**Workflow of VibeLinkModel\_EC2\_FINALDEPLOY:**

When a docker container is built and run it encapsulates the entire application environment, ensuring that it works fine across all systems. Here is on overview of how this works:

**Dockerfile**: Defines the environment for the application.

**Docker Compose:** Manages multi-container applications.

**Environment Variables:** Configures the application using a .env file.

**Project Structure:** Organizes the application files.

**Build and Run:** Instructions for building and running the Docker containers.

**Application Interaction:** Describes how the application components interact.

The Dockerfile is used to create Docker images for this Django application. Set the working directory, install all the dependencies, copy the rest of the application port, make port 8000 available to host and set the environment variables.

Docker-compose.yml is used to manage services for the application.

Environment variables are managed in the .env file.

Project Structure is like:

.idea

.venv

\_pycahce\_

Docs

Excerpt

\_pycache\_

Migrations

\_init\_.py

admin.py

apps.py

models.py

tests.py

urls.py

utils.py

views.py

Helper

Inputs

JacobRyanMedia

Logs

Services

Static

Staticfiles

Tmp

……

.env

.dockerignore

Dockerfile

Docker-compose.yml

…..

This is the project structure with important files and folders.

**Application Interaction**

The ChatBot class in views.py handles requests sent to the /bot endpoint when someone wants to interact with the chatbot. When a user makes a request through POST, the chatbot gathers important details like the user's ID, username, and the message they want to send. It then processes this information, prepares a response, and sends it back to the user in a structured format called JSON. This interaction allows users to query the chatbot and receive answers based on the input provided.

Same is the application interaction for the rest of API’s endpoints.