# DEPLOYMENT INSTRUCTIONS:

NOTE:
WE CAN MAKE THE WEBSITE LIVE IF INFORMED :
http://acerelocations-frontend.s3-website.us-east-2.amazonaws.com/

## Frontend Files:

Frontend is developed in React.js, HTML and CSS is used for styling. Frontend consist of the following folders and files.

- **HTML Folder** – Consist of static html files used to create components in React.js.
- **Public Folder** – Consists of default logo images and robots.txt file for security.
- SRC Folder consist of the main frontend code:
  - **Assets**: Has the images, CSS, javascript and book PDF used in the UI.
  - **Components**: components consist of commonly used React components by other unique components like header, sidebar and loader.
  - **Http**: This folder has the routing information to the backend API using axios package.
  - **Routes**: This folder consists of major frontend routes for redirection and linking,
  - **Store**: This folder is used to maintain the Redux code in React.js, it helps the code remember a state by storing as a local storage.
  - **Views**: This folder consists of all the major unique frontend pages as react components and integrated with relevant API.
- **Package.JSON** – is a file with all the package information and relevant versions.

## Frontend Deployment:

- **Ubuntu/ Linux**
  - Update the system
    ```
    sudo apt update
    ```
  - Install Node and npm
    ```
    sudo apt install nodejs npm
    ```
  - Check if the node version is 10.19 and npm is 6.13.0
    ```
    nodejs –version
    npm -v
    ```
  - Install all dependencies
    ```
    npm install
    ```
  - Start the Code
    ```
    npm start
    ```

- **Windows**
  - Use the following tutorial if you are using windows: https://blog.theodo.com/2017/01/use-git-ssh-and-npm-on-windows-with-git-bash/

## Backend

Following is the directory structure from the backend:

**Directory Tree Diagram**

https://github.com/sayeeshruthiwindsor/BookHUB/blob/main/Backend_DirectoryInfo

***Note: Backend is live and is not required to be deployed.***

Backend Deployment

## Django Configuration

- pip3 install Django
- pip3 install djangorestframework
- pip3 install djangorestframework-jsonapi
- pip3 install djongo
- pip3 install django-unixtimestampfield
- pip3 install django-fernet-fields==0.6
- pip3 install -r requirement.txt

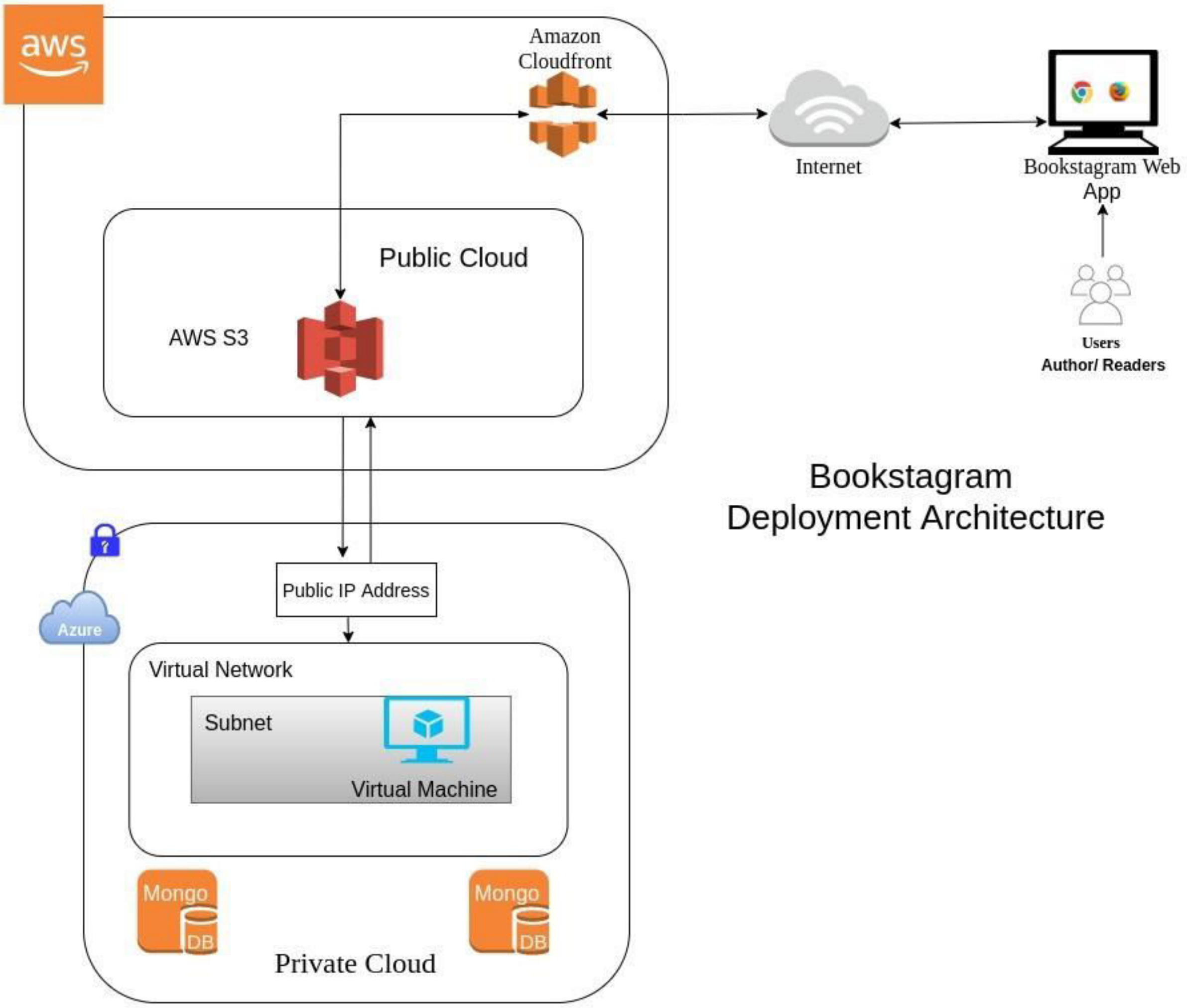
## Backend Server Configuration

- hostnamectl set-hostname Bookstagram //creating Host name
- sudo vi /etc/hosts server-ip hostname // entering in hosts
- sudo apt-get install ufw // uncomplecated firewall
- sudo ufw default allow outgoing
- sudo ufw allow 80 // Apache2 Hosted Post
- sudo ufw allow ssh
- sudo ufw enable
- sudo ufw status
- scp -r Bookstagram green@IP:/home/book/Bookstagram (cp local dir to server)

- sudo apt-get install python3-pip (pip3)
- sudo apt-get install python3-venv (virtual env)
- python3 -m venv venv (create venv)
- source venv/bin/activate (Activate)
- pip3 install django
- pip3 install djongo // Django to MongoDB helper
- install MongoDB Server --> https://docs.mongodb.com/manual/tutorial/install-mongodb-on-ubuntu/
- pip3 install wheel
- pip3 install -r require.txt
- pip3 install django-cors-headers // Django library to handle CORS headers
- python manage.py makemigrations (Migrate Django)
- python manage.py migrate
- check code
- sudo apt-get install apache2
- sudo apt-get install libapache2-mod-wsgi-py3
- sudo vi book.conf (Apache conf File)
- sudo systemctl start apache2

Page Break

# DEPLOYMENT ARCHITECTURE



Bookstagram Deployment Architecture

- Users: We have two kind of users Authors and Readers on this platform.
- Bookstagram Web App: Web app is hosted on the internet and can be accessed using browsers like Chrome and Mozilla.
- Amazon CloudFront: Is the content delivery network that helps to deliver images, videos and other data faster to the end user.
- AWS S3: It is a static bucket where we have hosted our frontend build. It is in the Public Cloud to facilitate connection from any IP.
- Azure Virtual Machine: Bookstagram backend is hosted on Azure virtual machine with MongoDB as the database in the private cloud. This private cloud has been given the access from the frontend URL and default localhost URL to decrease any chance of attacks.