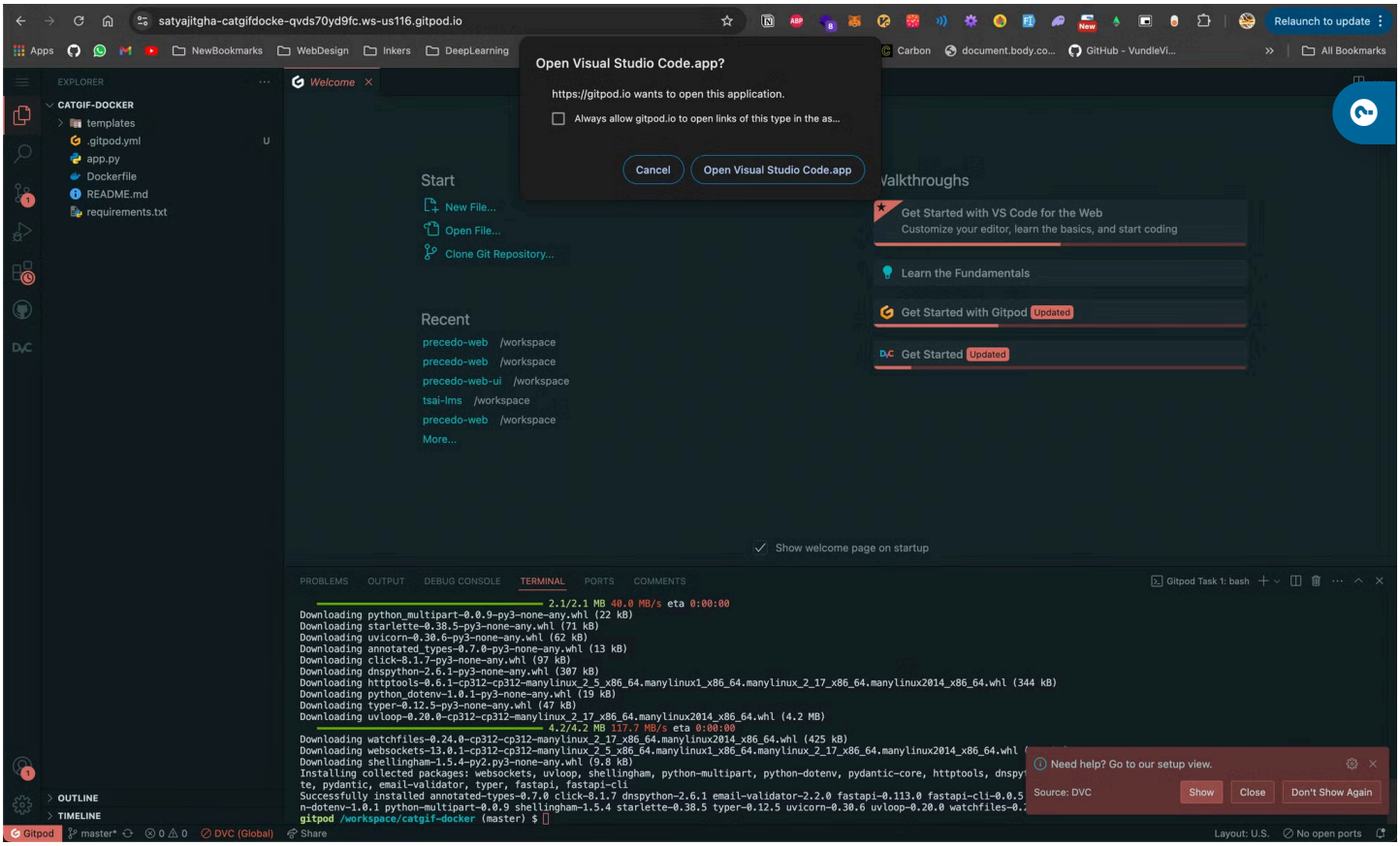
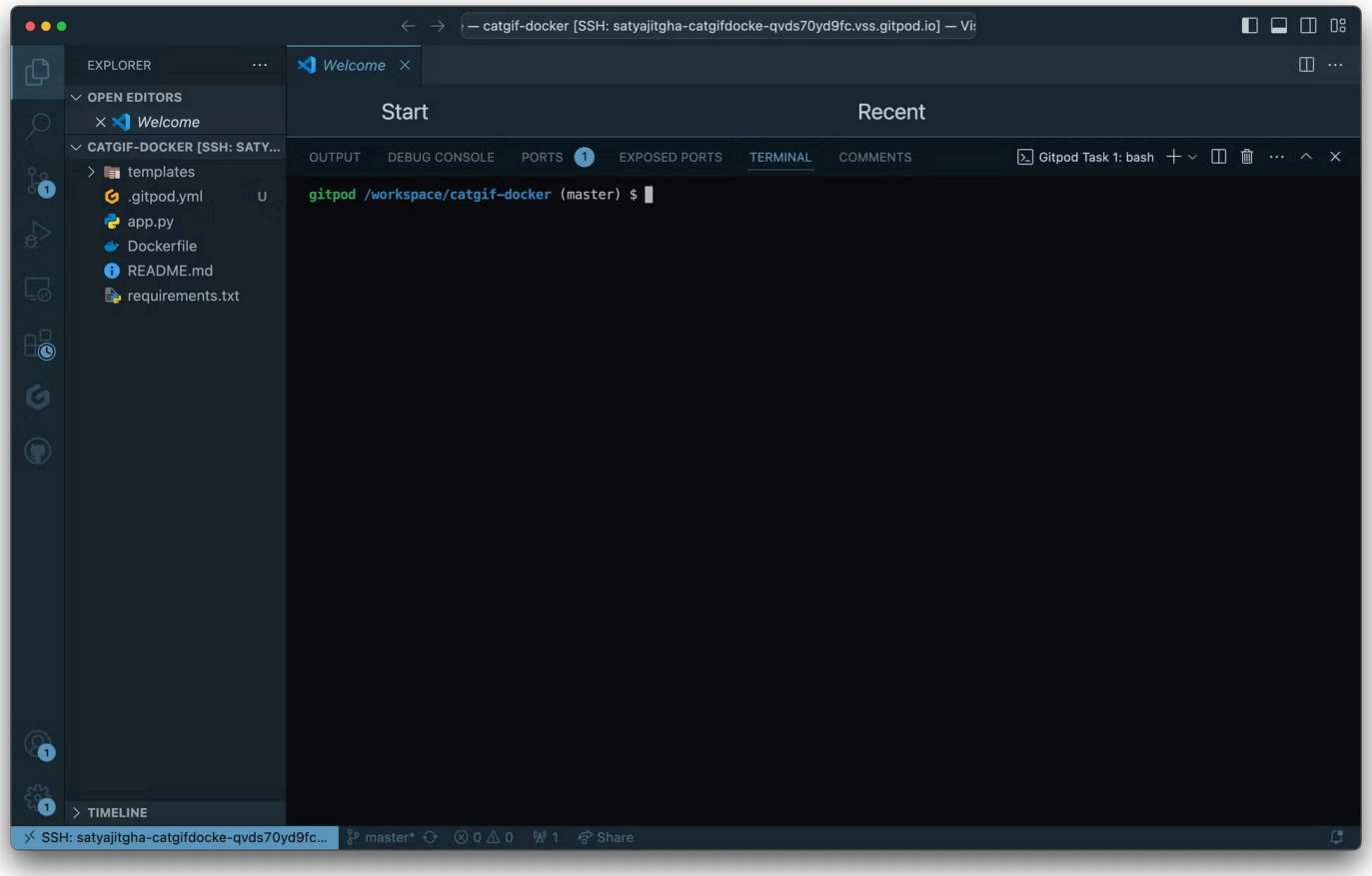
Session 03 - Docker - II

Start Assignment

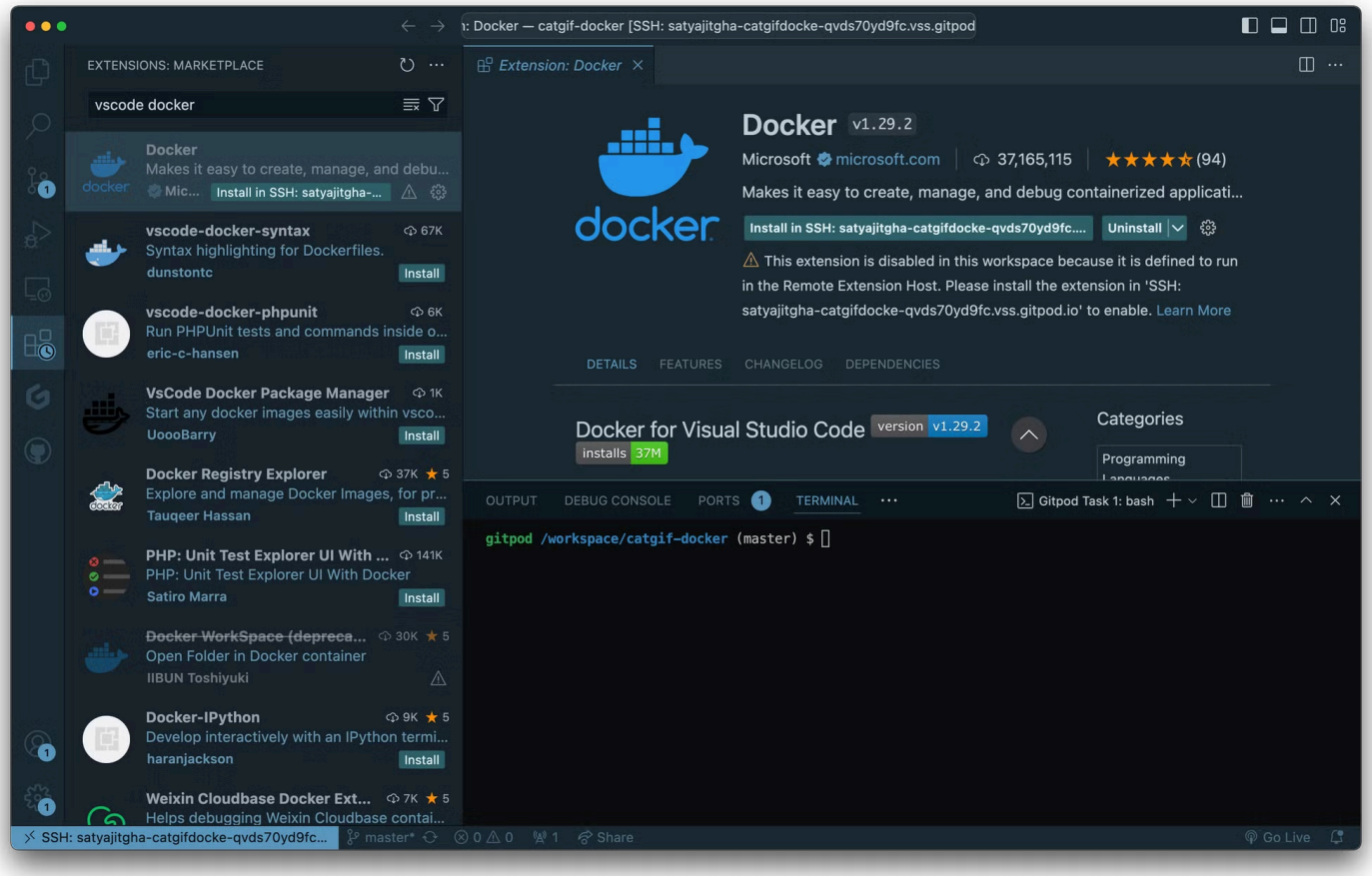
- Due Saturday by 23:59
- Points 500
- Submitting a text entry box

Docker VSCode





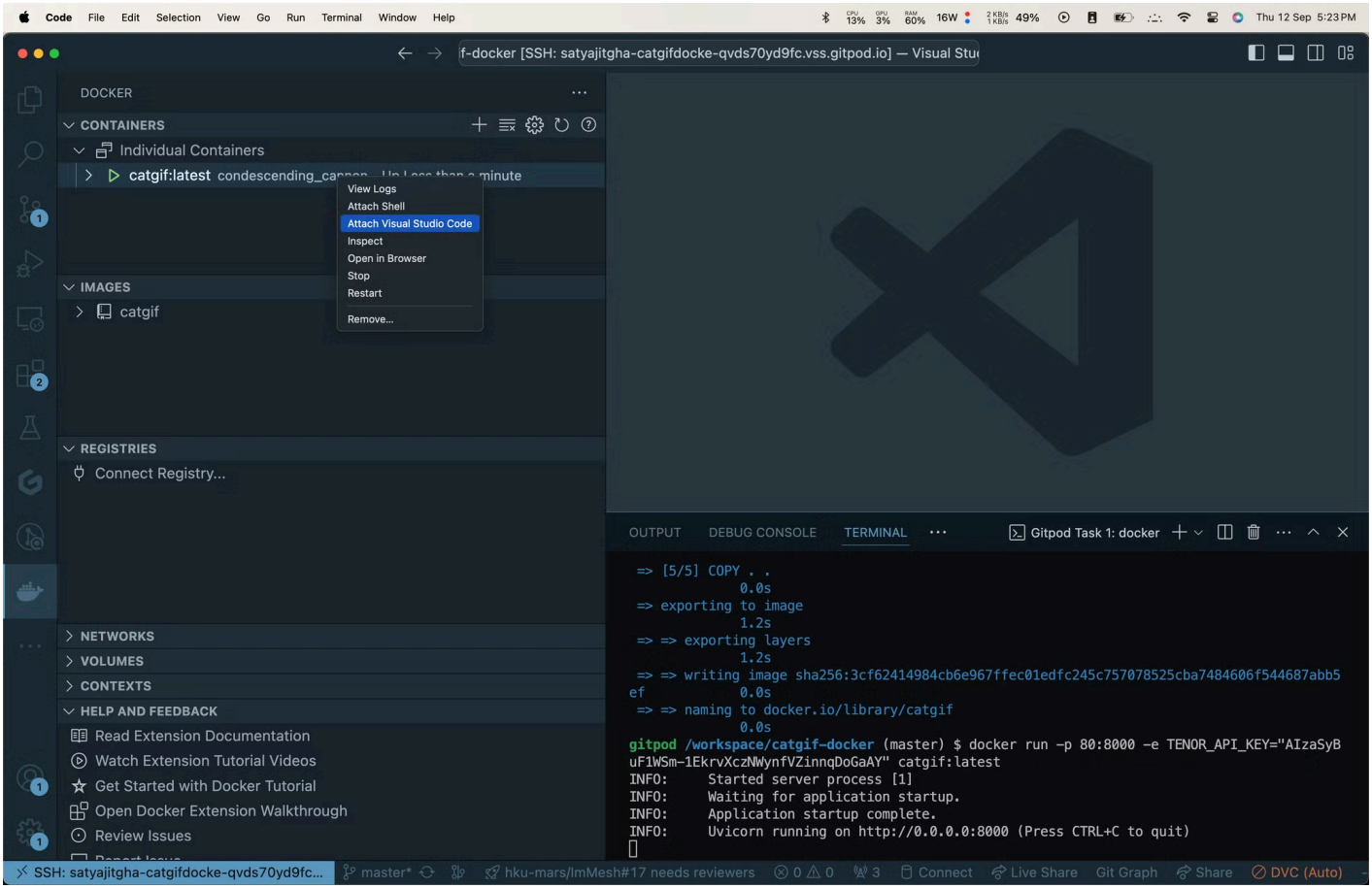
Install the docker extension in VSCode `ms-azuretools.vscode-docker`

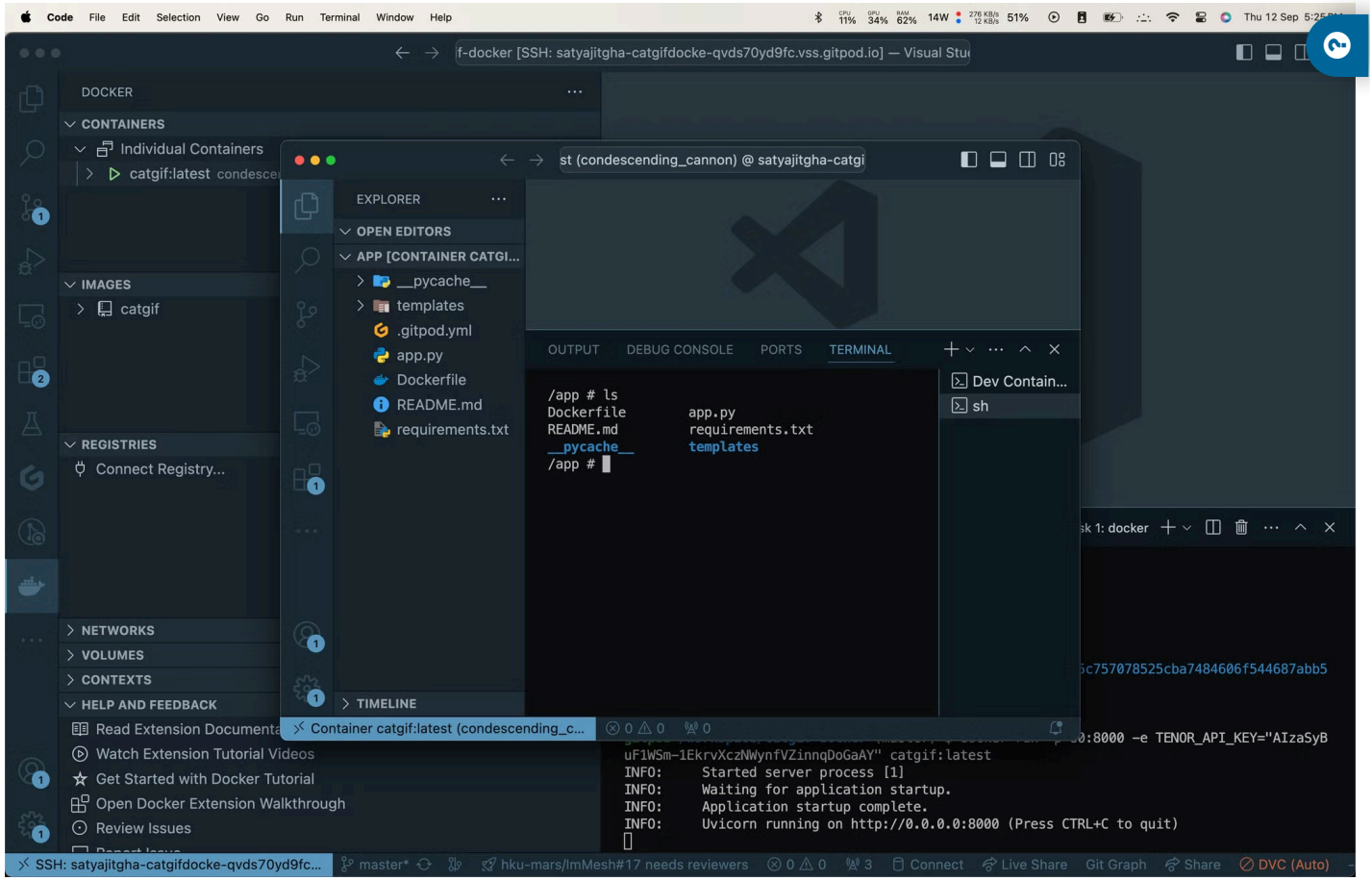



```
git clone <https://github.com/satyajitghana/catgif-docker>
```

```
docker build -t catgif .
```

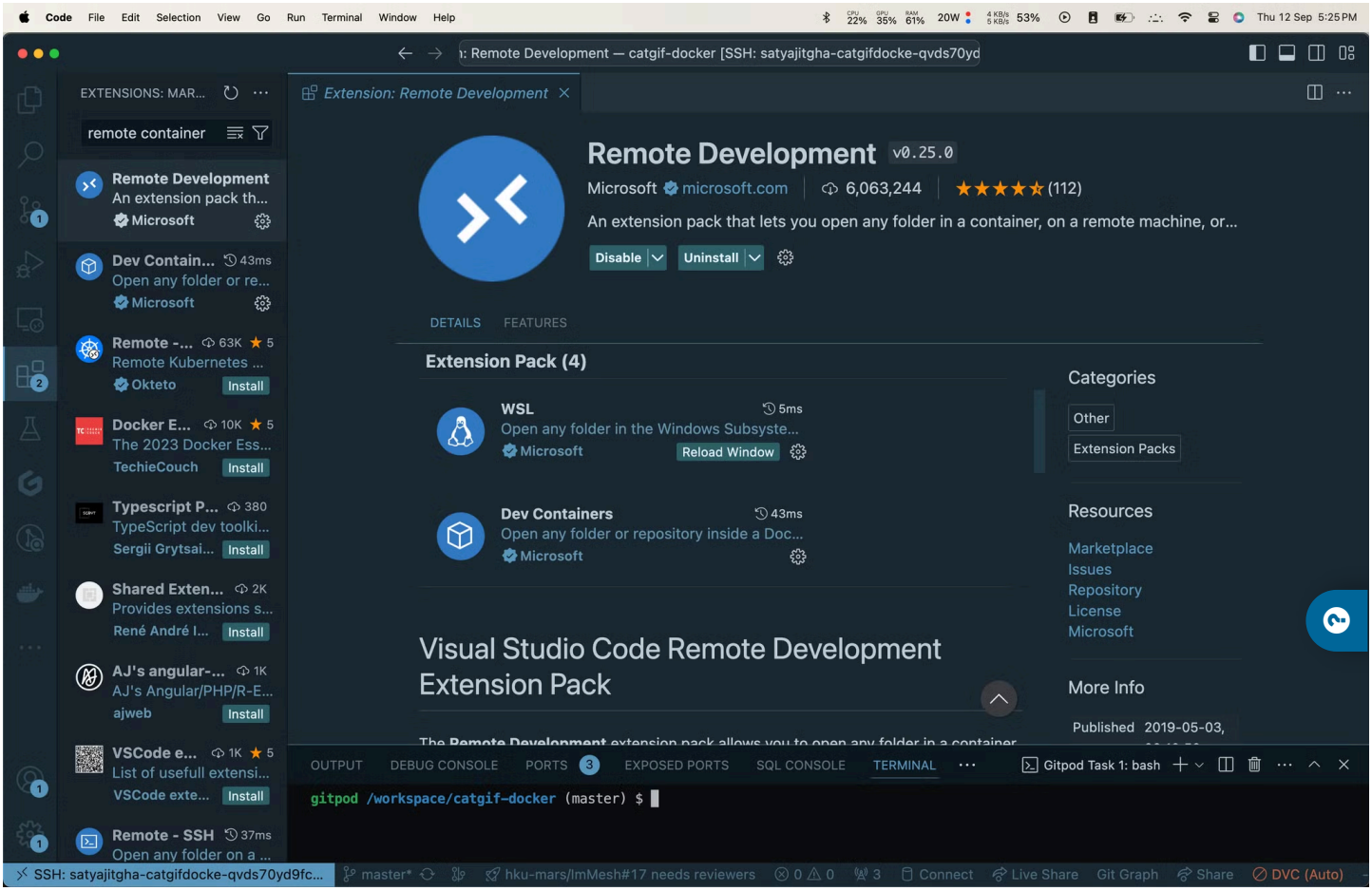
```
docker run -p 80:8000 -e TENOR_API_KEY="AIzaSyBuF1WSm-1EkrvXczNWynfVZinnqDoGaAY" catgif:latest
```

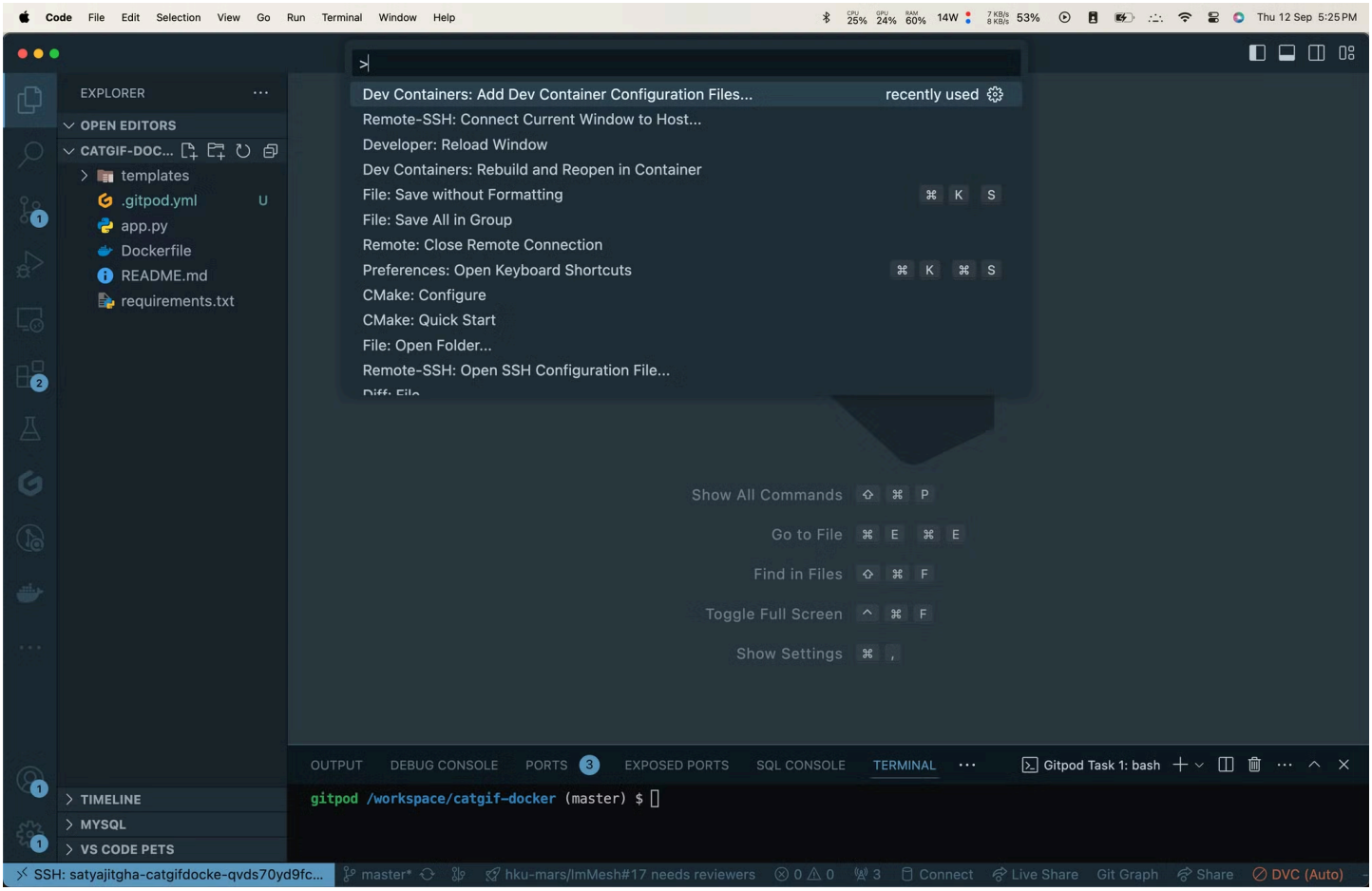

DOCKER
CONTAINERS
Individual Containers
catgif:latest condescending_cannon
View Logs
Attach Shell
Attach Visual Studio Code
Inspect
Open in Browser
Stop
Restart
Remove...
IMAGES
catgif
REGISTRIES
Connect Registry...
NETWORKS
VOLUMES
CONTEXTS
HELP AND FEEDBACK
Read Extension Documentation
Watch Extension Tutorial Videos
Get Started with Docker Tutorial
Open Docker Extension Walkthrough
Review Issues
OUTPUT
DEBUG CONSOLE
TERMINAL
Gitpod Task 1: docker
=> [5/5] COPY . .
0.0s
=> exporting to image
1.2s
=> => exporting layers
1.2s
=> => writing image sha256:3cf62414984cb6e967ffec01edfc245c757078525cba7484606f544687abb5ef
0.0s
=> => naming to docker.io/library/catgif
0.0s
gitpod /workspace/catgif-docker (master) $ docker run -p 80:8000 -e TENOR_API_KEY="AIzaSyBuF1WSm-1EkrvXczNWynfVZinnqDoGaAY" catgif:latest
INFO: Started server process [1]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
SSH: satyajitgha-catgifdocke-qvds70yd9fc...
master*
hku-mars/ImMesh#17 needs reviewers
Connect
Live Share
Git Graph
Share
DVC (Auto)


DOCKER
CONTAINERS
Individual Containers
catgif:latest
IMAGES
catgif
REGISTRIES
Connect Registry...
NETWORKS
VOLUMES
CONTEXTS
HELP AND FEEDBACK
Read Extension Documenta
Watch Extension Tutorial Videos
Get Started with Docker Tutorial
Open Docker Extension Walkthrough
Review Issues
EXPLORER
OPEN EDITORS
APP [CONTAINER CATGI...
__pycache__
templates
.gitpod.yml
app.py
Dockerfile
README.md
requirements.txt
TIMELINE
OUTPUT
DEBUG CONSOLE
PORTS
TERMINAL
Dev Contain...
sh
/app # ls
Dockerfile app.py
README.md requirements.txt
__pycache__ templates
/app #
Container catgif:latest (condescending_c...
INFO: Started server process [1]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
SSH: satyajitgha-catgifdocke-qvds70yd9fc...
master*
hku-mars/ImMesh#17 needs reviewers
Connect
Live Share
Git Graph
Share
DVC (Auto)

DevContainer

Extension `ms-vscode-remote.remote-containers`

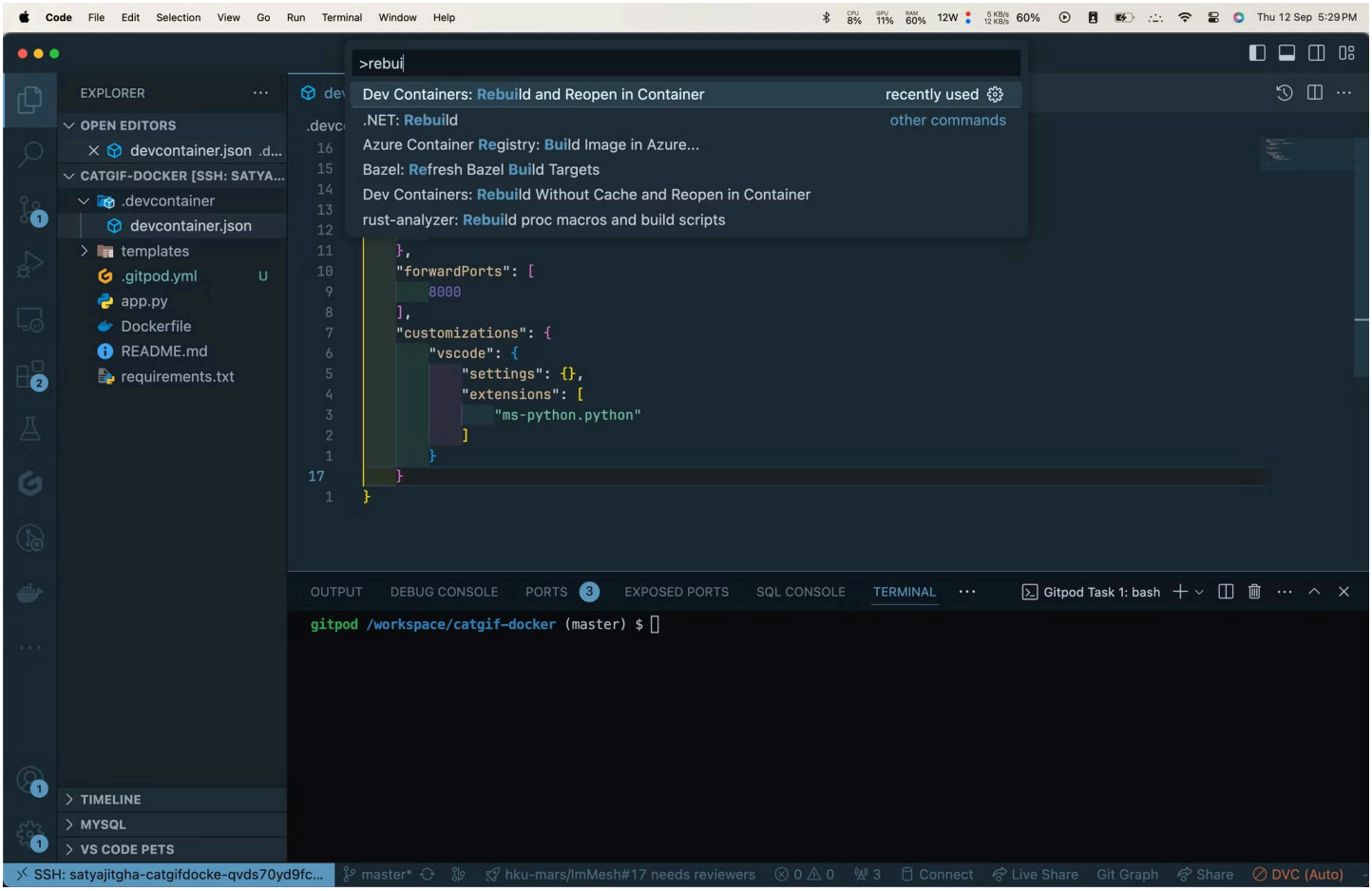


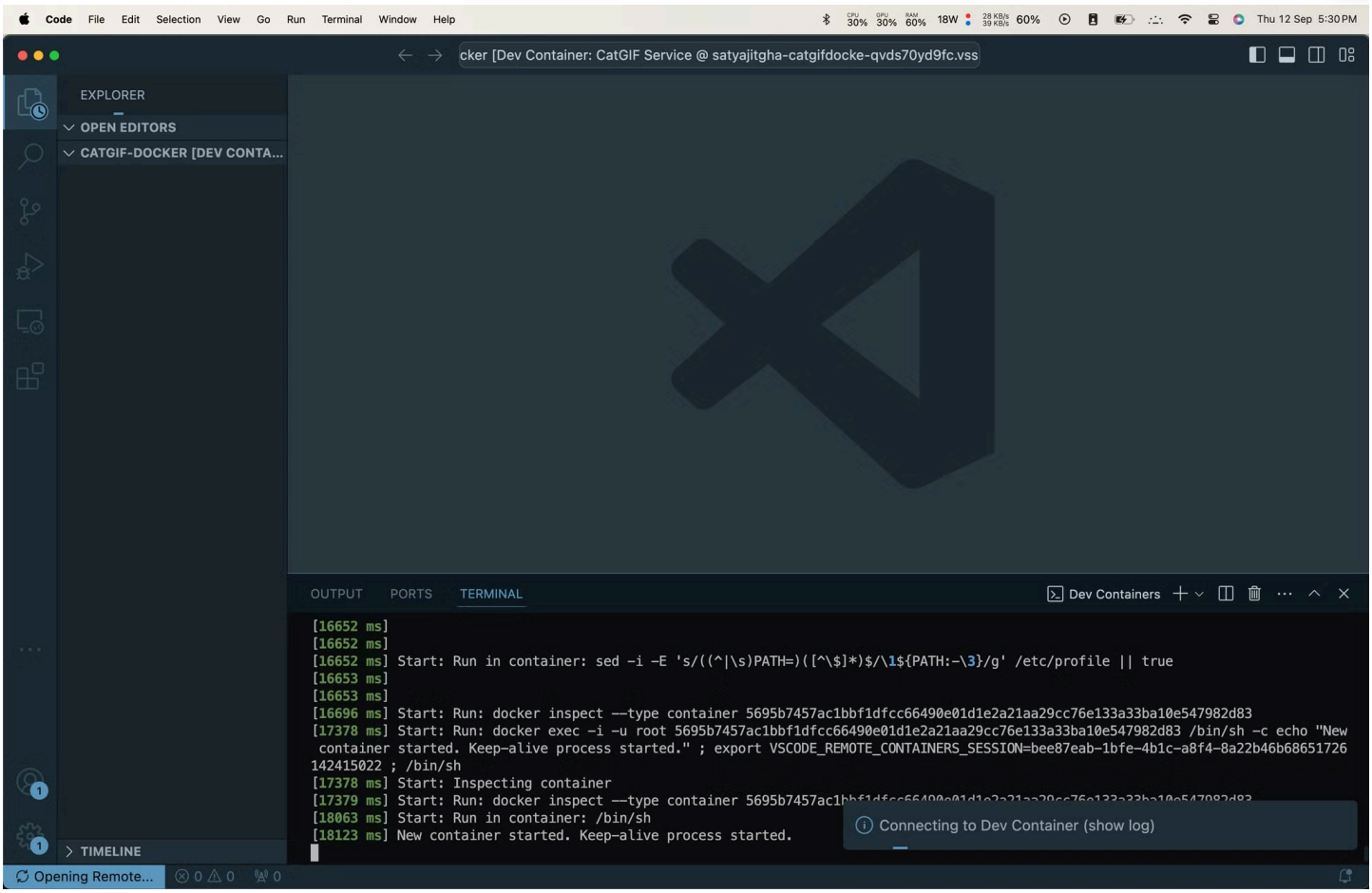


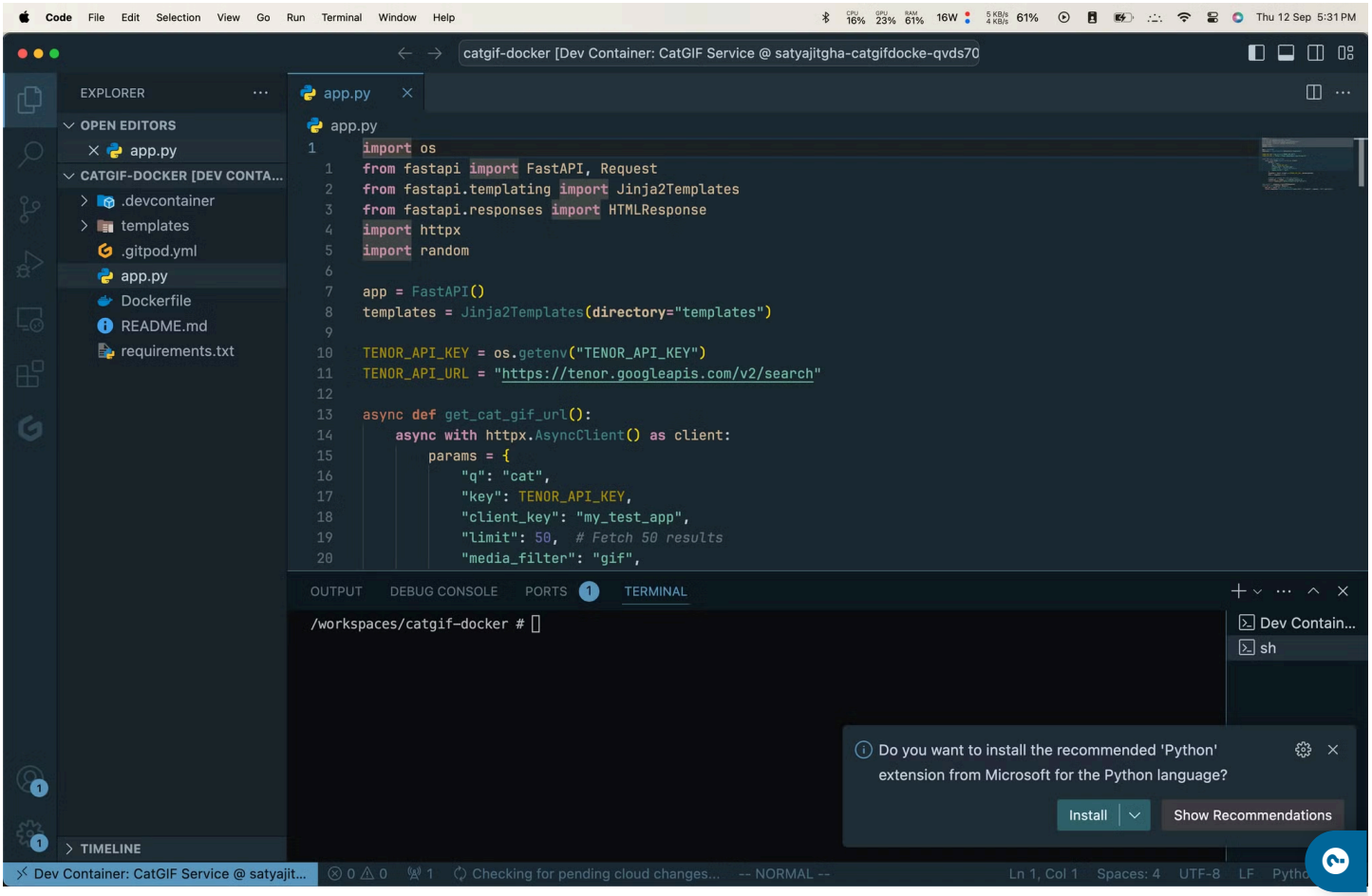

```
// For format details, see <https://aka.ms/devcontainer.json>. For config options, see the
// README at: <https://github.com/devcontainers/templates/tree/main/src/python>
{
	"name": "Python 3",
	// Or use a Dockerfile or Docker Compose file. More info: <https://containers.dev/guide/dockerfile>
	"image": "mcr.microsoft.com/devcontainers/python:1-3.11-bookworm"

	// Features to add to the dev container. More info: <https://containers.dev/features>.
	// "features": {},

	// Use 'forwardPorts' to make a list of ports inside the container available locally.
	// "forwardPorts": [],

	// Use 'postCreateCommand' to run commands after the container is created.
	// "postCreateCommand": "pip3 install --user -r requirements.txt",

	// Configure tool-specific properties.
	// "customizations": {},

	// Uncomment to connect as root instead. More info: <https://aka.ms/dev-containers-non-root>.
	// "remoteUser": "root"
}
```

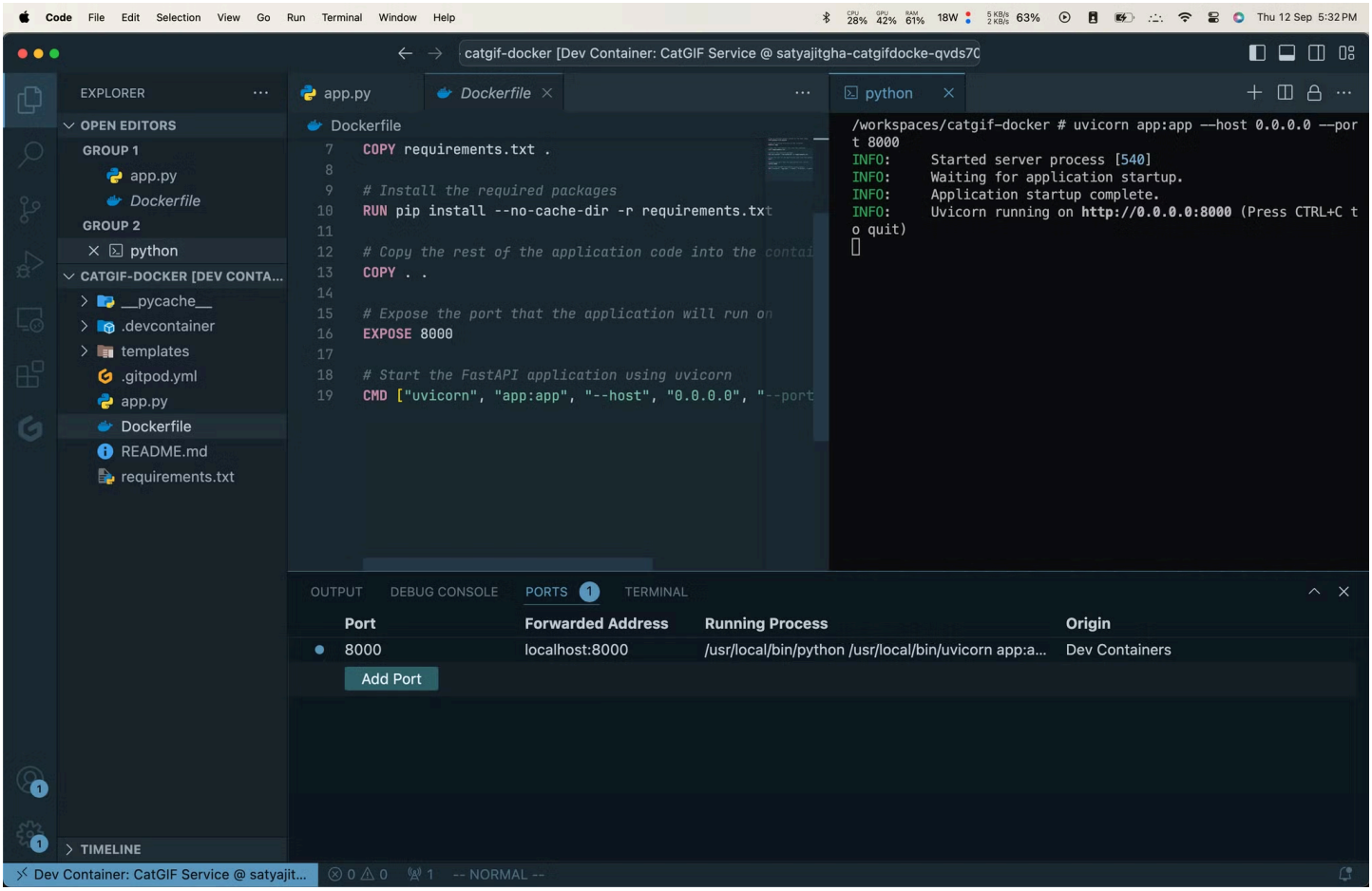
```
{
	"name": "CatGIF",
	// Or use a Dockerfile or Docker Compose file. More info: <https://containers.dev/guide/dockerfile>
```

```
        // "image": "mcr.microsoft.com/devcontainers/python:0-3.11",
    "build": {
        "dockerfile": "../Dockerfile",
        "context": ".."
    },

        // Features to add to the dev container. More info: <https://containers.dev/features>.
        // "features": {},

        // Configure tool-specific properties.
        "customizations": {
                // Configure properties specific to VS Code.
                "vscode": {
                        "settings": {},
                        "extensions": [
                                "ms-python.python"
                        ]
                }
        },

        // Use 'forwardPorts' to make a list of ports inside the container available locally.
        // "forwardPorts": [9000],

        // Use 'portsAttributes' to set default properties for specific forwarded ports.
        // More info: <https://containers.dev/implementors/json_reference/#port-attributes>
        "portsAttributes": {
                "8000": {
                        "label": "Flask Backend",
                        "onAutoForward": "notify"
                }
        }

        // Use 'postCreateCommand' to run commands after the container is created.
        // "postCreateCommand": "pip3 install -r requirements.txt"

        // Uncomment to connect as root instead. More info: <https://aka.ms/dev-containers-non-r
ot>.
        // "remoteUser": "root"
}
```


Code File Edit Selection View Go Run Terminal Window Help
Thu 12 Sep 5:29 PM
>rebui
Dev Containers: Rebuild and Reopen in Container
recently used
.NET: Rebuild
other commands
Azure Container Registry: Build Image in Azure...
Bazel: Refresh Bazel Build Targets
Dev Containers: Rebuild Without Cache and Reopen in Container
rust-analyzer: Rebuild proc macros and build scripts
EXPLORER
OPEN EDITORS
devcontainer.json .d...
CATGIF-DOCKER [SSH: SATYA...
.devcontainer
devcontainer.json
templates
.gitpod.yml U
app.py
Dockerfile
README.md
requirements.txt
"forwardPorts": [
8000
"customizations": {
"vscode": {
"settings": {},
"extensions": [
"ms-python.python"
OUTPUT DEBUG CONSOLE PORTS 3 EXPOSED PORTS SQL CONSOLE TERMINAL
Gitpod Task 1: bash
gitpod /workspace/catgif-docker (master) $
TIMELINE
MYSQL
VS CODE PETS
SSH: satyajitgha-catgifdocke-qvds70yd9fc...
master*
hku-mars/ImMesh#17 needs reviewers
Connect
Live Share
Git Graph
Share
DVC (Auto)


Code File Edit Selection View Go Run Terminal Window Help
Thu 12 Sep 5:30 PM
cker [Dev Container: CatGIF Service @ satyajitgha-catgifdocke-qvds70yd9fc.vss
EXPLORER
OPEN EDITORS
CATGIF-DOCKER [DEV CONTA...
OUTPUT PORTS TERMINAL
Dev Containers
[16652 ms]
[16652 ms]
[16652 ms] Start: Run in container: sed -i -E 's/((^|\s)PATH=)([^\$]*)$/\1${PATH:-\3}/g' /etc/profile || true
[16653 ms]
[16653 ms]
[16696 ms] Start: Run: docker inspect --type container 5695b7457ac1bbf1dfcc66490e01d1e2a21aa29cc76e133a33ba10e547982d83
[17378 ms] Start: Run: docker exec -i -u root 5695b7457ac1bbf1dfcc66490e01d1e2a21aa29cc76e133a33ba10e547982d83 /bin/sh -c echo "New container started. Keep-alive process started." ; export VSCODE_REMOTE_CONTAINERS_SESSION=bee87eab-1bfe-4b1c-a8f4-8a22b46b68651726142415022 ; /bin/sh
[17378 ms] Start: Inspecting container
[17379 ms] Start: Run: docker inspect --type container 5695b7457ac1
[18063 ms] Start: Run in container: /bin/sh
[18123 ms] New container started. Keep-alive process started.
Connecting to Dev Container (show log)
TIMELINE
Opening Remote...


Code File Edit Selection View Go Run Terminal Window Help
Thu 12 Sep 5:31 PM
catgif-docker [Dev Container: CatGIF Service @ satyajitgha-catgifdocke-qvds70
EXPLORER
OPEN EDITORS
app.py
CATGIF-DOCKER [DEV CONTA...
.devcontainer
templates
.gitpod.yml
app.py
Dockerfile
README.md
requirements.txt
TIMELINE
app.py
import os
from fastapi import FastAPI, Request
from fastapi.templating import Jinja2Templates
from fastapi.responses import HTMLResponse
import httpx
import random
app = FastAPI()
templates = Jinja2Templates(directory="templates")
TENOR_API_KEY = os.getenv("TENOR_API_KEY")
TENOR_API_URL = "https://tenor.googleapis.com/v2/search"
async def get_cat_gif_url():
async with httpx.AsyncClient() as client:
params = {
"q": "cat",
"key": TENOR_API_KEY,
"client_key": "my_test_app",
"limit": 50, # Fetch 50 results
"media_filter": "gif",
OUTPUT
DEBUG CONSOLE
PORTS 1
TERMINAL
/workspaces/catgif-docker #
Dev Contain...
sh
Do you want to install the recommended 'Python' extension from Microsoft for the Python language?
Install
Show Recommendations
Dev Container: CatGIF Service @ satyajit...
Checking for pending cloud changes...
-- NORMAL --
Ln 1, Col 1
Spaces: 4
UTF-8
LF



https://github.com/satyajitghana/catgif-docker/tree/devcontainer
(https://github.com/satyajitghana/catgif-docker/tree/devcontainer)

Installing packages in Dockerfile

```
RUN apt-get update && export DEBIAN_FRONTEND=noninteractive \
    && apt-get -y install git
```

Introduction to Docker Compose

Docker Compose is a tool that allows you to define and manage multi-container Docker applications. It uses YAML files to configure the application's services and with a single command, you can create and start all the services specified in your configuration.

Understanding Docker Compose

Before we delve deeper into Docker Compose, it's crucial to understand the basic components of Docker Compose, which include:

1. **Dockerfile**: This is a text file that contains all commands needed to build a Docker image.
2. **Docker Images**: These are read-only templates used to create containers. Images are created from Dockerfiles and are stored in Docker registries.
3. **Docker Containers**: These are runnable instances of Docker images.
4. **Docker Compose File (docker-compose.yml)**: This YAML file describes your app's services and defines which software, services, and tools are needed to run them.

Key Features of Docker Compose

- Multiple isolated environments on a single host
- Preserve volume data when containers are created
- Only recreate containers that have changed
- Variables and moving a composition between environments

Docker Compose Workflow

The Docker Compose tool follows a simple three-step process:

1. **Define your app's environment with a Dockerfile**: This file helps Docker to build the image of your application.
2. **Define the services that make up your app in docker-compose.yml**: This file specifies all the services that your application needs, such as databases, queues, caches, etc.
3. **Run docker-compose up and Compose starts and runs your entire app**: This command starts the whole system. If the images are not yet built or up-to-date, it builds them using the Dockerfile. It then (re)creates the containers for the services defined in the docker-compose.yml file.

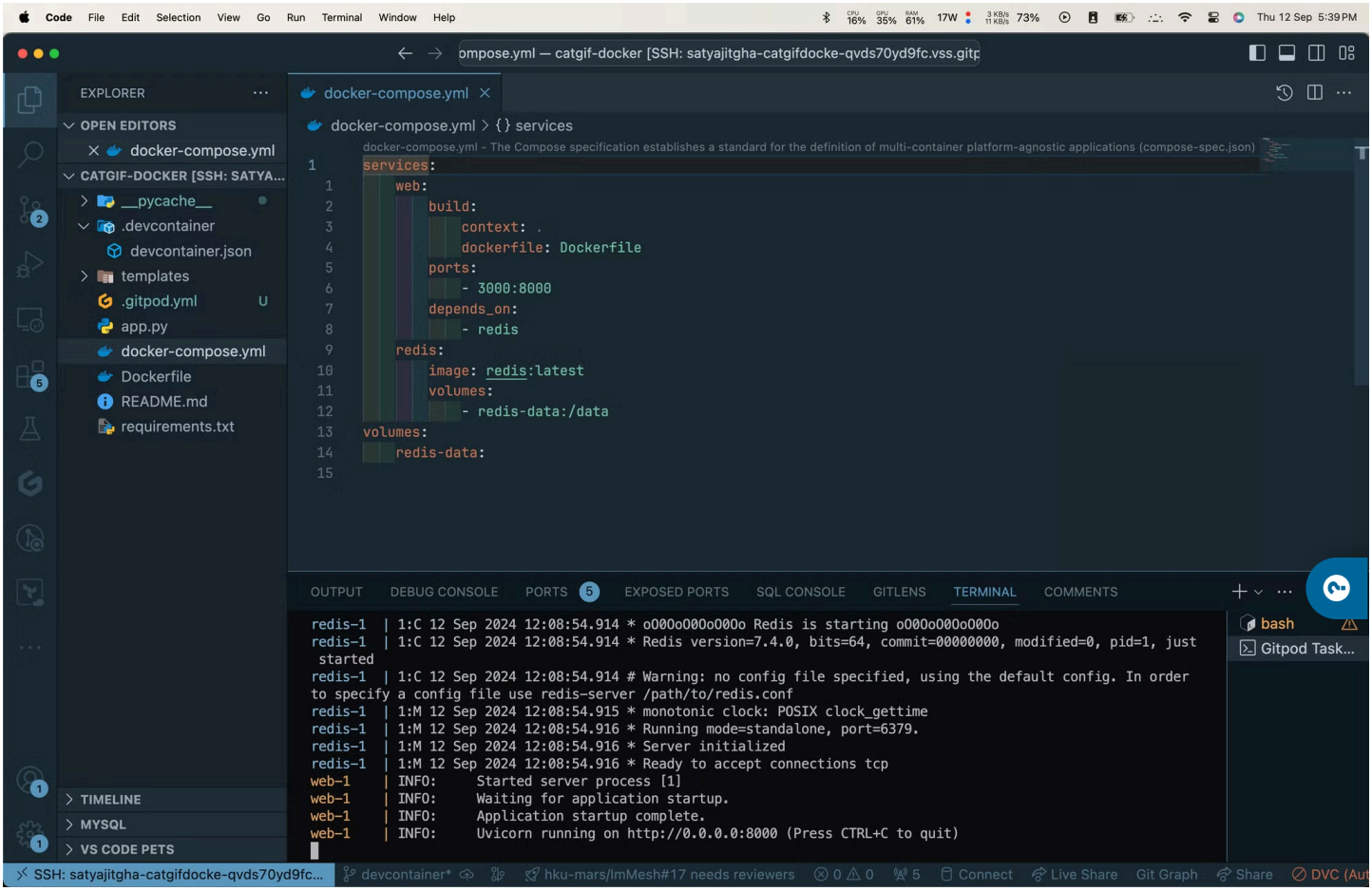
Migrate to Docker Compose

```
git clone https://github.com/satyajitghana/catgif-docker (https://github.com/satyajitghana/catgif-docker)

cd catgif-docker

git checkout origin/compose
```

`docker-compose.yml`

```
services:
    web:
        build:
            context: .
            dockerfile: Dockerfile
        ports:
            - 3000:8000
        depends_on:
            - redis
```

```
    redis:
        image: redis:latest
        volumes:
            - redis-data:/data
volumes:
    redis-data:
```




```
docker compose build

docker compose up
```

Docker Volumes

```
docker volume create hello
```

```
docker run -it --rm -v hello:/data ubuntu bash
```

Let’s see what is inside the redis volume

```
gitpod /workspace/catgif-docker (devcontainer) $ docker volume ls
DRIVER    VOLUME NAME
local     catgif-docker_redis-data
local     vscode
```

```
docker run -it --rm -v catgif-docker_redis-data:/data ubuntu bash
```

DevContainer with Docker Compose

https://containers.dev/guide/dockerfile#docker-compose-dockerfile
(https://containers.dev/guide/dockerfile#docker-compose-dockerfile)

Training, Eval & Deploy Example

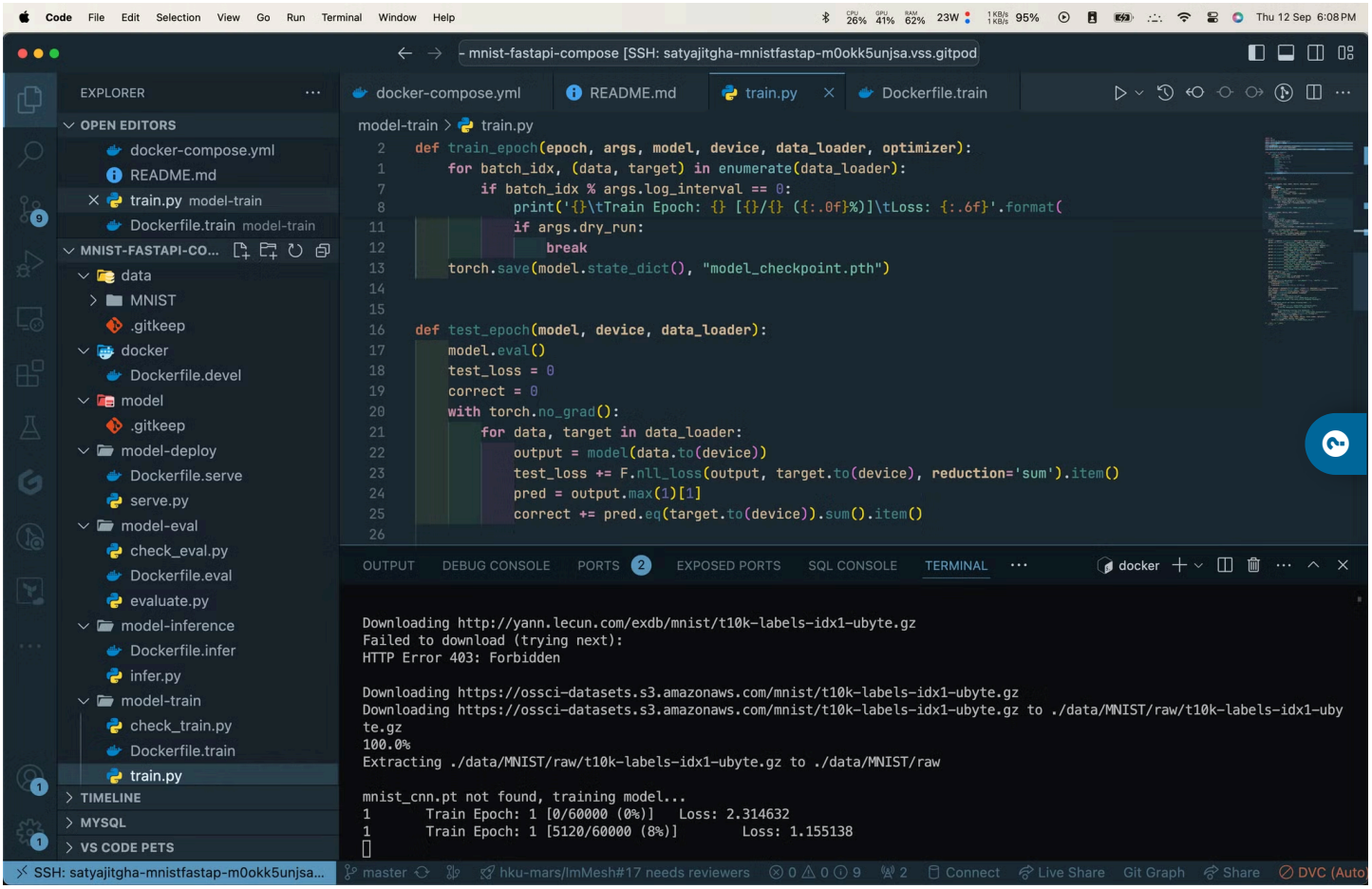
https://github.com/satyajitghana/mnist-fastapi-compose
(https://github.com/satyajitghana/mnist-fastapi-compose)

```
git clone https://github.com/satyajitghana/mnist-fastapi-compose (https://github.com/satyajitghana/mnist-fastapi-compose)
```

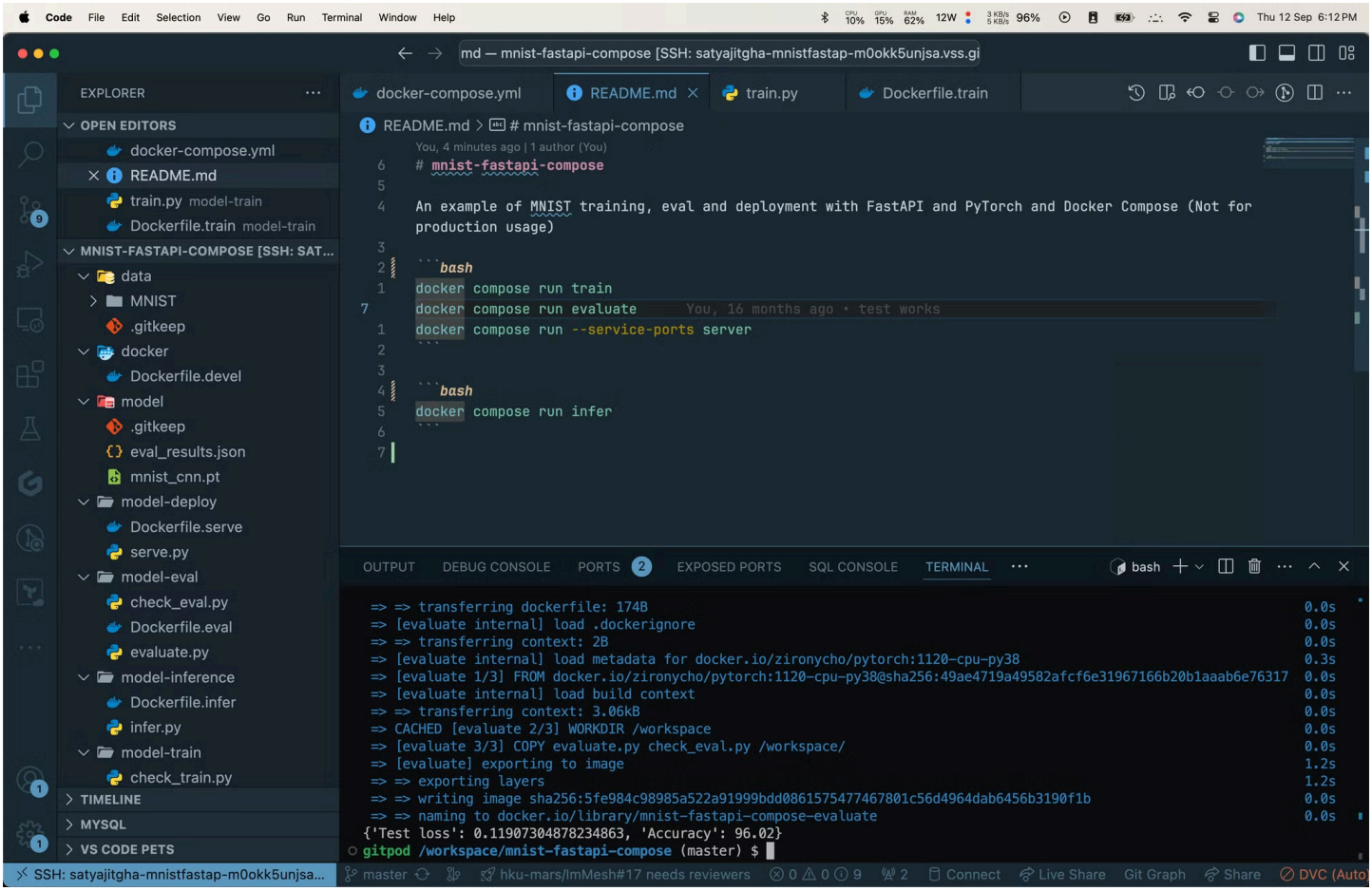
```
version: '3.8'

services:
  train:
    build:
      context: model-train
      dockerfile: Dockerfile.train
    volumes:
      - ./model:/workspace/model
      - ./data:/workspace/data
    restart: on-failure

  evaluate:
    build:
      context: model-eval
      dockerfile: Dockerfile.eval
    volumes:
      - ./model:/workspace/model
      - ./data:/workspace/data
    restart: on-failure

  server:
    build:
      context: model-deploy
      dockerfile: Dockerfile.serve
    volumes:
      - ./model:/app/model
    ports:
      - "8000:80"
    restart: unless-stopped

  infer:
```

```
    build:
      context: model-inference
      dockerfile: Dockerfile.infer
    volumes:
      - ./data:/workspace/data
      - ./requests:/workspace/requests
    network_mode: "host"
    restart: no
```

```
docker compose run train
```

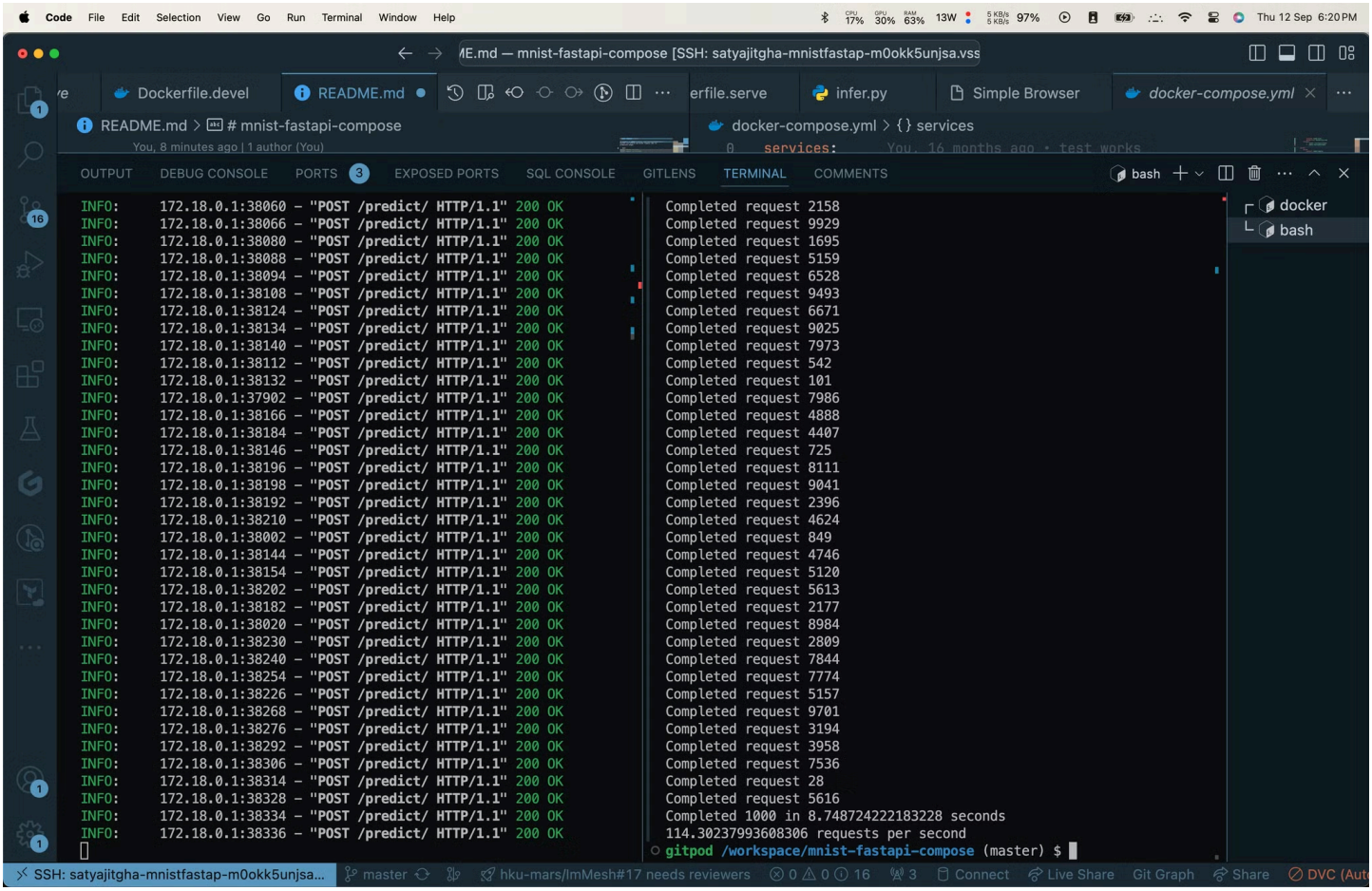



```
docker compose run evaluate
```




```
docker compose run --service-ports server
```

```
docker compose run infer
```



Multiple Compose Files

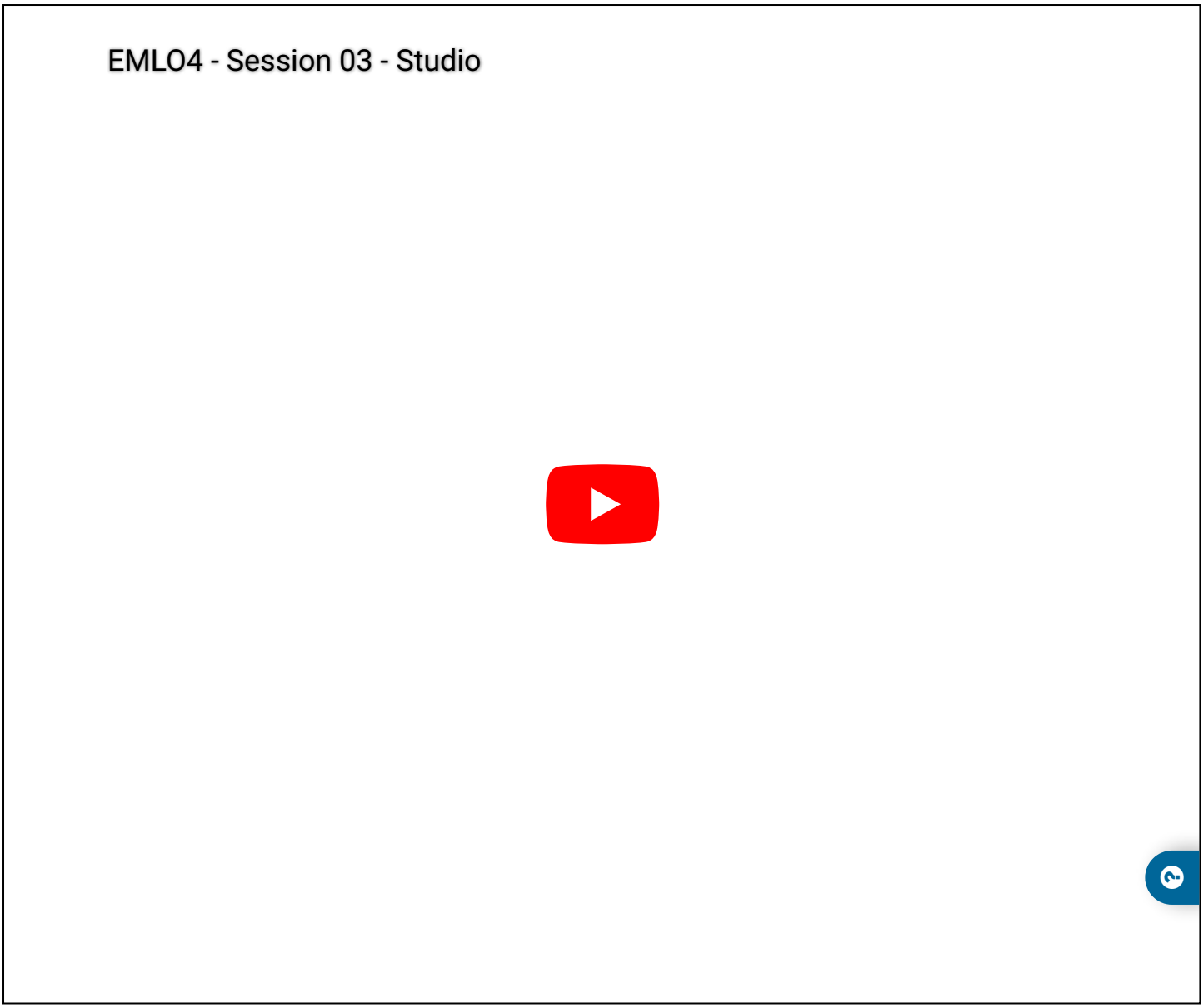
https://docs.docker.com/compose/extends/ (https://docs.docker.com/compose/extends/)

Load Balancing and Scaling with Docker Compose

https://medium.com/@vinodkrane/microservices-scaling-and-load-balancing-using-docker-compose-78bf8dc04da9 (https://medium.com/@vinodkrane/microservices-scaling-and-load-balancing-using-docker-compose-78bf8dc04da9)

Recordings

Studio



Assignment

Docker Compose for MNIST Training, Evaluation, and Inference

In this assignment, you will create a Docker Compose configuration to perform training, evaluation, and inference on the MNIST dataset.

Requirements:

1. You'll need to use this model and training technique (MNIST Hogwild): **https://github.com/pytorch/examples/tree/main/mnist_hogwild**

(https://github.com/pytorch/examples/tree/main/mnist_hogwild)

2. Set Num Processes to 2 for MNIST HogWild
3. Create three services in the Docker Compose file: `train`, `evaluate`, and `infer`.
4. Use a shared volume called `mnist` for sharing data between the services.
5. The `train` service should:
   - Look for a checkpoint file in the volume. If found, resume training from that checkpoint. Train for **ONLY 1 epoch** and save the final checkpoint. Once done, exit.
6. The `evaluate` service should:
   - Look for the final checkpoint file in the volume. Evaluate the model using the checkpoint and save the evaluation metrics in a json file. Once done, exit.
   - Share the model code by importing the model instead of copy-pasting it in **eval.py** (http://eval.py)
7. The `infer` service should:
   - Run inference on any 5 random MNIST images and save the results (images with file name as predicted number) in the `results` folder in the volume. Then exit.
8. After running all the services, ensure that the model, and results are available in the `mnist` volume.

Detailed Instructions:

1. Build all the Docker images using `docker compose build`.
2. Run the Docker Compose services using `docker compose run train`, `docker compose run evaluate`, and `docker compose run infer`. Verify that all services have completed successfully.
3. Check if the checkpoint file (`mnist_cnn.pt`) is saved in the `mnist` volume. If found, display "Checkpoint file found." If not found, display "Checkpoint file not found!" and exit with an error.
4. Check if the evaluation results file (`eval_results.json`) is saved in the `mnist` volume.
   1. Example: `{"Test loss": 0.0890245330810547, "Accuracy": 97.12}`
5. Check the contents of the `results` folder in the `mnist` volume see if the inference results are saved.
6. You can always comment/uncomment parts of the **grading.sh** (http://grading.sh) to skip some tests, this would reduce time while debugging and remove the need to run all the tests everytime you run **grading.sh** (http://grading.sh). But make sure you don't commit changes in **grading.sh** (http://grading.sh) to repo

The provided grading script will run the Docker Compose configuration, check for the required files, display the results, and perform size and version checks.

You can run it yourself before pushing the code to your repo

```
bash tests/grading.sh
```

Also, You can use ChatGPT o1, Claude, Cursor for the Assignment 😄

Github Classroom:

https://classroom.github.com/a/H1dh0F7f

(https://classroom.github.com/a/H1dh0F7f)