Containerize the web application by Using AWS, GitHub and Docker

Class 18

31/5/2025

Acknowledgement

The series of the IT & Japanese language course is Supported by AOTS and OEC.

Ministry of Economy, Trade and Industry

Overseas Employment Corporation

What you have Learnt Last Week

We were focused on following points.

- Usage of control and loop flow statement
- Performing Linear Algebra in Numpy
- Why Requirement Analysis is so important in the process?
- Software development Life cycle
- Importance of Security compliance
- Basic Linux and its networking Commands.
- Introduction of Bash Scripting
- Introduction of docker and docker compose

What you will Learn Today

We will focus on following points.

1. Clone the GitHub repository and deploy web application on AWS server by running command
2. Dockerized the web application and deploy on AWS server
3. Create docker compose file for the web application and automatically deploy on AWS server
4. Quiz
5. Q&A Session

Deploying Live Project on AWS

Learn how to deploy a Django project on AWS

Using:

- EC2,
- GitHub,
- Python,
- and DevOps tools.

Python for DevOps

Why Python Matters

- ✅ Python powers automation, scripting, and backend
- ✅ Django framework used for this project
- ✅ Understand managing Python environments

Project Setup with GitHub

Setup the Code on Your Local PC

Clone the project in VS Code: https://github.com/shreys7/django-todo

Install virtual environment: `sudo apt install python3-virtualenv`

Make Virtual Environment: python3 -m venv ~/myenv

Activate The Environment: source ~/myenv/bin/activate

Go to project folder: cd django-todo

Install Django: pip install django

How To Run The Django Project

Run the project on local

Run The Following Commands:

Python manage.py makemigrations

Python manage.py migrate

Python manage.py runserver

Your Project will start running on local host

Deploying on AWS EC2

Setting Up Cloud Infrastructure

- ✅ Launch AWS EC2 instance (Linux)
- ✅ SSH into instance securely
- ✅ Clone the Django project from GitHub

For Creating Docker File

4 Steps Follow to automate:

- Apt-get update
- Install python
- Code
- Run

Running Django on EC2

Getting the App Live

- **Make a folder:** mkdir projects
- **Go into folder:** cd projects/
- **Clone the app:** git clone (https://github.com/shreys7/django-todo.git)
- **Go into project:** cd django-todo/
- **Install Django:** pip install django

If not work then

pip3 install django or

sudo apt install python3-pip

Manual Deploy on EC2

Run the Following Commands:

- python3 manage.py migrate

- Python mnage.py runserver

Starting development Server at: http://127.0.0.1:8000/

Note: 127.0.0.1 is a local host

To Give Global Access to App

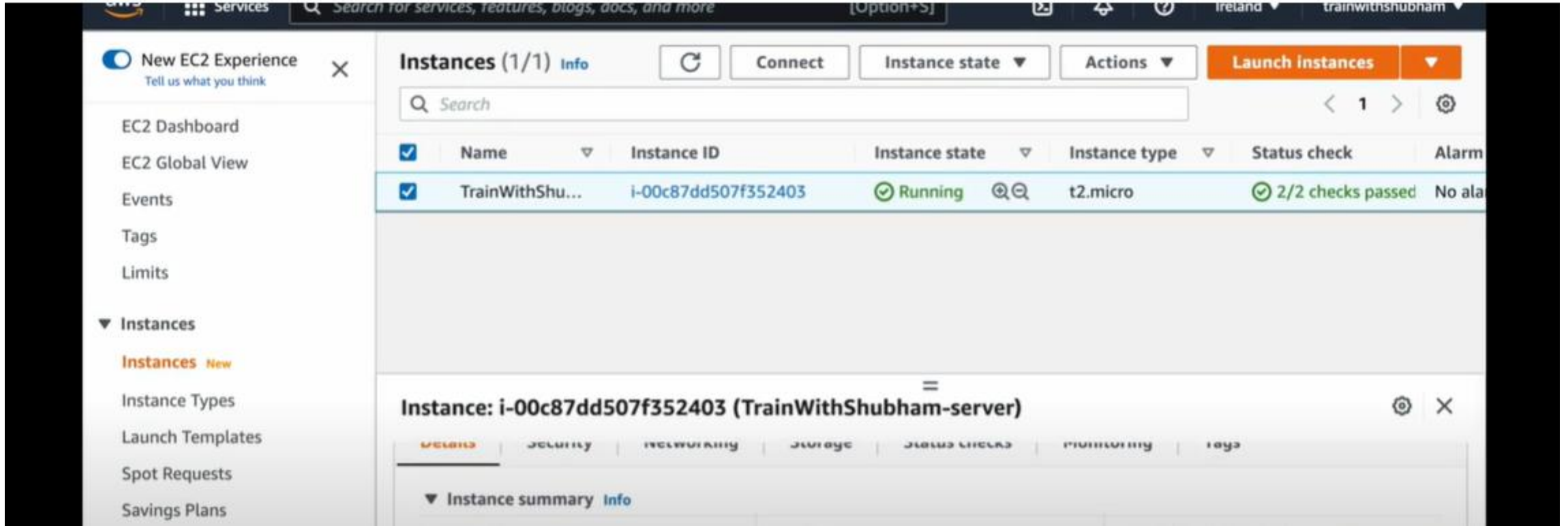
Makes the app accessible to the world

- python mnage.py runserver 0.0.0.0:8001
- But you Can’t Access the App
- The EC2 instance firewall (security group) is blocking the port.
- So we have to add a rule to allow traffic on port 8001 from 0.0.0.0/0.

Understanding EC2 Security Group Rules

How to Add Security Rule:

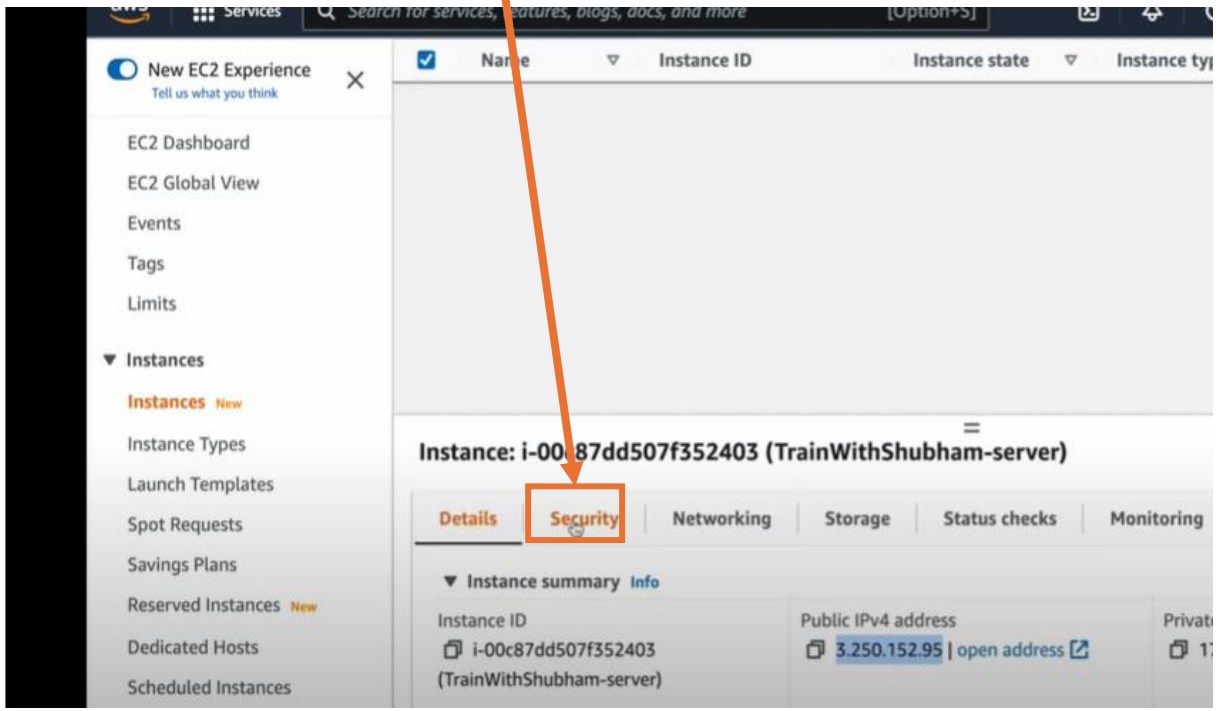
Step1: Go to AWS EC2



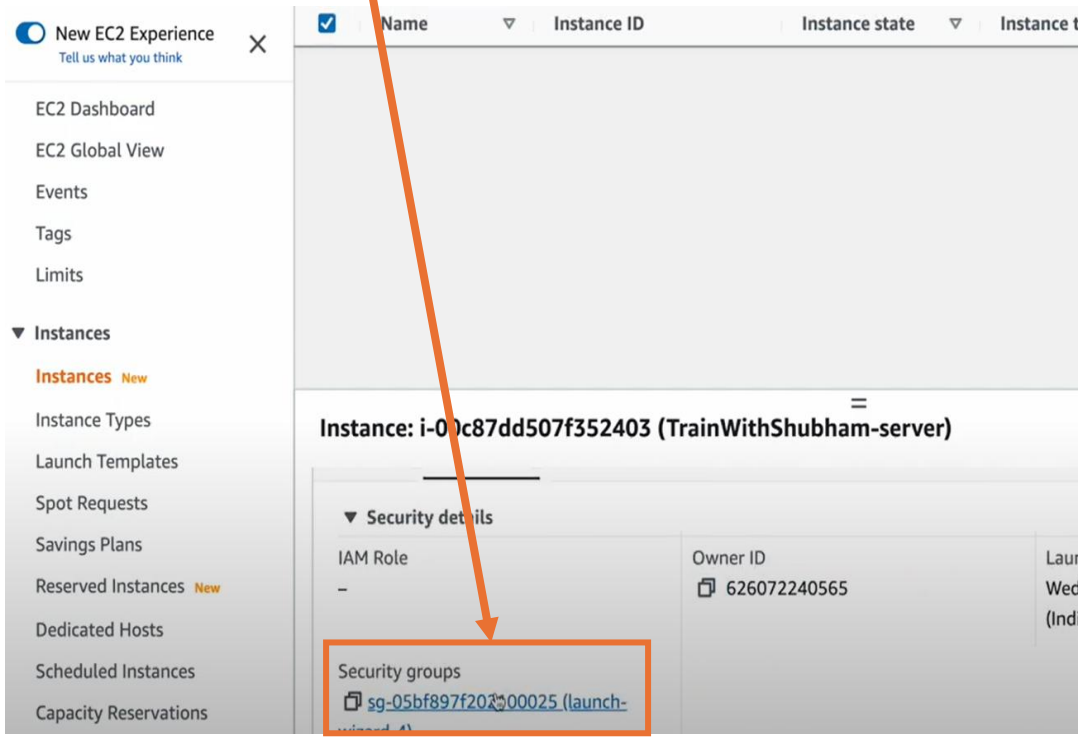
Understanding EC2 Security Group Rules

How to Add Security Rule:

Step2: Go to Security



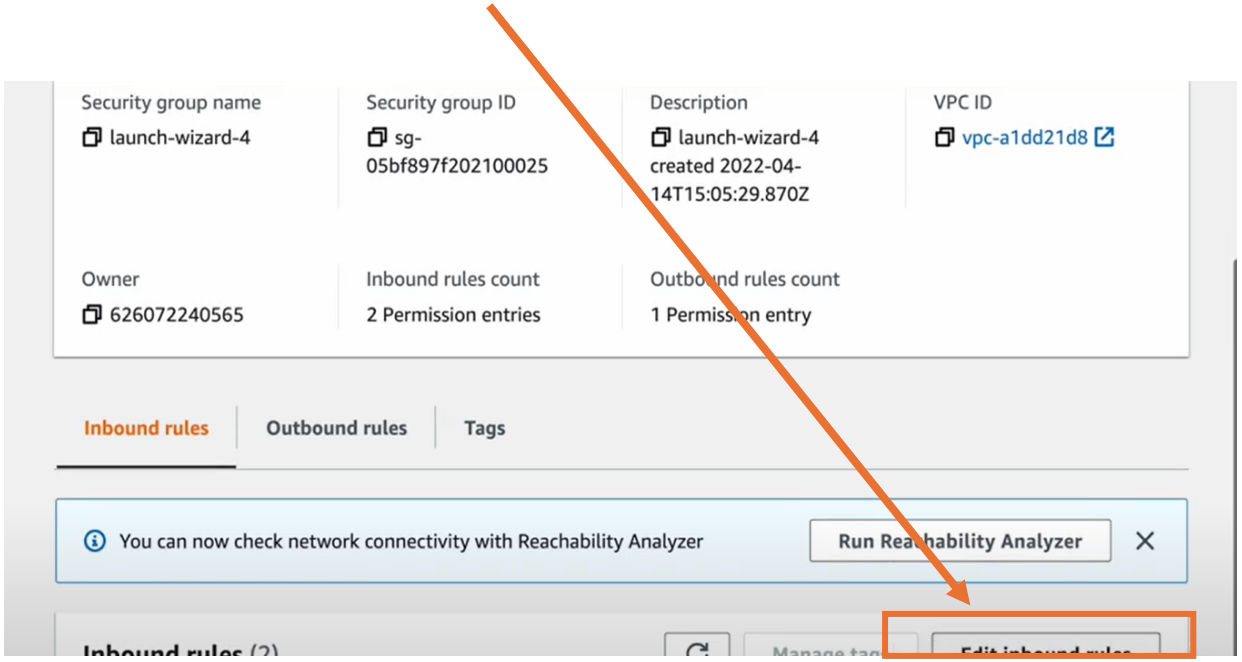
Step3: Go to Group



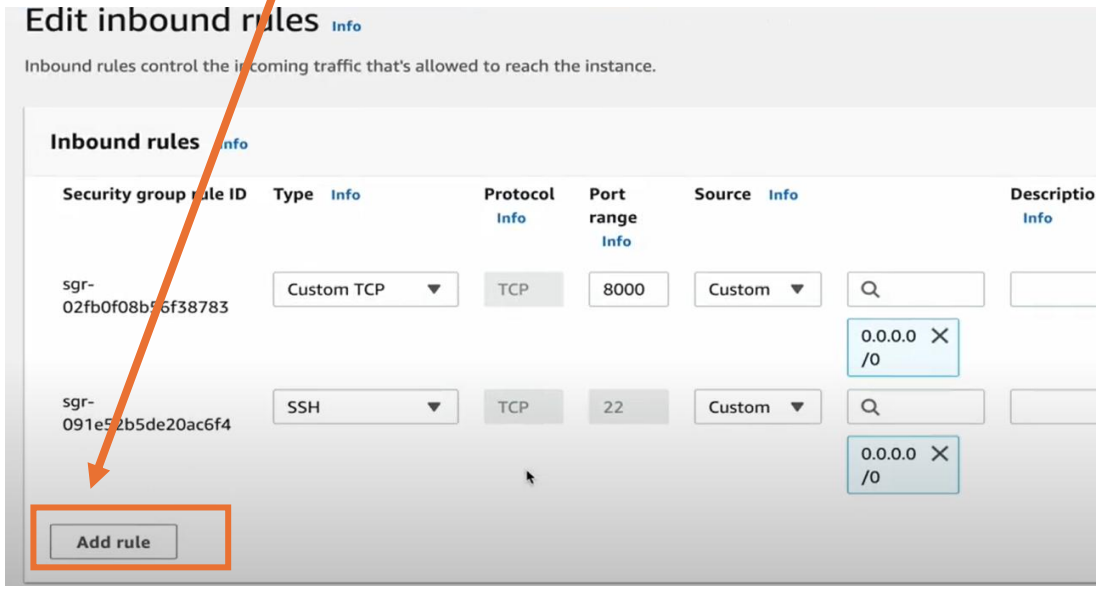
Understanding EC2 Security Group Rules

How to Add Security Rule:

Step4: Edit Inbound rules



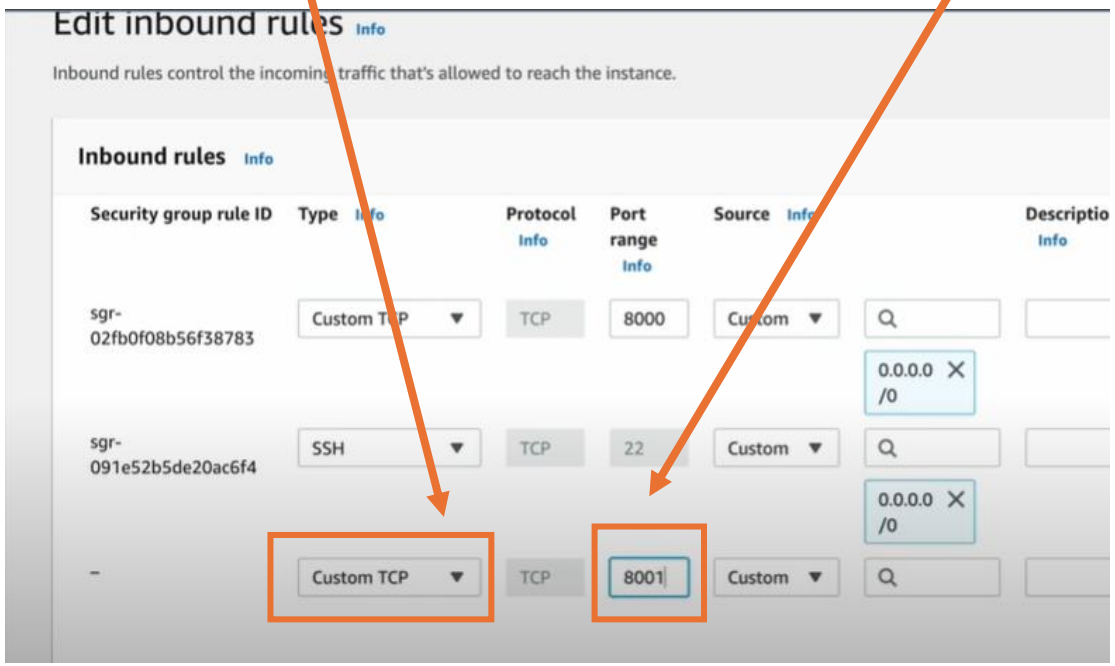
Step5: Click on Add Rule

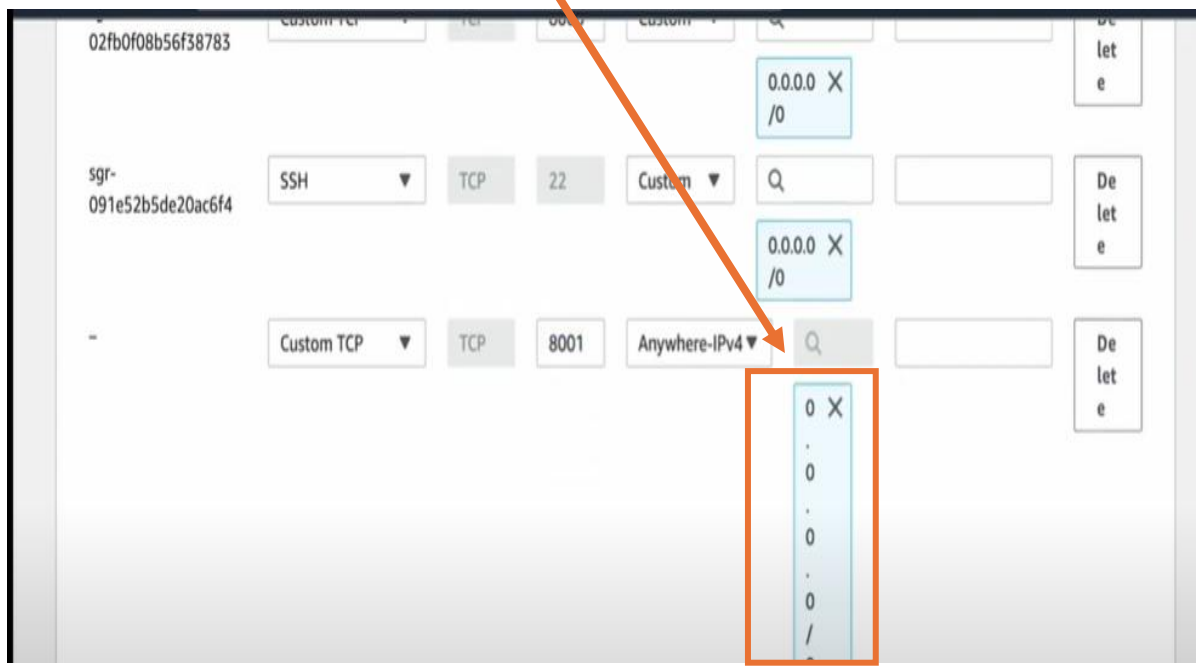


Understanding EC2 Security Group Rules

How to Add Security Rule:

Step6: Select Type (Custom TCP) Select Port Range 8001 and 0.0.0.0/0





Save The Changes After adding this 8001 can be accessible from anywhere

Disallowed Host Error Explained

Why This Happens?

- Django has a security setting called ALLOWED_HOSTS

  which restricts incoming requests to only approved hosts.

- By default, it only allows localhost or 127.0.0.1.

  When accessed by public IP, Django blocks it.

Disallowed Host Error Explained

How To Fix That

1. Go to vi todoApp/setting.py

2. Find the line: ALLOWED_HOSTS = []

3.Change it to: ALLOWED_HOSTS = ['<your-ec2-public-ip>', 'localhost’] or

4. ALLOWED_HOSTS = ['*’]

5. Now run your App (python3 manage.py runserver 0.0.0.0:8001)

6. It will Start Runing

Create Docker File

Automate Through Docker Which We manually do

Vi Dockerfile

- FROM pyhton:3
- RUN pip install django==3.2
- COPY . .
- RUN pyhton manage.py migrate
- CMD [“python”, “manage.py”, ”runserver, “0.0.0.0:8001”]

Build The Docker Image

Run this in terminal:

- sudo docker build . –t todo-app
- sudo docker run –p 8001:8001 (container id)

✅ **Summary:**

Dockerfile automates creating a ready-to-run container image for your project — no manual setup needed!

Assignment

Assignment

Commands

Upload this whole project into your GitHub profile

with proper documentation

Quiz Section

Quiz

Everyone student should click on submit button before time ends otherwise MCQs will not be submitted

[Guidelines of MCQs]

1. There are 20 MCQs
2. Time duration will be 10 minutes
3. This link will be share on 12:25pm (Pakistan time)
4. MCQs will start from 12:30pm (Pakistan time)
5. This is exact time and this will not change
6. Everyone student should click on submit button otherwise MCQs will not be submitted after time will finish
7. Every student should submit Github profile and LinkedIn post link for every class. It include in your performance

Assignment

Assignment should be submit before the next class

[Assignments Requirements]

1. Create a post of today's lecture and post on LinkedIn.
2. Make sure to tag @Plus W @Pak-Japan Centre and instructors LinkedIn profile
3. Upload your code of assignment and lecture on GitHub and share your GitHub profile in respective your region group WhatsApp group
4. If you have any query regarding assignment, please share on your region WhatsApp group.
5. Students who already done assignment, please support other students

Q&A Session

ありがとうございます。
Thank you.
شکریا

For the World with Diverse Individualities