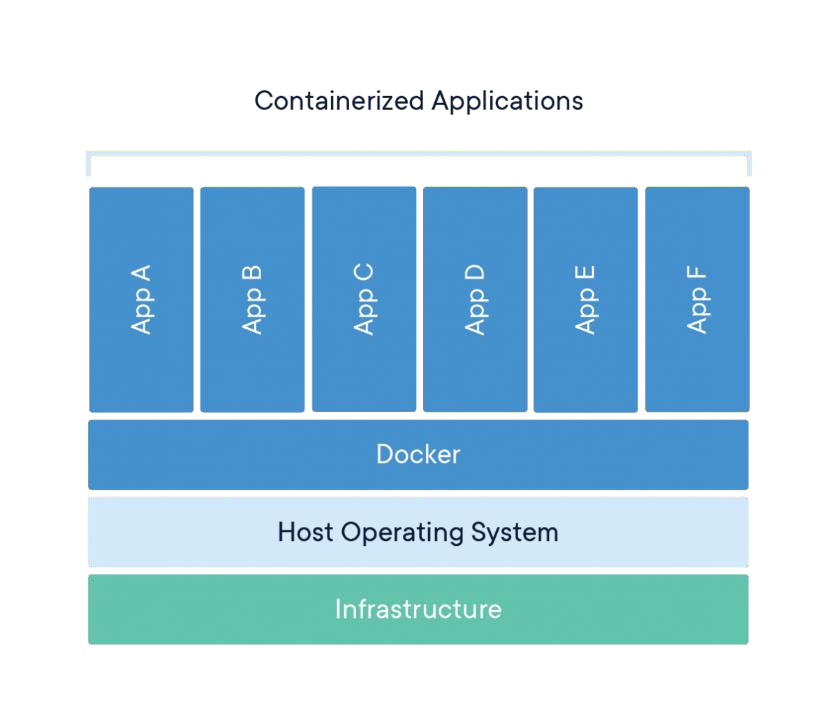
|  |  |
| --- | --- |
| Name Of The Student | Saikumar Bitra |
| Internship Project Topic | Set up docker container for application development using BlockChain network |
| Name of the Organization | TCSiON |
| Name of the Industry Mentor | Debashis Roy |
| Name of the Institute | K.B.N. College |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Start Date | End Date | Total Effort (hrs.) | Project Environment | Tools used |
| 14/07/2022 | 14/07/2022 | 156 | Linux/Ubuntu | GitHub, Docker, Ethereum, JDK, Eclipse, Solc, NodeJs, Truffle, Ganache, Solidity,  VIM, Git, etc |
| Project Synopsis:  Docker is a computer program that performs operating-system-level virtualization, also known as  containerization. It runs software packages as “containers.” The word “container” is borrowed from the transport industry. The container we see on the back of a truck, on a train, or on the ship, are all the same containers. Because these containers were standardized, it made transportation a whole lot easier. Cranes could be built to lift a container from a ship to a train. Imagine what we’d have to do before that? We had to open every container and unpack and pack goods from ship to train, train to truck, and it was all so inefficient.  Despite all the above, in the software development industry, this mistake is made each and every time. Every single time we have an application we need to deliver, we go through the same old rigmarole of setting up a Web server, setting up a website, a database, a firewall.  The advent of the cloud made these tasks seem possible with efficiency and ease because we want to control the environment our application runs in. We don't want other people's applications on a shared infrastructure to interfere with ours. We want efficiency and reliability. We want to ship our application packaged up as a container, easily configurable by our customers, so they can set things up quickly and easily. Most of all, we want reliability and security!  Docker simplifies all of the above. It is originated on Linux, but now it is over the Windows platform also. Docker can package an application along with all its dependencies in a virtual container and run it on any Linux server. This means that when we ship our application, we gain the advantages of virtualization, but we don't pay the cost of virtualizing the operating system.  What needs to be done in this project is set up a docker container from application development using a Blockchain network. The topic when broken down simplifies to:   1. Creating a dynamic docker image with docker compose and its variants 2. Test the dynamic docker image through the same docker 3. Creating a Blockchain network with Ganache | | | | |

|  |
| --- |
| 1. Install necessary plugins to support solidity 2. Once all of these are done, we also need to test the running processes and create a blockchain network. |
| Solution Approach:   * Creating a docker file with the following specifications:   1. Installing git (for version control)   2. Installing vim (required for editing the files)   3. Installing build-essential   4. Installing openJDK (Java Development Kit)   5. Configuring OpenJDK   6. Setting the environment variables if not already configured in the GUI setup   7. Installing Eclipse IDE and configuring the same in GUI setup   8. Install the YAKINDU plugin for Eclipse to support solidity   9. Set the YAKINDU plugin   10. Installing EVM and configuring the same   11. Installing Solc and configuring the same * Creating a docker file and creating an image * Once Image is created and up and running testing the image * Once Image is up and running update the Docker file and add the following   1. Install and configure NodeJS   2. Install and configure truffle packages   3. Install and configure testrpc   4. Initialize truffle projects   5. Deploying the contracts   6. Creating DAPP   7. Launch the DAPP server   8. Install and configure Ganache   9. Exposing the port   10. Configure the environment variables if needed * Create the docker image after updating the Docker files * Create a blockchain network with Ganache using the image created after updating the Docker file * Test the workspace and interface * Perform transactions |
| Assumptions:   * The OS used is Linux (Ubuntu) * Git is pre-installed and a root folder is created. * Docker is updated * The IDE used is Eclipse and the text editor used is VIM * Docker’s official GPG key is used |
| Project Diagrams: |



**INTERNSHIP: PROJECT REPORT**

|  |
| --- |
|  |

|  |
| --- |
| Algorithms:     * Setting Base ubuntu as ubuntu 18.04 * Running an update and installing git, vim, curl, OpenJDK * Install JRE * Installing the Eclipse IDE * Set environment variable:   ENV JAVA\_HOME=/usr/lib/jvm/java-11-openjdk-amd64/jre/bin/java   * Installing Ethereum * Installing NodeJS and NPM * Installing ganache, express and solc * Updating the system again after the installations are done * Making a directory and installing [truffle@4.1.15](mailto:truffle@4.1.15) * Exposing the port: 3000 in my case * While setting the truffleconfig set host to 127.0.0.1, port to 7545and chain id to 1337 * Set contracts and migration * Creating a simple bank application using Java script and testing it * Before doing of these stuff, we need to install metamask extension for doing transactions and it will helps us to linking the BlockChain network |
| Outcome:   * Dynamic Docker Image is up and running * Docker compose file is ready * Image was tested and ready * BlockChain network created with Ganache CLI * The application development container is ready * The platform is tested using smart contracts * Pictures of transactions from one account to another account is done through ganache test coins   Step-1: - The picture below demonstrates the starting page of application.    Step-2: - To do the transaction, first of all we have to connect the wallet by clicking on the  blue button as shown in the above picture.    Step-3: - Before going to next button, we have to import the test accounts by using ganache  private keys. In this case, I had imported two accounts by using ganache. For doing  transaction, we have to select only one account with some sufficient Ethereum  coins. Lastly, click on the next button as shown in the picture below.    Step-4: - Now we are successfully connected with wallet. It’s time to do transaction. For that  we need to copy the address of any account from our wallet. In this case, I am  copying address of account 1 from my wallet as shown in the picture below.      Step-5: - In this step we have to paste the address which we had copied in the previous, add  the required amount of Ethereum. At last add the text message and hit the send  button.    Step-6: - After hitting the send now button, it will pop us a dialog box of MetaMask, which  shows the transactions fee and some other details of transactions. Now click the  confirm button for confirming the transaction as shown in the picture below.          Step-7: - Finally check the transaction by opening the ganache application. In the application  we have to choose the option transactions. In there we can find our all transactions  with all necessary details. |
| Exceptions considered:   * The image will run but the site can’t be open if there is any anti-virus in the pc or issues   with versions of ubuntu os and docker.   * The Ubuntu version installed is 18.04 * The system was updated at the very first place * Environment variable for Eclipse and the relevant packages are installed |
| Enhancement Scope:   * The docker file can be enhanced and the execution can be made fast * The security of the whole thing can be made better * Other than the two above points the all the other things are covered to the best of my knowledge. |
| Link to Code and executable file: -  URL: - <http://thirupathitr.unaux.com/>  Docker-Command: - docker pull thriu124/thirupathi  Git-Hub: - https://github.com/software-babooi/BlockChain.git |