



Software Production Engineering (CS816) (1)

Major Project Buttercrust

Under the guidance of Professor B. Thangaraju and Vaibhav Tandon

Group Members

- Harshit Agrawal (MT2021051)
- Samrat Seth (MT2021113)

Major Project Buttercrust

Group Members

1. Abstract

2. What is DevOps?

3. Why DevOps?

4. System Configuration

4.1 Operation System

4.2 CPU and RAM

4.3 Frameworks

4.4 Database

4.5 Building Tools

4.6 AWS EC2 Instance

4.6 DevOps Tools

5. Software Development Life Cycle (SDLC)

5.1. Source Code Management (SCM):

(Link to the repository: <https://github.com/sethsamrat/Buttercrust-App>)

5.2 Build

5.3 Testing

5.4 Docker

1. Abstract

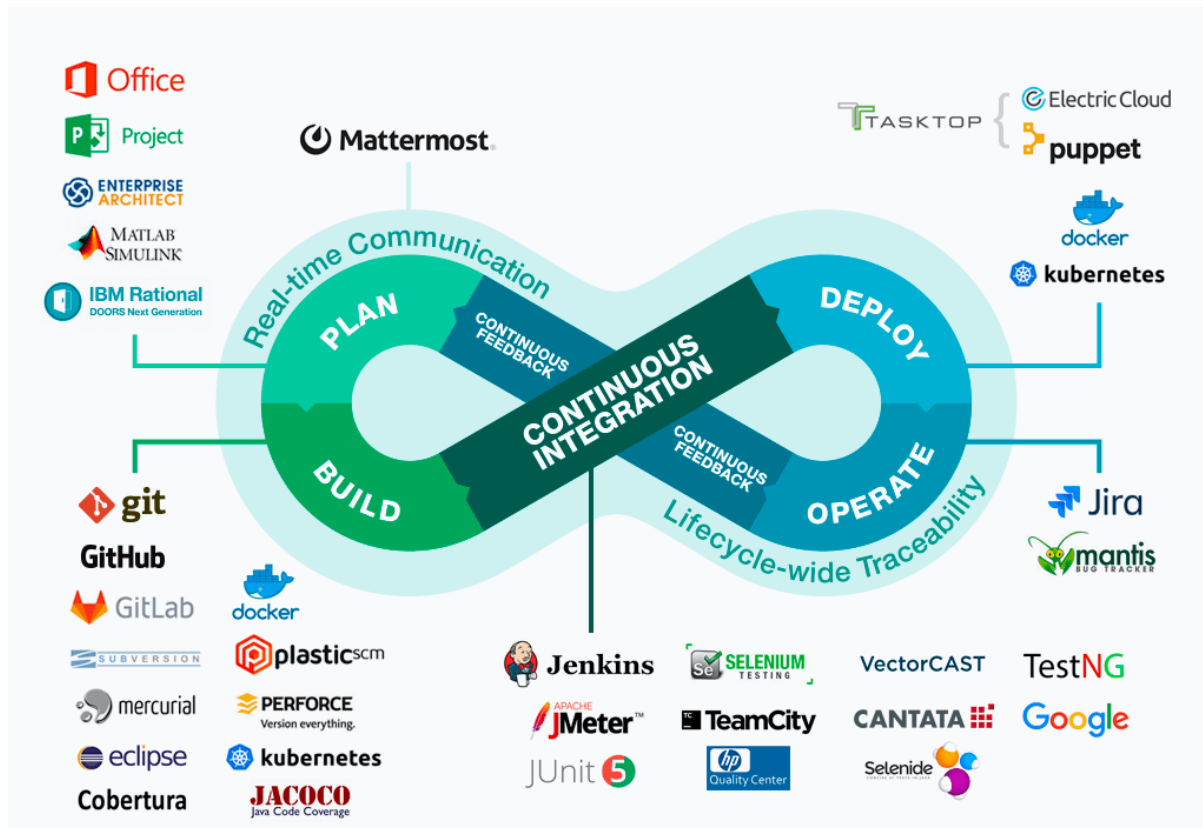
The online Pizza ordering system is a web-based application that enables customers to order their pizzas online for home delivery. Each country has its own kind of dishes to offer. But if we pick a food item that is loved by all the people on this planet, then pizza will be a clear winner in it. The whole world is in love with pizzas. The billions of dollars earned by different pizzerias across the globe just prove this. The love of pizzas has enabled the rise of large pizza companies like Pizza Hut, Domino's, Papa John's, and much more.

As the internet users are increasing exponentially, these companies have introduced an Online Pizza ordering system for taking orders from customers. This system not only improves the customer experience but also eases the workload on the staff of pizzerias.

This is a Full Stack (MERN) Pizza Delivery Application developed using React for Front End, Redux-Thunk for Asynchronous operations, Node JS for Runtime environment, Express JS for Backend Routing, and Mongo DB for Database.

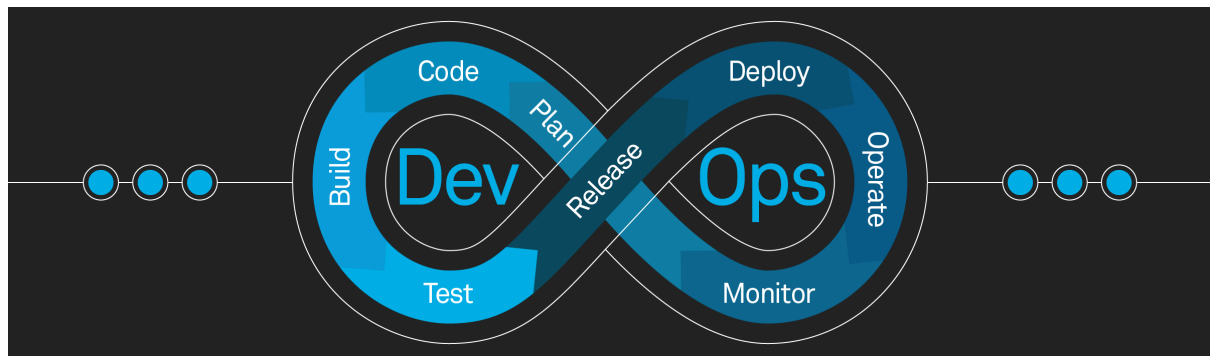
2. What is DevOps?

1. DevOps is the practice of operations and development engineers participating together in the entire service lifecycle, from design through the development process to production support.
2. DevOps is also characterized by operations staff making use of many of the same techniques/tools as developers for their systems work.
3. DevOps is the combination of cultural philosophies, practices, and tools that increases an organization's ability to deliver applications and services at high velocity: evolving and improving products at a faster pace than organizations using traditional software development and infrastructure management processes. This speed enables organizations to better serve their customers and compete more effectively in the market.



3. Why DevOps?

We plan to build this project in a growing way. The design has ideas and services that work independently. Given the complexity of the project, it is impossible for any of us to create and test the entire code manually every time we make a small change. And since the three of us work from different locations, the automatic pipeline will not only make our job easier, and make it more efficient. The amount of communication that must take place between us will decrease. DevOps helps us focus on key aspects of the project, improving efficiency, stability, and security. There is a small range of manual errors as well. And since we plan to build this product at some point, continuous delivery makes it easier. Also, monitoring allows us to better understand usage and help us improve the application.



4. System Configuration

4.1 Operation System

- Ubuntu 20.04.4 LTS (Focal Fossa)

4.2 CPU and RAM

- Ryzen 9 CPU and 16 GB Ram

4.3 Frameworks

- React JS
- Node JS

4.4 Database

- MongoDB

4.5 Building Tools

- npm (npm is a package manager for the JavaScript programming language)

4.6 AWS EC2 Instance

- Type - T2.medium
- OS - Ubuntu 20.04
- Ram - 2 GiB
- Storage - 15 GiB

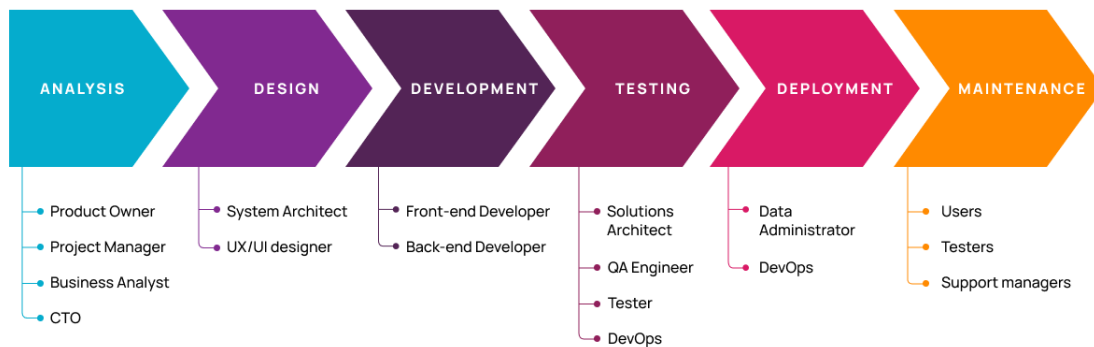
4.6 DevOps Tools

- GitHub: Version control system

- Jenkins: CI/CD pipeline
- Ansible: Configuration management and infrastructure as code
- ELK: Monitoring
- Docker: Deployment/Containerization

5. Software Development Life Cycle (SDLC)

6 Phases of the Software Development Life Cycle



5.1. Source Code Management (SCM):

(Link to the repository:

<https://github.com/sethsamrat/Buttercrust-App>)

- SCMs are used to give versions/revisions to the program. Each type is given a timestamp and includes the person responsible for the change. Even different versions can be compared and integrated with other types. That is why SCM is also called Version Control, Revision Control or Source Control.
- In order to achieve SCM, we need to create a GitHub repository on github.com by specifying the name and description of the project. This creates an empty repository on GitHub.

- We can also add a readme file in the repository that contains some information about the project. After creating an empty repository on GitHub we need to clone an empty project to the local system. This would create a directory in the name of the project in which we can add the files of our project. In this directory, add all the files of the project.
- Now add these files to the staging area.

```
$ git add *
```

Then commit those changes so that the files would get added to the local repo.

```
$ git commit -m "message"
```

Now in order to add these files to the GitHub repo, we need to push the files to the repo.

```
$ git push origin master
```

We have now successfully added our project to the GitHub which enables the other users to use the same project and made required modifications to the project by git pull. After pushing the files to GitHub the GitHub would look as below.

Repository

sethsamrat / Buttercrust-App Public

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

master 1 branch 0 tags Go to file Add file Code

File	Commit Message	Time Ago
sethsamrat Update inventory		4981061 18 hours ago 27 commits
client	updated docker files	2 days ago
models	Initial commit	17 days ago
node_modules	Logging And Testing	18 hours ago
routes	Logging And Testing	18 hours ago
utils	Logging And Testing	18 hours ago
Dockerfile	updated docker files	2 days ago
Jenkinsfile	Added Jenkinsfile	yesterday
combined.log	Logging And Testing	18 hours ago
db.js	Initial commit	17 days ago
docker-compose.yml	updated compose	2 days ago
inventory	Update inventory	18 hours ago
package-lock.json	Logging And Testing	18 hours ago
package.json	Logging And Testing	18 hours ago
playbook.yml	updated playbook	yesterday
server.js	Initial commit	17 days ago
test.py	Added test.py	22 hours ago

About

No description, website, or topics provided.

0 stars
1 watching
0 forks

Releases

No releases published
[Create a new release](#)

Packages

No packages published
[Publish your first package](#)

Languages

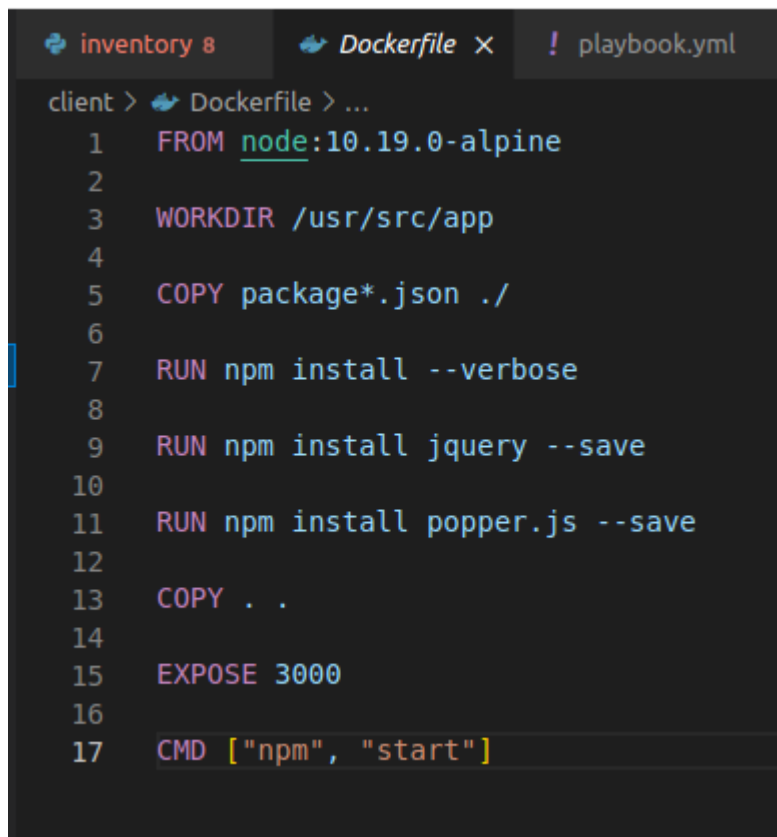
JavaScript 92.2% CSS 3.1%
HTML 3.0% Python 1.1%
Dockerfile 0.6%

5.2 Build

5.3 Testing

5.4 Docker

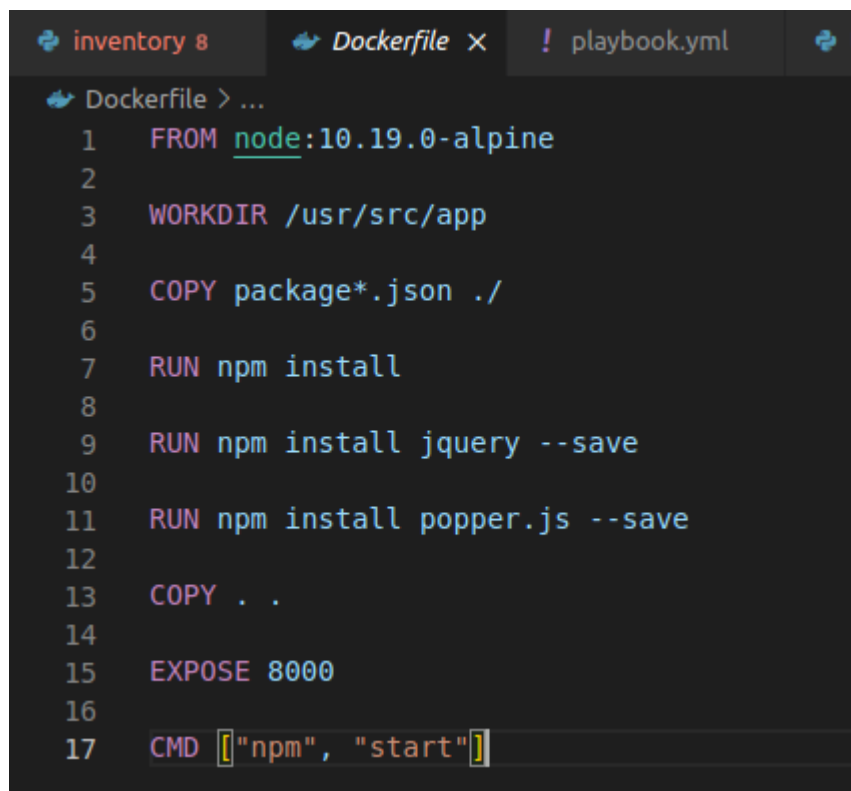
- Docker enables developers to easily pack, ship, and run any application as a lightweight, portable, self-sufficient container, which can run virtually anywhere. As Bottomley told me, containers give you instant application portability.
- Containers do this by enabling developers to isolate code into a single container. This makes it easier to modify and update the program. It also lends itself, as Docker points out, for enterprises to break up big development projects among multiple smaller, Agile teams using Jenkins, an open-source CI/CD program, to automate the delivery of new software in containers.
- **Dockerfile for Client Image**



The screenshot shows a code editor with three tabs: 'inventory 8', 'Dockerfile x', and 'playbook.yml'. The 'Dockerfile' tab is active, displaying a Dockerfile with 17 lines of code. The code starts with 'FROM node:10.19.0-alpine', sets 'WORKDIR /usr/src/app', copies 'package*.json' to the current directory, runs 'npm install --verbose', 'npm install jquery --save', and 'npm install popper.js --save'. It then copies the current directory to the container, exposes port 3000, and sets the command to 'npm start'.

```
client > Dockerfile > ...
1 FROM node:10.19.0-alpine
2
3 WORKDIR /usr/src/app
4
5 COPY package*.json ./
6
7 RUN npm install --verbose
8
9 RUN npm install jquery --save
10
11 RUN npm install popper.js --save
12
13 COPY . .
14
15 EXPOSE 3000
16
17 CMD ["npm", "start"]
```

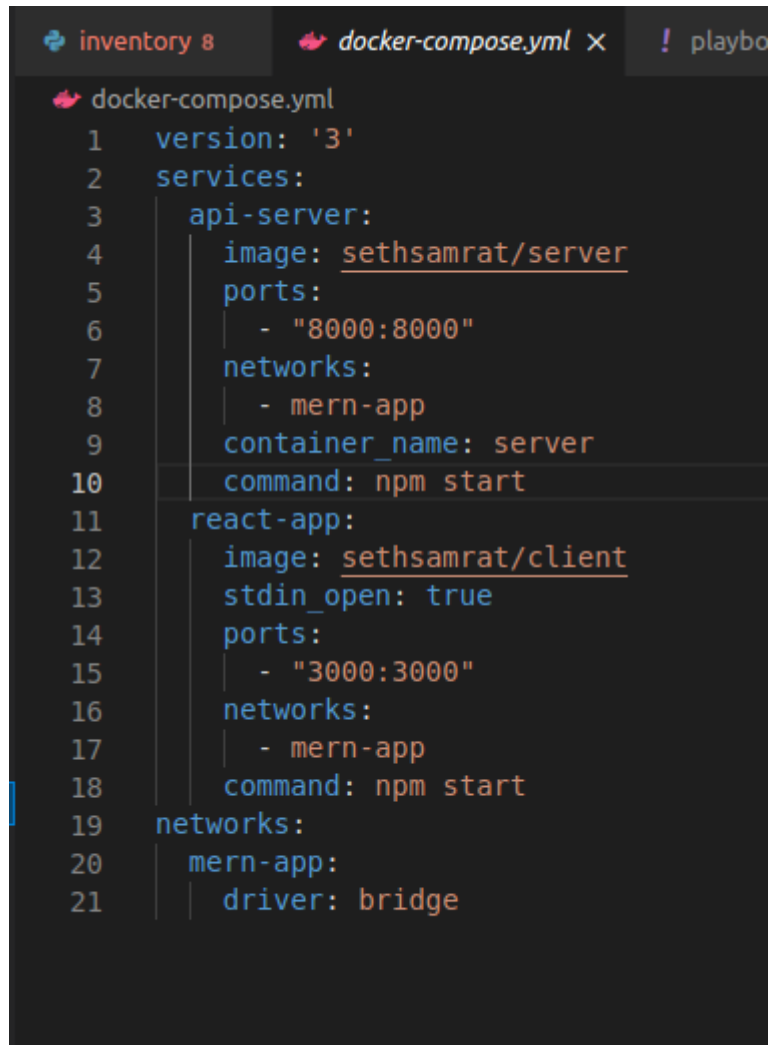
- Dockerfile for Server Image



The screenshot shows a code editor with three tabs: 'inventory 8', 'Dockerfile x', and 'playbook.yml'. The 'Dockerfile' tab is active, displaying a Dockerfile with 17 lines of code. The code starts with 'FROM node:10.19.0-alpine', sets 'WORKDIR /usr/src/app', copies 'package*.json' to the current directory, runs 'npm install', 'npm install jquery --save', and 'npm install popper.js --save'. It then copies the current directory to the container, exposes port 8000, and sets the command to 'npm start'.

```
Dockerfile > ...
1 FROM node:10.19.0-alpine
2
3 WORKDIR /usr/src/app
4
5 COPY package*.json ./
6
7 RUN npm install
8
9 RUN npm install jquery --save
10
11 RUN npm install popper.js --save
12
13 COPY . .
14
15 EXPOSE 8000
16
17 CMD ["npm", "start"]
```


- Docker-Compose



```
1 version: '3'
2 services:
3   api-server:
4     image: sethsamrat/server
5     ports:
6       - "8000:8000"
7     networks:
8       - mern-app
9     container_name: server
10    command: npm start
11  react-app:
12    image: sethsamrat/client
13    stdin_open: true
14    ports:
15      - "3000:3000"
16    networks:
17      - mern-app
18    command: npm start
19  networks:
20    mern-app:
21      driver: bridge
```

- Repositories in DockerHub

[Explore](#)
[Repositories](#)
[Organizations](#)
[Help](#)
[Upgrade](#)

[Create Repository](#)

sethsamrat / server

Not Scanned
0
3
Public

sethsamrat / client

Not Scanned
0
3
Public

sethsamrat / butternutcrust

Not Scanned
0
0
Public

sethsamrat / calcimage

Not Scanned
0
1
Public

Tip: Not finding your repository? Try switching namespace via the top left dropdown.

Create an Organization
Manage Docker Hub repositories
with your team

May 9 - 10

DockerCon 2022

• Client Repository

sethsamrat / client

This repository does not have a description

Last pushed: 18 hours ago

Docker commands

To push a new tag to this repository,

docker push sethsamrat/client:tagname

Tags and Scans

VULNERABILITY SCANNING - DISABLED
Enable

This repository contains 1 tag(s).

TAG	OS	PULLED	PUSHED
latest		a day ago	18 hours ago

[See all](#)

Automated Builds

Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available with Pro, Team and Business subscriptions.

[Upgrade to Pro](#)
[Learn more](#)

• Server Repository

sethsamrat / server

This repository does not have a description

Last pushed: 18 hours ago

Docker commands

To push a new tag to this repository,

docker push sethsamrat/server:tagname

Tags and Scans

VULNERABILITY SCANNING - DISABLED
Enable

This repository contains 1 tag(s).

TAG	OS	PULLED	PUSHED
latest		---	18 hours ago

[See all](#)

Automated Builds

Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available with Pro, Team and Business subscriptions.

[Upgrade to Pro](#)
[Learn more](#)

