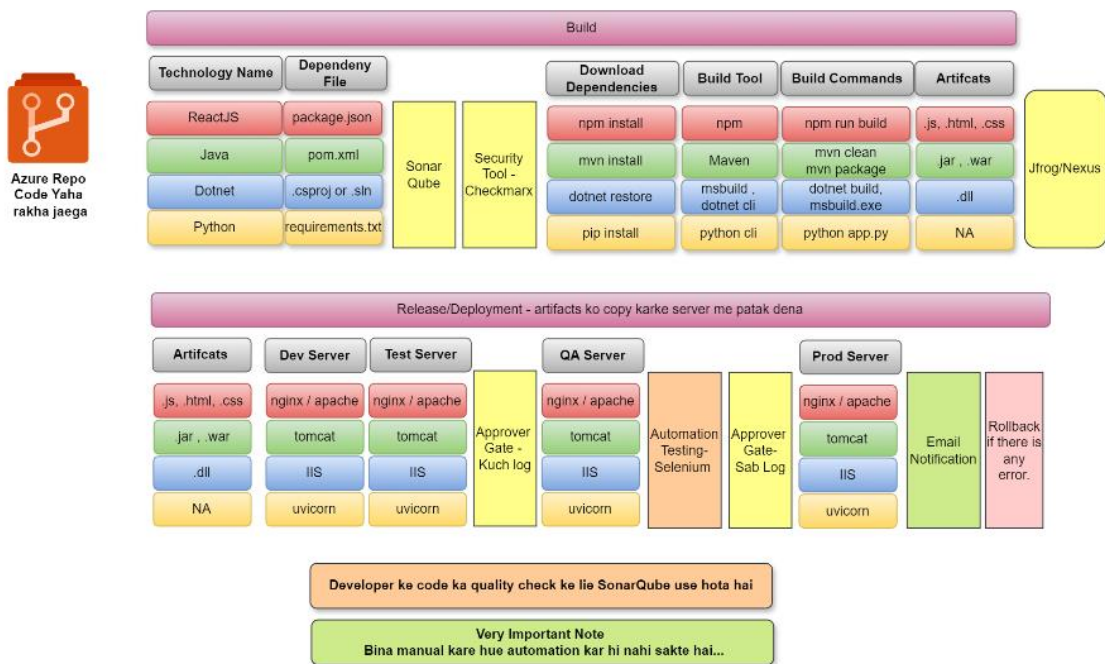
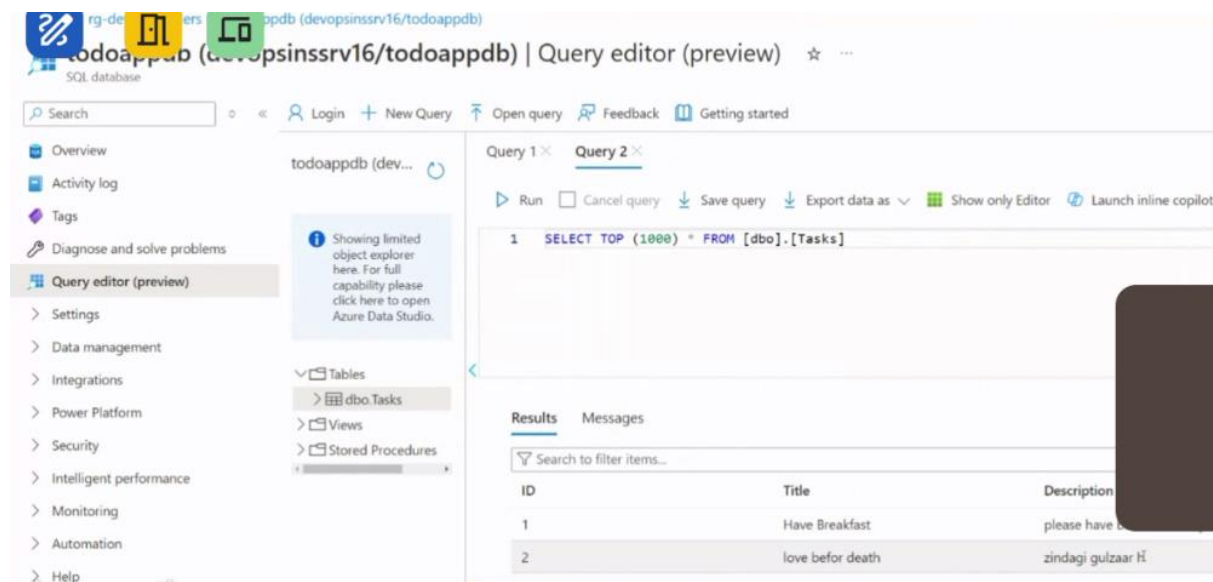


4 August 2024



1) Arte factory package management strategies – Packages are kept on jfrog and nexus tools from google. And developers use packages from jfrog and nexus tools instead of directly fetching packages from google, which increases the security and reduces vulnerability.

AGENDA – Create SQL DB



AGENDA – SETTING BACKEND CODE ON BACKEND VM

PyTodoBackendMonolith Public

 PyTodoBackendMonolith  repository serves as the backbone for the Todo application's backend, written in Python, embracing a monolithic architecture. It orchestrates data storage, executes busines...

python api monolithic todoapp

 Python  8  17 Updated on Jun 30

2) Now we have to put backend code on backend VM

3) ssh into backend vm and just work as per read me file

4) **SEARCH** - How to Install Python Pip on Ubuntu 20.04 and then git clone the repository

<https://linuxize.com/post/how-to-install-pip-on-ubuntu-20.04/>

```
devopsadmin@backendvm:~$ pip3 --version
pip 20.0.2 from /usr/lib/python3/dist-packages/pip (python 3.8)
devopsadmin@backendvm:~$
```

Step 1: Clone the Repository

Clone the application's source code from your version control system or download it as a zip archive and extract it to your local machine.

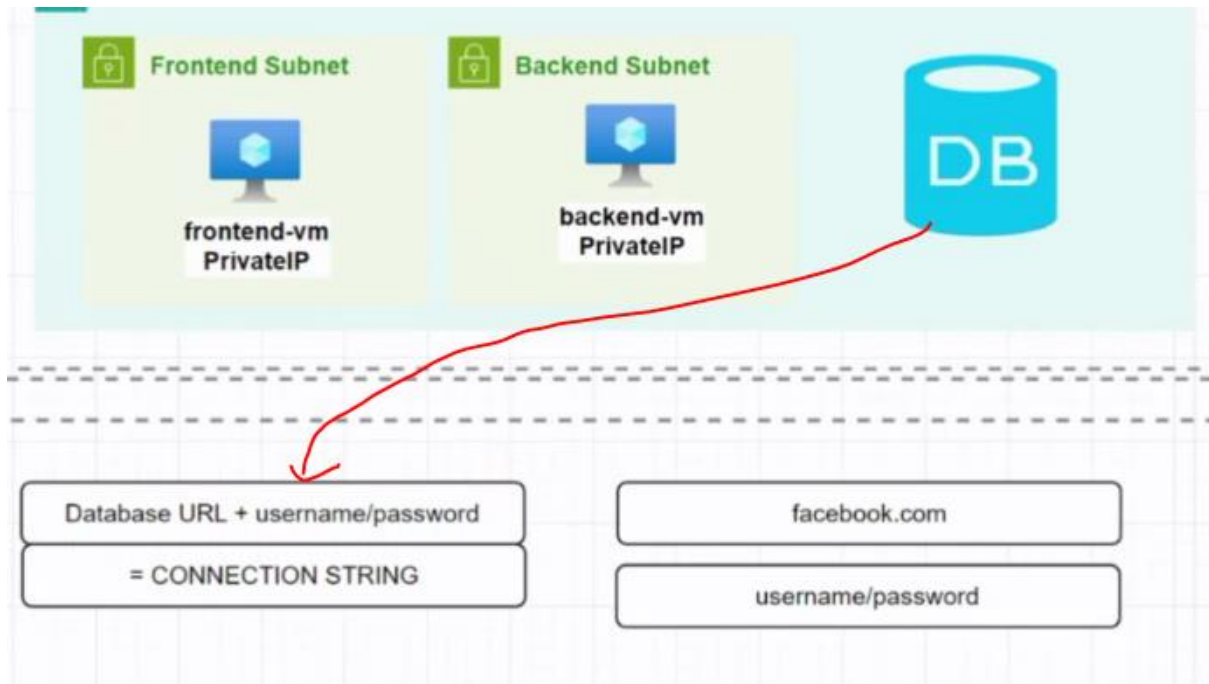
```
git clone <repository_url>
cd <repository_directory>
```



ls

cd PyTodoBackendMonolith\$

5) Connection string = username + password for database



6) **nano app.py**

Update connection string version as 17 and username and password also as given in code

7) Run given commands

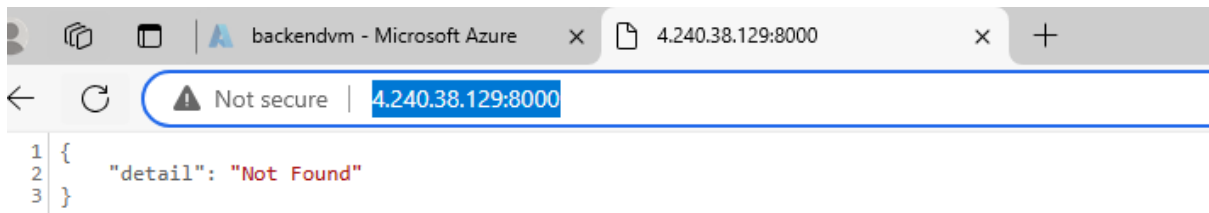
Step 3: Run Below Commands to make the application running

To Run the Application, open a terminal, navigate to the project directory, and run the following command:

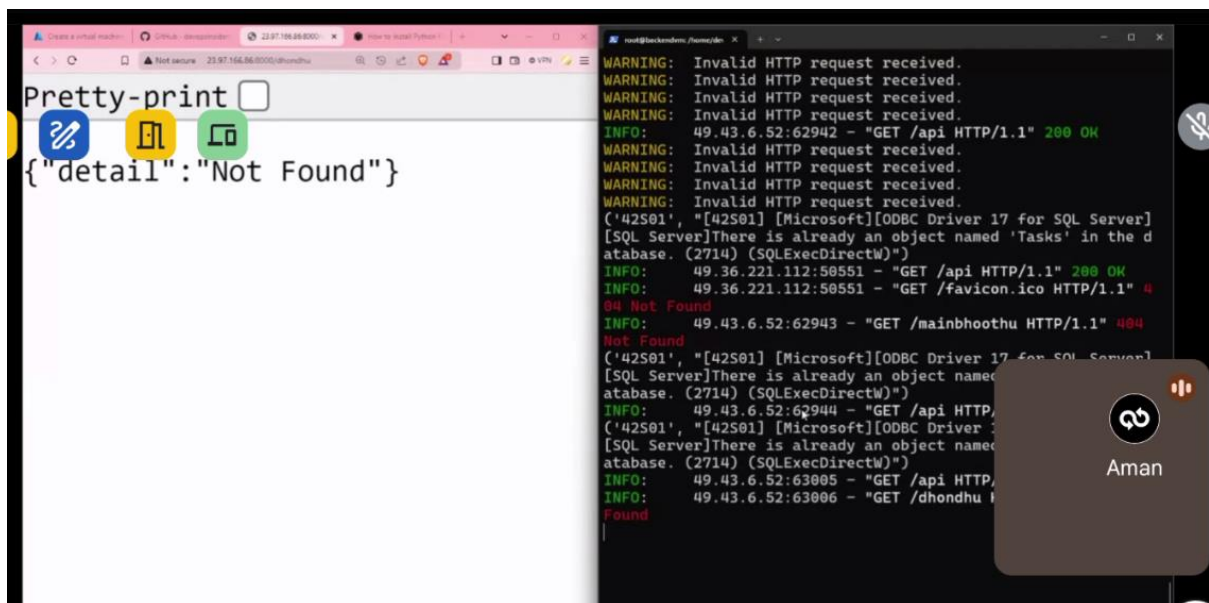
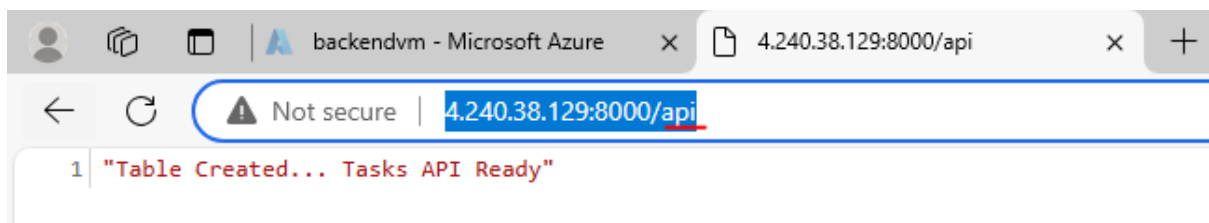
```
sudo su
apt-get update && apt-get install -y unixodbc unixodbc-dev
curl https://packages.microsoft.com/keys/microsoft.asc | apt-key add -
curl https://packages.microsoft.com/config/debian/10/prod.list > /etc/apt/sources.list.d/mssql-release.
apt-get update
ACCEPT_EULA=Y apt-get install -y msodbcsql17
pip install -r requirements.txt
uvicorn app:app --host 0.0.0.0 --port 8000
```

8) Run ip of backend vm and port of db

4.240.38.129:8000





4.240.38.129:8000/api



+++++ AGENDA – SETTING FRONTEND CODE ON FRONTEND VM +++++

ReactTodoUIMonolith Public

 ReactTodoUIMonolith  Repository is housing the user interface code developed in React for the Todo application's monolithic architecture. This repository is responsible for managing the frontend...

reactjs

monolithic

todoapp

JavaScript ☆ 6 🍴 26 Updated on Jan 4

- 1) ssh into frontend vm
- 2) Read readme file
- 3) **curl -s https://deb.nodesource.com/setup_16.x | sudo bash**
- sudo apt install nodejs -y**
- 4) git clone <https://github.com/devopsinsiders/ReactTodoUIMonolith.git>

ls

```
PyTodoBackendMonolith ReactTodoUIMonolith
azureuser@vm-docker:~$ ls
PyTodoBackendMonolith ReactTodoUIMonolith
azureuser@vm-docker:~$
```

cd ReactTodoUIMonolith

ls

```
azureuser@vm-docker:~$ cd ReactTodoUIMonolith
azureuser@vm-docker:~/ReactTodoUIMonolith$ ls
README.md build package-lock.json package.json public src
azureuser@vm-docker:~/ReactTodoUIMonolith$
```

cd src

ls

nano TodoApp.js

Now in below field give the url of backend.

```
const API_BASE_URL = 'http://4.240.38.129:8000/api';
```

Ctrl+s

Ctrl+x

npm install – this command actually downloads dependencies from package.json on our vm or computer. After this node_modules folder comes in which dependencies are downloaded

```
-rw-rw-r-- 1 devopsadmin devopsadmin 294 Aug  4 05:28 theme.js
devopsadmin@frontendvm:~/ReactTodoUIMonolith/src$ nano TodoApp.js
devopsadmin@frontendvm:~/ReactTodoUIMonolith/src$ npm install
( [REDACTED] ) : reify: vary: http fetch GET 200 https://registry.npmjs.org/vary/-/vary-1.1.2
devopsadmin@frontendvm:~/ReactTodoUIMonolith/src$ cd ..
devopsadmin@frontendvm:~/ReactTodoUIMonolith$ ls
README.md  build  node_modules  package-lock.json  package.json  public  src
devopsadmin@frontendvm:~/ReactTodoUIMonolith$
```

ls

cd build

npm run build – By this our code will be converted into html, css, javascript code

```
devopsadmin@frontendvm:~/ReactTodoUIMonolith$ ls
README.md  build  node_modules  package-lock.json  package.json  public  src
devopsadmin@frontendvm:~/ReactTodoUIMonolith$ npm run build
```

```
devopsadmin@frontendvm:~/ReactTodoUIMonolith$ ls
README.md  build  node_modules  package-lock.json  package.json  public  src
devopsadmin@frontendvm:~/ReactTodoUIMonolith$ cd bui
-bash: cd: bui: No such file or directory
devopsadmin@frontendvm:~/ReactTodoUIMonolith$ cd build/
devopsadmin@frontendvm:~/ReactTodoUIMonolith/build$ ls
asset-manifest.json  devopsinsiderslogo.png  index.html  logo512.png  robots.txt
background.jpg       favicon.ico               logo192.png  manifest.json  static
devopsadmin@frontendvm:~/ReactTodoUIMonolith/build$ cd static/
devopsadmin@frontendvm:~/ReactTodoUIMonolith/build/static$ ls
css  js
devopsadmin@frontendvm:~/ReactTodoUIMonolith/build/static$
```

5) Now our artefacts are ready so now putting these on nginx webserver. So the code which is in build folder has to be put in nginx server

```
devopsadmin@frontendvm:~/ReactTodoUIMonolith$ ls
README.md  build  node_modules  package-lock.json  package.json  public  src
devopsadmin@frontendvm:~/ReactTodoUIMonolith$ cd build
devopsadmin@frontendvm:~/ReactTodoUIMonolith/build$ ls
asset-manifest.json  background.jpg  devopsinsiderslogo.png  favicon.ico  index.html  logo192.png  logo512.png  manifest.json  robots.txt  static
devopsadmin@frontendvm:~/ReactTodoUIMonolith/build$
```

sudo systemctl status nginx

```
devopsadmin@frontendvm:~/ReactTodoUIMonolith/src$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2024-10-09 08:30:24 UTC; 5h 29min ago
     Docs: man:nginx(8)
   Main PID: 2316 (nginx)
    Tasks: 2 (limit: 4081)
   Memory: 5.0M
   CGroup: /system.slice/nginx.service
           └─2316 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
             └─2317 nginx: worker process

Oct 09 08:30:24 frontendvm systemd[1]: Starting A high performance web server and a reverse proxy server...
Oct 09 08:30:24 frontendvm systemd[1]: Started A high performance web server and a reverse proxy server.
devopsadmin@frontendvm:~/ReactTodoUIMonolith/src$
```

sudo cp -r * /var/www/html/- putting all files from build folder into nginx

sudo systemctl restart nginx

98.70.107.15:80

[DevOpsInsiders](#)

