new contributions to IBM Developer". The main content area is a form titled "Fill out the form and click submit". The form contains the following fields:

- What would you like to contribute?**
 - Select a type of contribution:*** (Dropdown menu with "Select one" selected)
 - Select the primary topic:*** (Dropdown menu with "Select one" selected)
- Language of this content submission**
 - Are you submitting content to contribute to IBM Developer?*** (Radio buttons: Yes, No. "No" is selected)
 - Select the language of your contribution:*** (Dropdown menu with "English" selected)
- Give us some details.** (Text area)

At the bottom of the page, there are links for "W3 Home", "Terms of Use", "Privacy", "Feedback", "Accessibility", and "Cookie Preferences".

The GitHub issue will trigger the ProBot



As can be seen above the ProBot picked up the URL from the git issue comment and then used it to run the feasibility checklist and gives user the output of the checklist as well as if the asset passes or fails.

Documentation and Finishing:

The final part of the project was the documentation of the work done, in which the complete flow from setup to installation to running the module as individual components is written and published on GitHub. For every module a different GitHub module was created as approach from the start has been that each module should be reusable for developers so that when they are working on their projects and they want to do something similar they can use these modules individually.

At the end the project I was working on was one of the three projects that were selected amongst all the intern projects at IBM, India to present to the IBM executives visiting, India from different countries. The project was presented to the executives at Bangalore office. The response was massive as the approach of the EcoLabs at India is changing to empower TechZone. This project exemplifies the entire approach and helps getting more and more assets on to TechZone for better reach and ease of use for customers, tech sellers and developers.

PROJECT DEMO:

The project demo sample video can be accessed by scanning the QR code below.

