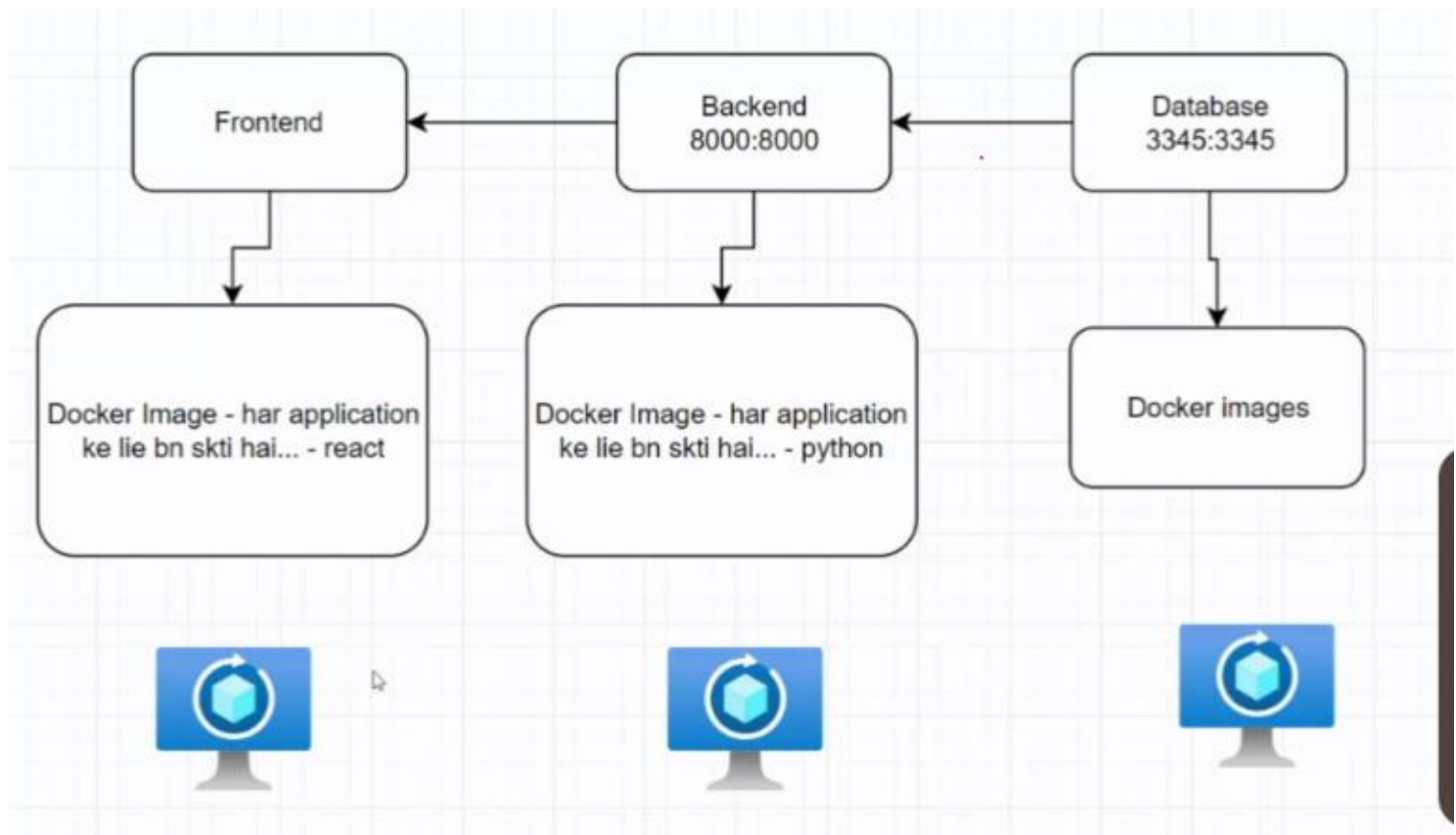
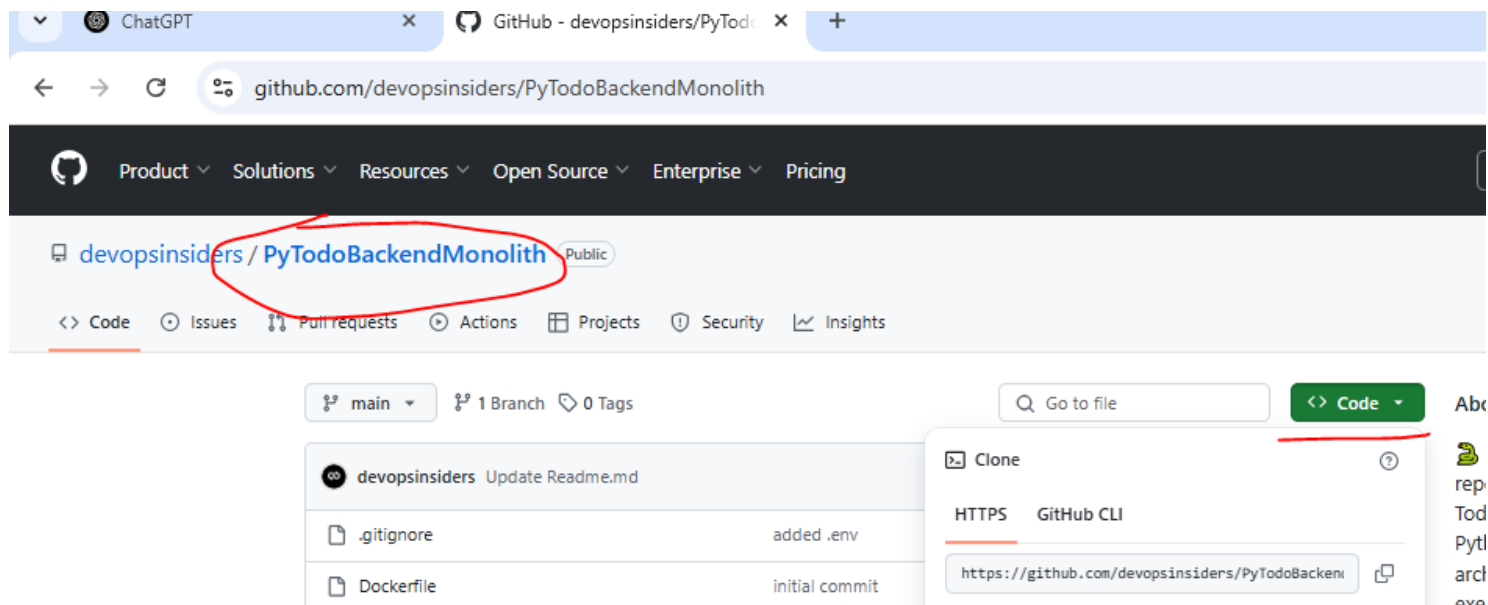


28 September 20204



1) Now bringing backend code



git clone https://github.com/devopsinsiders/PyTodoBackendMonolith.git

Is

```

azureuser@vmdocker:~$ git clone https://github.com/devopsinsiders/PyTodoBackendMonolith.git
Cloning into 'PyTodoBackendMonolith'...
remote: Enumerating objects: 59, done.
remote: Counting objects: 100% (59/59), done.
remote: Compressing objects: 100% (52/52), done.
remote: Total 59 (delta 30), reused 21 (delta 5), pack-reused 0 (from 0)
Receiving objects: 100% (59/59), 14.69 KiB | 1.33 MiB/s, done.
Resolving deltas: 100% (30/30), done.
azureuser@vmdocker:~$ ls
PyTodoBackendMonolith  ReactTodoUIMonolith  snap
azureuser@vmdocker:~$

```

2) **cd PyTodoBackendMonolith/**

**ls**

```

PyTodoBackendMonolith  ReactTodoUIMonolith  snap
azureuser@vmdocker:~$ cd PyTodoBackendMonolith/
azureuser@vmdocker:~/PyTodoBackendMonolith$ ls
Dockerfile  Readme.md  app.py  requirements.txt

```

3) So we have a Dockerfile into it also

```

azureuser@vmdocker: ~/PyTodoBackendMonolith
GNU nano 7.2 Dockerfile
# Use the official Python image as the base image
FROM python:3.9

# Set the working directory in the container
WORKDIR /app

# Copy the application files into the container
COPY . .

# Install necessary packages
RUN apt-get update && apt-get install -y unixodbc unixodbc-dev
RUN curl https://packages.microsoft.com/keys/microsoft.asc | apt-key add -
RUN curl https://packages.microsoft.com/config/debian/10/prod.list > /etc/apt/sources.list.d/mssql-release.list
RUN apt-get update
RUN ACCEPT_EULA=Y apt-get install -y msodbcsql17

RUN pip install -r requirements.txt

# Start the FastAPI application
CMD ["uvicorn", "app:app", "--host", "0.0.0.0", "--port", "8000"]

```

4) Now editing docker file to create custom image

root@vmdocker: /home/azureuser/PyToDoBackendMonolith

GNU nano 7.2

Dockerfile

```
# Use the official Python image as the base image
FROM python:3.9.20

# Set the working directory in the container
WORKDIR /backendapp/

# Copy the application files into the container
COPY . /backendapp/

# Install necessary packages
RUN apt-get update && apt-get install -y unixodbc unixodbc-dev
RUN curl https://packages.microsoft.com/keys/microsoft.asc | apt-key add -
RUN curl https://packages.microsoft.com/config/debian/10/prod.list > /etc/apt/sources.list.d/mssql-release.list
RUN apt-get update
RUN ACCEPT_EULA=Y apt-get install -y msodbcsql17

RUN pip install -r requirements.txt

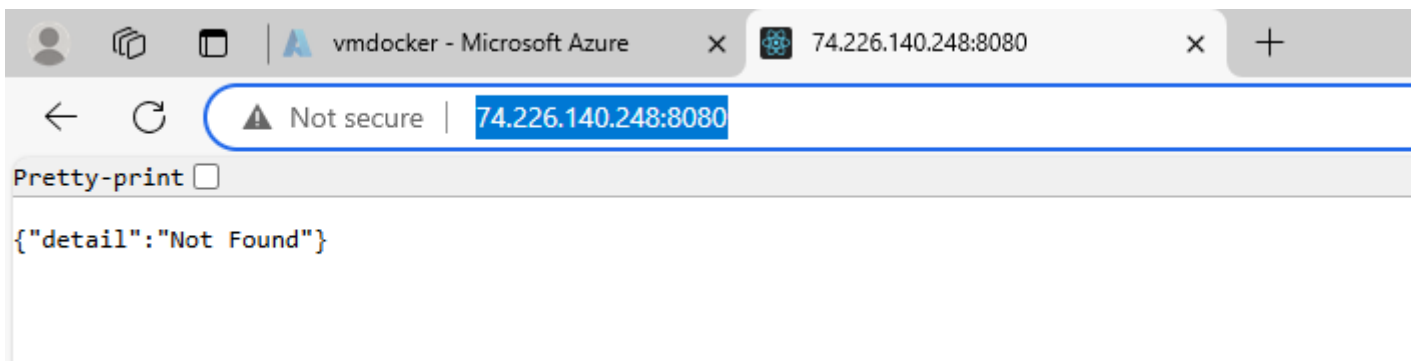
# Start the FastAPI application
#CMD ["uvicorn", "app:app", "--host", "0.0.0.0", "--port", "8000"]
```

## 5) docker images

```
root@vmdocker:/home/azureuser/PyToDoBackendMonolith# docker images
REPOSITORY          TAG         IMAGE ID      CREATED        SIZE
pytodo              latest      8badf51cfb98  17 seconds ago 1.11GB
bahutpatliimage    latest      d639e855c309  29 hours ago  50.4MB
patliimage          latest      94df29409897  29 hours ago  195MB
todoui              latest      8cfbe43bc4ca  31 hours ago  1.34GB
<none>              <none>      fea1e545debc  34 hours ago  1.34GB
<none>              <none>      4d09465a1759  2 days ago    1.34GB
<none>              <none>      f1ebcad628d3  3 days ago    1.34GB
root@vmdocker:/home/azureuser/PyToDoBackendMonolith# nano Dockerfile
root@vmdocker:/home/azureuser/PyToDoBackendMonolith#
```

6) **docker run -d -p 8080:8000 pytodo uvicorn app:app --host 0.0.0.0 --port 8000** = running image to create and run container

[74.226.140.248:8080](http://74.226.140.248:8080)



7) Docker file is useful to make any command as default so

root@vmdocker: /home/azureuser/PyTodoBackendMonolith

GNU nano 7.2

Dockerfile \*

```
# Use the official Python image as the base image
FROM python:3.9.20

# Set the working directory in the container
WORKDIR /backendapp/

# Copy the application files into the container
COPY . /backendapp/

# Install necessary packages
RUN apt-get update && apt-get install -y unixodbc unixodbc-dev
RUN curl https://packages.microsoft.com/keys/microsoft.asc | apt-key add -
RUN curl https://packages.microsoft.com/config/debian/10/prod.list > /etc/apt/sources.list.d/mssql-release.list
RUN apt-get update
RUN ACCEPT_EULA=Y apt-get install -y msodbcsql17

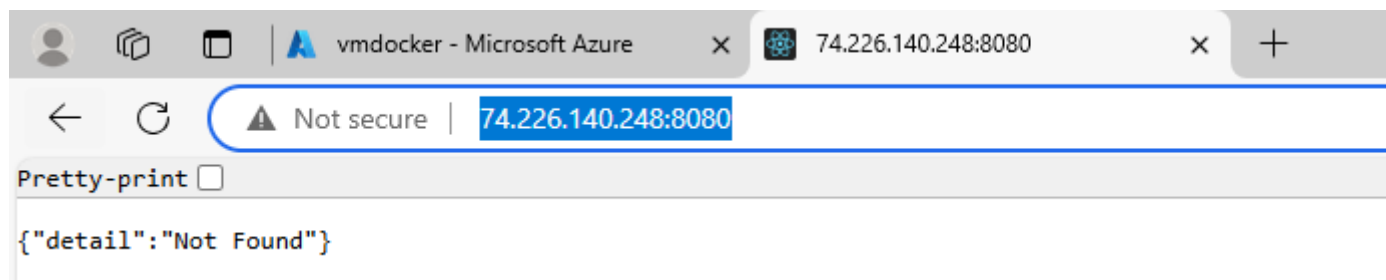
RUN pip install -r requirements.txt

# Start the FastAPI application
CMD ["uvicorn", "app:app", "--host", "0.0.0.0", "--port", "8000"]
```

**docker run -d -p 8000:8000 pytodo =**

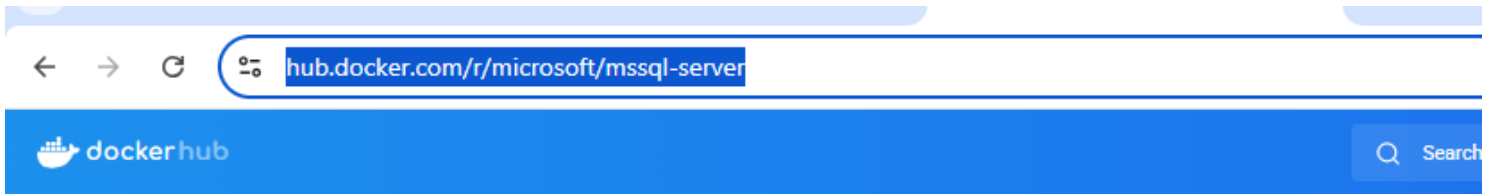
```
root@vmdocker:/home/azureuser/PyTodoBackendMonolith# docker run -d -p 8000:8000 pytodo
66d2a20f7c3e551e458cc1c60aa1019e960f0ca49d4790ab1671753ddb4a2304
```

[74.226.140.248:8080](http://74.226.140.248:8080)



## **AGENDA – SQL SERVER DOCKER IMAGE**

1) SEARCH - <https://hub.docker.com/r/microsoft/mssql-server>



[Explore](#) / [microsoft](#) / mssql-server



## Microsoft SQL Server - Ubuntu based images Verified Publisher

mssql/server

By [Microsoft](#) · Updated 11 months ago

Official images for Microsoft SQL Server based on Ubuntu

☆158

### Overview

## Featured Tags

- 2022-latest  
`docker pull mcr.microsoft.com/mssql/server:2022-latest`
- 2019-latest  
`docker pull mcr.microsoft.com/mssql/server:2019-latest`
- 2017-latest  
`docker pull mcr.microsoft.com/mssql/server:2017-latest`

2) `docker run -e "ACCEPT_EULA=Y" -e "MSSQL_SA_PASSWORD=Test@123" -p 1433:1433 -d mcr.microsoft.com/mssql/server:2022-latest`

`docker ps`

```
root@vmdocker:/home/azureuser/PyToDoBackendMonolith# docker run -e "ACCEPT_EULA=Y" -e "MSSQL_SA_PASSWORD=Test@123" -p 1433:1433 -d mcr.microsoft.com/mssql/server:2022-latest
Unable to find image 'mcr.microsoft.com/mssql/server:2022-latest' locally
2022-latest: Pulling from mssql/server
86752e6fd4ef: Pull complete
186a7fa30ef: Pull complete
d9043883789c: Pull complete
Digest: sha256:ea73825f3d88a23c355ac2f9fdc6bd960fec90171c12c572109b36a558f77bb8
Status: Downloaded newer image for mcr.microsoft.com/mssql/server:2022-latest
b0d3d3d50448fa82b882e22a3f5975a06d0d617d147f3582e5b901e66b1caee9
root@vmdocker:/home/azureuser/PyToDoBackendMonolith# ls
Dockerfile  Readme.md  app.py  requirements.txt
root@vmdocker:/home/azureuser/PyToDoBackendMonolith# docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                                NAMES
b0d3d3d50448   mcr.microsoft.com/mssql/server:2022-latest   "/opt/mssql/bin/laun..."   About a minute ago   Up About a minute   0.0.0.0:1433->1433/tcp, :::1433->1433/tcp   practical_shaw
ce5042b963fb   pytodo                                     "uvicorn app:app --h..."   38 minutes ago     Up 38 minutes     0.0.0.0:8080->8080/tcp, [::]:8080->8080/tcp   dazzling_chandrasekh
```

3) `docker logs <container id> = docker logs b0d3d3d50448`

4)



Learn Microsoft

<https://learn.microsoft.com> › [Learn](#) › [SQL](#) › [SQL Server](#) ›

## Docker: Install Containers for SQL Server on Linux

20 Dec 2024 — This quickstart shows how to use Docker to run the SQL Server Linux container images. You connect to a database and run a query.

[Configure and Customize SQL...](#)

[Password Policy](#)

5) docker exec -it sql1 "bash" = **docker exec -it b0d3d3d50448 "bash"**

**Class over as sql configuration is not easy**