Manual de instalación pruebas unitarias automatizadas:

1. Install Docker on Ubuntu:
    - Update the apt package index: sudo apt-get update
    - Install the required packages to allow apt to use a repository over HTTPS: sudo apt-get install apt-transport-https ca-certificates curl gnupg lsb-release
    - Add Docker's official GPG key: curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
    - Set up the stable repository: echo "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
    - Update the apt package index again: sudo apt-get update
    - Install Docker Engine: sudo apt-get install docker-ce docker-ce-cli containerd.io
    - Verify that Docker is installed correctly by running the hello-world container: sudo docker run hello-world
2. Install the ODBC Driver for SQL Server on Ubuntu:
    - Download the latest ODBC Driver for SQL Server package for Ubuntu from Microsoft's website: curl https://packages.microsoft.com/config/ubuntu/22.04/prod.list > /etc/apt/sources.list.d/mssql-release.list
    - Import the package signing GPG key: curl https://packages.microsoft.com/keys/microsoft.asc | sudo apt-key add -
    - Update the apt package index: sudo apt-get update
    - Install the ODBC Driver for SQL Server: sudo apt-get install msodbcsql17
3. Clone the project repository and navigate to the directory containing the Docker Compose file:
    - Install git if it is not already installed: sudo apt-get install git
    - Clone the project repository: git clone <repository_url>
    - Navigate to the project directory: cd <project_directory>
4. Modify the backend environment variable in the Docker Compose file:
    - Open the Docker Compose file in a text editor: nano docker-compose.yml
    - Replace the DB_CONNECTION_STR value with your SQL Server connection string, if necessary
5. Build and start the Docker containers:
    - Run the docker-compose up command to build and start the Docker containers: sudo docker-compose up
    - Verify that the containers are running by visiting the following URLs in a web browser:
        - Frontend: http://localhost
        - Backend: http://localhost:8000

That's it! You have successfully installed the Docker Compose file on your Ubuntu 22.04 virtual machine and can now access the frontend and backend services.

To specify the URL of the SQL Server database in the Docker Compose file, you can use the DB_CONNECTION_STR environment variable in the backend service definition.

In the provided Docker Compose file, the value of DB_CONNECTION_STR is set to "mssql+pyodbc://sa:YourStrong!Passw0rd@db/api_tester_db?driver=ODBC+Driver+17+for+SQL+Server", where db is the name of the SQL Server container and api_tester_db is the name of the database that will be used.

If you need to modify the URL to connect to a different SQL Server instance or database, you can update the DB_CONNECTION_STR value accordingly. For example, to connect to a SQL Server instance running on a different host and port with a different database name and user credentials, you could use a URL like the following:

DB_CONNECTION_STR:
"
mssql+pyodbc://<user>:<password>@<hostname>:<port>/<database>?driver=ODBC+Driver+17+for+SQL+Server
"

Replace <user>, <password>, <hostname>, <port>, and <database> with the appropriate values for your SQL Server instance. Note that you may need to install additional ODBC drivers or configure other settings depending on your specific environment.