

A screenshot of the Visual Studio Code interface. The title bar shows "prometheus.yml - tmp - Visual Studio Code". The left sidebar has sections for "OPEN EDITORS" (Get Started, Untitled-1, prometheus.yml), "TMP" (prometheus.yml), and "OUTLINE" (No symbols found in document 'prometheus.yml'). The main editor area displays a YAML configuration file for Prometheus:

```
global:
  scrape_interval:      15s # Set the scrape interval to every 15 seconds.
  evaluation_interval: 15s # Evaluate rules every 15 seconds. The default
  # scrape_timeout is set to the global default (10s).

  # Attach these labels to any time series or alerts when communicating with
  # external systems (federation, remote storage, Alertmanager).
  external_labels:
    monitor: 'codelab-monitor'

# Load rules once and periodically evaluate them according to the global
rule_files:
  # - "first.rules"
  # - "second.rules"

# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any timeseries sc
  - job_name: 'prometheus'

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

    static_configs:
      - targets: ['host.docker.internal:9090'] # Only works on Docker Des
        I
      - job_name: 'docker'
        I I I # metrics_path defaults to '/metrics'
        I I I # scheme defaults to 'http'.

        static_configs:
          - targets: ['192.168.65.1:9323']
```

The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** Shows two files: "Get Started" and "prometheus.yml".
- Terminal:** Displays a Windows PowerShell session with the following command history:
  - PS C:\> **docker service create** --replicas 1 --name my-prometheus --mount type=bind,source=c:/tmp/prometheus.yml --publish published=9090,target=9090,protocol=tcp prom/prometheus
  - invalid argument "type=bind,source=c:/tmp/prometheus.yml" for "--mount" flag: target is required
  - See 'docker service create --help'.
  - PS C:\> **docker service create** --replicas 1 --name my-prometheus --mount type=bind,source=c:/tmp/prometheus.yml,destination=/etc/prometheus/prometheus.yml --publish published=9090,target=9090,protocol=tcp prom/prometheus
  - Error response from daemon: rpc error: code = AlreadyExists desc = name conflicts with an existing object: service my-prometheus already exists
  - PS C:\> **docker service create** --replicas 10 --name ping\_service alpine ping docker.com
  - nnfjdc24w13hwth7amcr31dnb
  - overall progress: 10 out of 10 tasks
    - 1/10: running
    - 2/10: running
    - 3/10: running
    - 4/10: running
    - 5/10: running
    - 6/10: running
    - 7/10: running

A screenshot of the Visual Studio Code interface. The top menu bar includes File, Edit, Selection, View, Go, Run, Terminal, and Help. The title bar shows "prometheus.yml - tmp - Visual Studio Code".

The left sidebar has sections for PDF, Document, RAM, and TMP. The TMP section contains two files: "prometheus.yml" (selected) and "prometheus.yml".

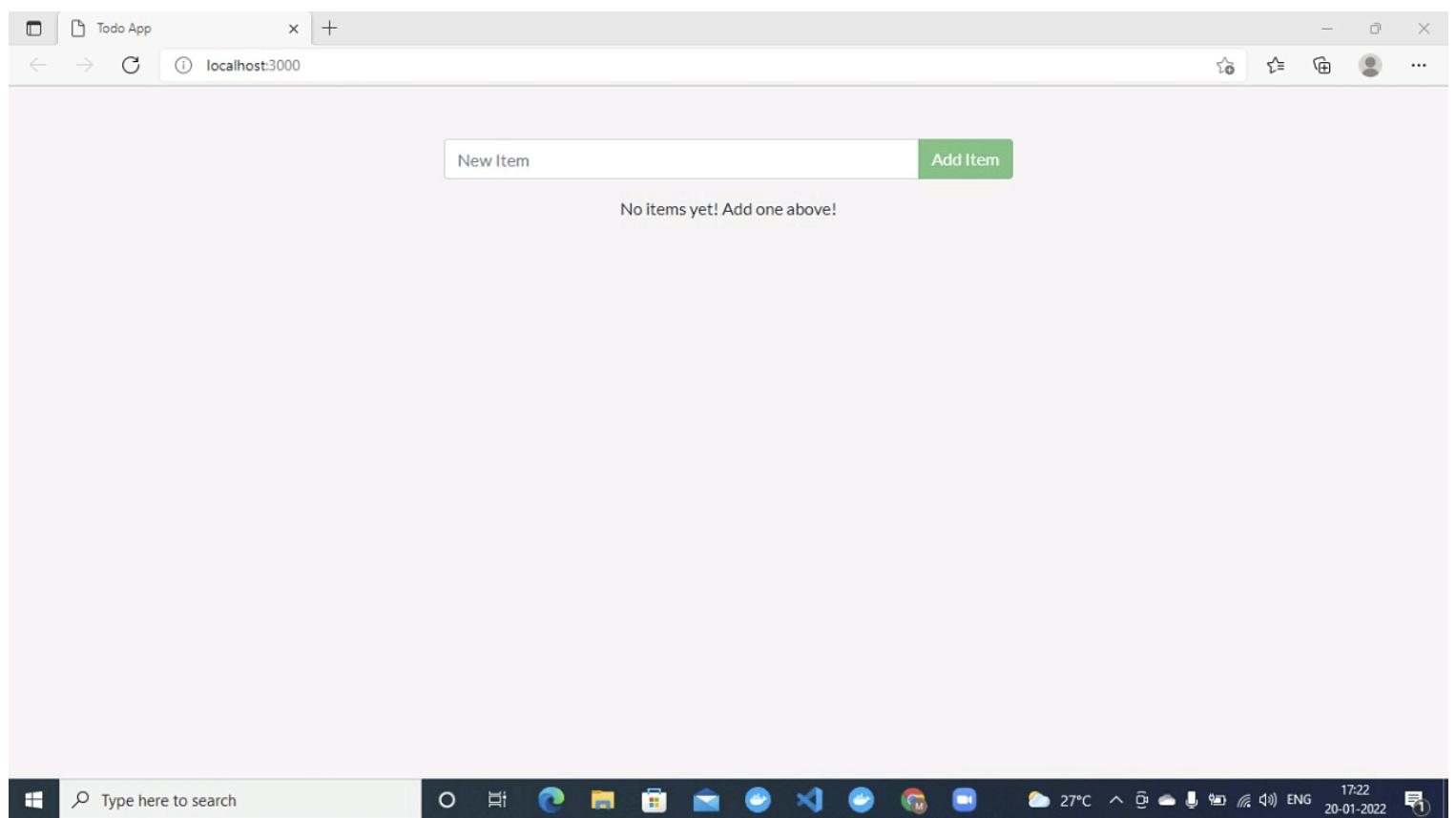
The main editor area displays the content of "prometheus.yml":

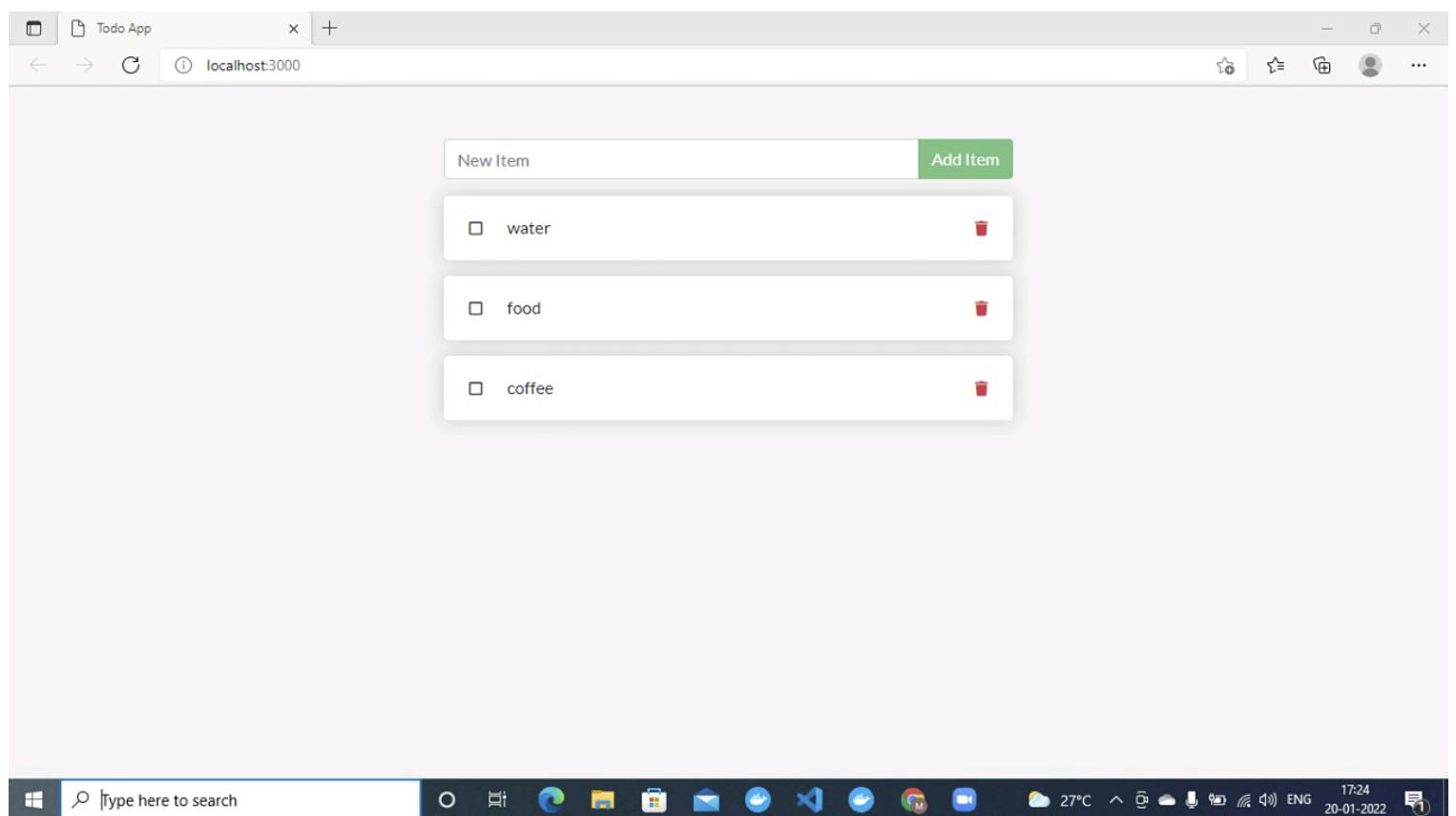
```
targets: [ host.docker.internal:9090 ] # Only works on Docker Desktop
job_name: 'docker'
# metrics_path defaults to '/metrics'
# scheme defaults to 'http'.
static_configs:
- targets: ['192.168.65.1:9323']
```

The bottom navigation bar includes PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL. The TERMINAL tab is active, showing the following output:

```
2/10: running
3/10: running
4/10: running
5/10: running
6/10: running
7/10: running
8/10: running
9/10: running
10/10: running
verify: Service converged
PS C:\> docker service remove ping_service
ping_service
PS C:\>
```

The bottom left corner shows a sidebar with sections for OUTLINE, PDF, and RAM.





A screenshot of a Windows desktop showing a browser window and a taskbar.

The browser window displays a Docker container monitoring interface for a session named "c7kno1vn\_c7kno3nnjsv000br5ong". The session details show:

- IP: 192.168.0.28
- Memory: 13.84% (553.4MiB / 3.906GiB)
- CPU: 0.33%

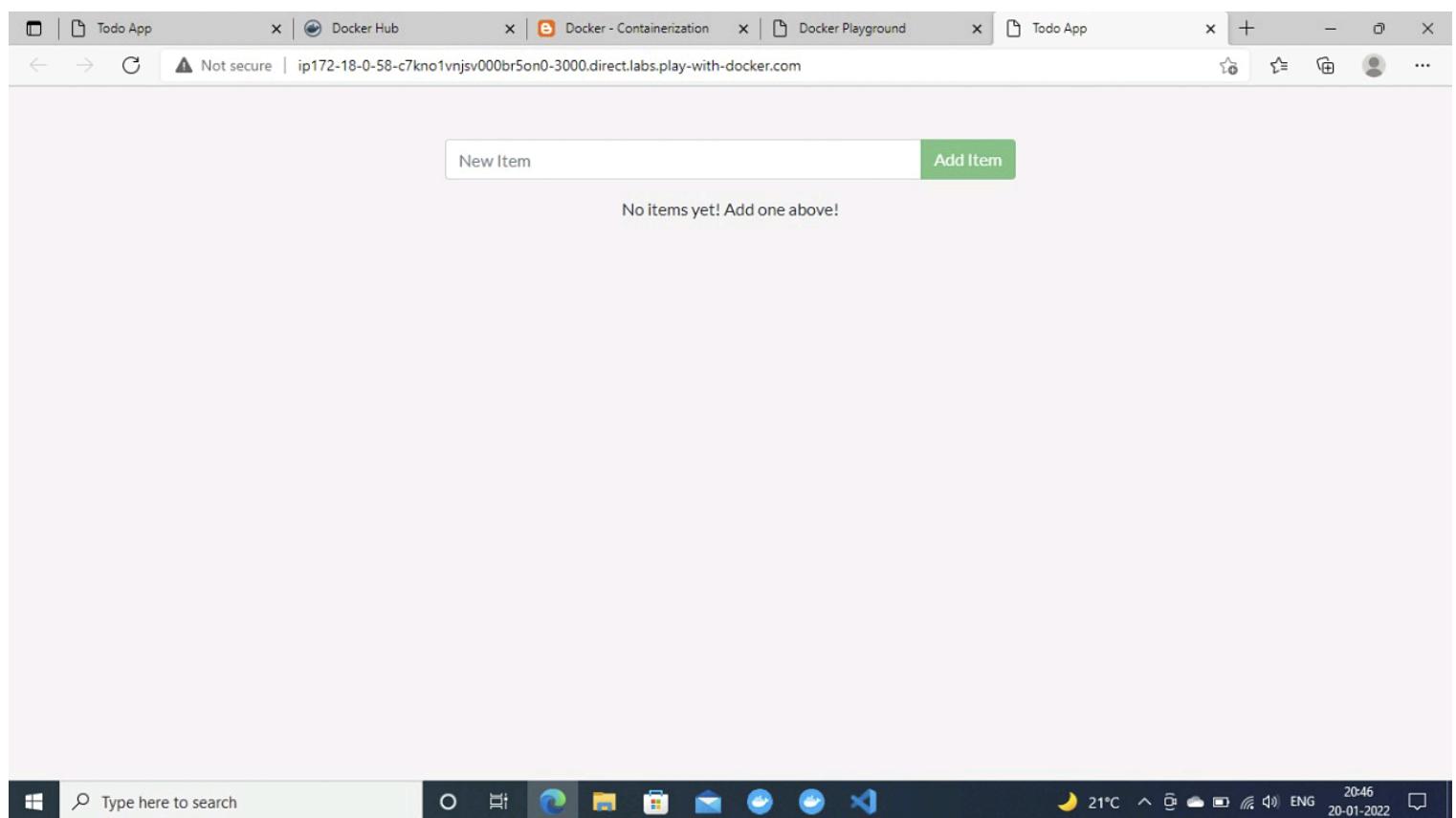
The "OPEN PORT" button is highlighted with the value "3000". Below the IP section, there is an SSH link: "ssh ip172-18-0-58-c7kno1vnjsv000br5on0@direct.labs.play-".

On the left sidebar, there is a "CLOSE SESSION" button and a "Instances" section with a "node1" entry. A "+ ADD NEW INSTANCE" button is also present.

The main content area shows a terminal session output:

```
# The PWD team.
#####
[node1] (local) root@192.168.0.28 ~
$ docker run -d -p 3000:3000 mounikai1213/getting-started
Unable to find image 'mounikai1213/getting-started:latest' locally
latest: Pulling from mounikai1213/getting-started
97518928ae5f: Pull complete
58ab2943ea3a: Pull complete
da5d3df7401d: Pull complete
7384cfecbf77: Pull complete
6c66ce49109e: Pull complete
80bdb47ff447: Pull complete
c81e5200edb1: Pull complete
40db6297131c: Pull complete
Digest: sha256:73d0c80ce4546596611e994ae60336d44adc0c837a19493fdbb11f9dca3bd869
Status: Downloaded newer image for mounikai1213/getting-started:latest
9b0ddcce3dbe6b6ba1a543f2172e8b61ffad8a1514d1adc79a96b5fa1e86240
[node1] (local) root@192.168.0.28 ~
$ 
```

The taskbar at the bottom includes icons for File Explorer, Edge, Mail, and Task View, along with system status indicators like battery level, network, and date/time (20:46, 20-01-2022).



Docker

Images on disk

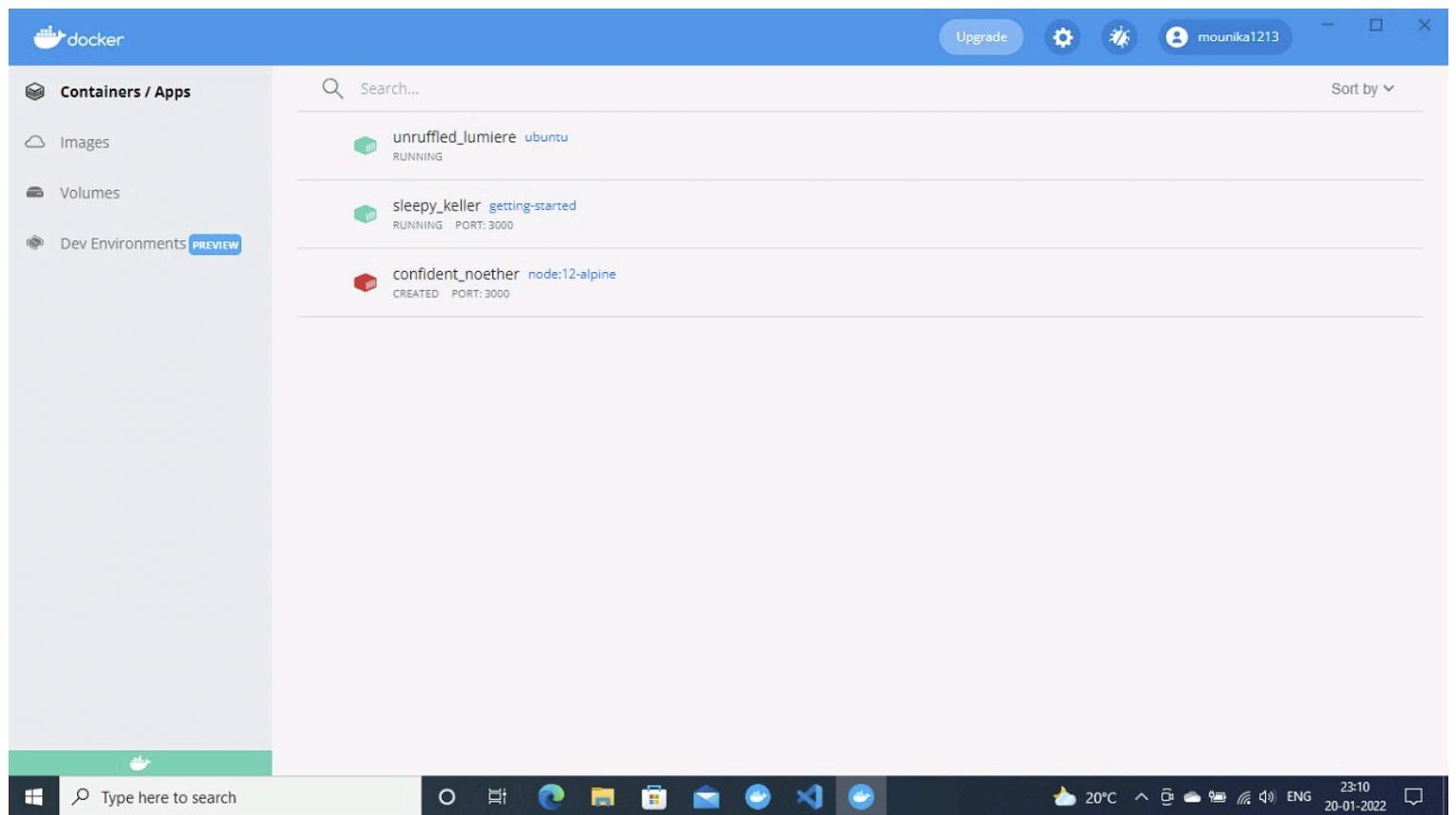
4 images Total size: 528.37 MB IN USE UNUSED Clean up...

LOCAL REMOTE REPOSITORIES

NAME	TAG	IMAGE ID	CREATED	SIZE
getting-started	latest	a121143bb638	about 3 hours ago	455.6 MB
mounika1213/getting-started	latest	a121143bb638	about 3 hours ago	455.6 MB
node	12-alpine	8a6e486e9817	9 days ago	91.15 MB
ubuntu	latest	d13c942271d6	14 days ago	72.78 MB

Type here to search

20°C ENG 23:10 20-01-2022



Docker

Containers / Apps

Images

Volumes

Dev Environments PREVIEW

## Volumes

Search

NAME ↑	CREATED	SIZE
todo-db	IN USE	about 11 hours ago 16 kB
todo-mysql-data	IN USE	19 minutes ago 210.1 MB

Type here to search

09:59 21-01-2022

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\himav\Desktop\kubernetes> kubectl logs demo
PING 8.8.8.8 (8.8.8.8): 56 data bytes
64 bytes from 8.8.8.8: seq=0 ttl=37 time=3066.839 ms
64 bytes from 8.8.8.8: seq=1 ttl=37 time=1871.848 ms
64 bytes from 8.8.8.8: seq=2 ttl=37 time=871.676 ms
64 bytes from 8.8.8.8: seq=3 ttl=37 time=89.771 ms
64 bytes from 8.8.8.8: seq=4 ttl=37 time=79.639 ms
64 bytes from 8.8.8.8: seq=6 ttl=37 time=61.242 ms
64 bytes from 8.8.8.8: seq=7 ttl=37 time=89.583 ms
64 bytes from 8.8.8.8: seq=8 ttl=37 time=89.083 ms
64 bytes from 8.8.8.8: seq=9 ttl=37 time=58.410 ms
64 bytes from 8.8.8.8: seq=10 ttl=37 time=119.920 ms
64 bytes from 8.8.8.8: seq=11 ttl=37 time=88.040 ms
64 bytes from 8.8.8.8: seq=13 ttl=37 time=238.489 ms
64 bytes from 8.8.8.8: seq=14 ttl=37 time=104.452 ms
64 bytes from 8.8.8.8: seq=15 ttl=37 time=167.542 ms
64 bytes from 8.8.8.8: seq=16 ttl=37 time=112.235 ms
64 bytes from 8.8.8.8: seq=17 ttl=37 time=95.679 ms
64 bytes from 8.8.8.8: seq=18 ttl=37 time=139.429 ms
64 bytes from 8.8.8.8: seq=20 ttl=37 time=81.616 ms
64 bytes from 8.8.8.8: seq=21 ttl=37 time=78.046 ms
64 bytes from 8.8.8.8: seq=23 ttl=37 time=86.478 ms
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

```
Run 'kubectl --help' for usage.
PS C:\Users\himav\Desktop\kubernetes> kubectl delete -f pod.yaml
pod "demo" deleted
PS C:\Users\himav\Desktop\kubernetes> docker swarm init
Error response from daemon: This node is already part of a swarm. Use "docker swarm leave"
PS C:\Users\himav\Desktop\kubernetes> docker swarm join --token
flag needs an argument: --token
See 'docker swarm join --help'.
PS C:\Users\himav\Desktop\kubernetes> docker service ps demo
no such service: demo
PS C:\Users\himav\Desktop\kubernetes> docker service logs demo
no such task or service: demo
PS C:\Users\himav\Desktop\kubernetes> docker service rm demo
Error: No such service: demo
PS C:\Users\himav\Desktop\kubernetes> docker swarm join --token
flag needs an argument: --token
See 'docker swarm join --help'.
PS C:\Users\himav\Desktop\kubernetes> docker service ps demo
no such service: demo
PS C:\Users\himav\Desktop\kubernetes> docker service logs demo
no such task or service: demo
PS C:\Users\himav\Desktop\kubernetes> docker service rm demo
```

File Selection View Go Run Terminal Help

bb.yaml - kubernetes - Visual Studio Code

EXPLORER ... Get Started bb.yaml

OPEN EDITORS

Get Started bb.yaml

KUBERNETE

bb.yaml pod.yaml

bb.yaml

bb.yaml

! bb.yaml

U U U

15 26 30

spec:

type: NodePort

selector:

bb: web

ports:

- port: 3000

targetPort: 3000

nodePort: 30001

33

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\Users\himav\Desktop\kubernetes> kubectl apply -f bb.yaml

deployment.apps/bb-demo created

service/bb-entrypoint created

PS C:\Users\himav\Desktop\kubernetes> kubectl get deployment

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
bb-demo	0/1	1	0	31s

PS C:\Users\himav\Desktop\kubernetes> kubectl get service

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
bb-entrypoint	NodePort	10.106.146.87	<none>	3000:30001/TCP	47s
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	2d

PS C:\Users\himav\Desktop\kubernetes> kubectl delete -f bb.yaml

Error: unknown command "delete" for "kubectl"

Did you mean this?

delete

Run 'kubectl --help' for usage.

PS C:\Users\himav\Desktop\kubernetes> kubectl delete -f bb.yaml

error: the path "bb.yaml" does not exist

PS C:\Users\himav\Desktop\kubernetes> kubectl delete -f bb.yaml

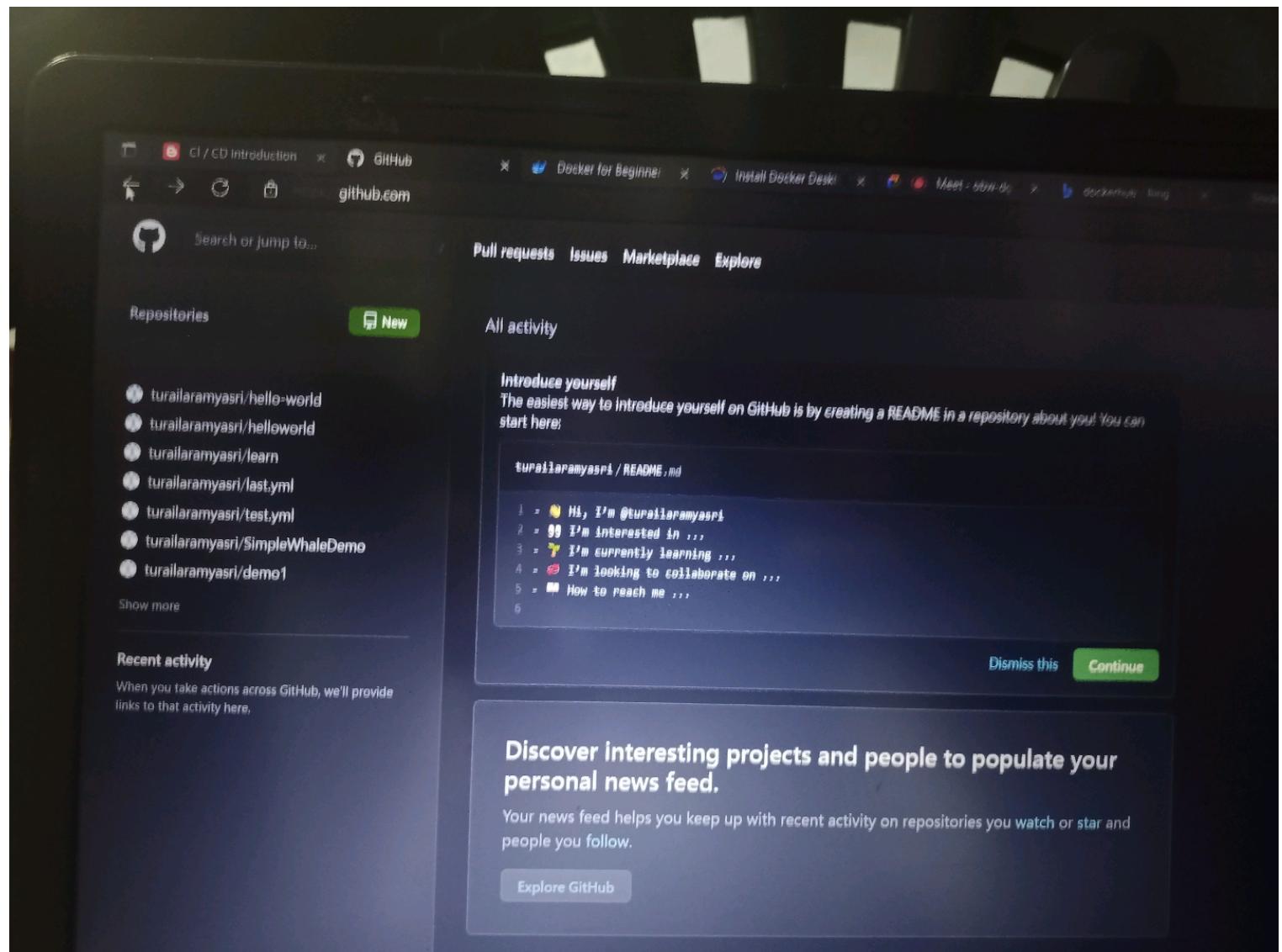
deployment.apps "bb-demo" deleted

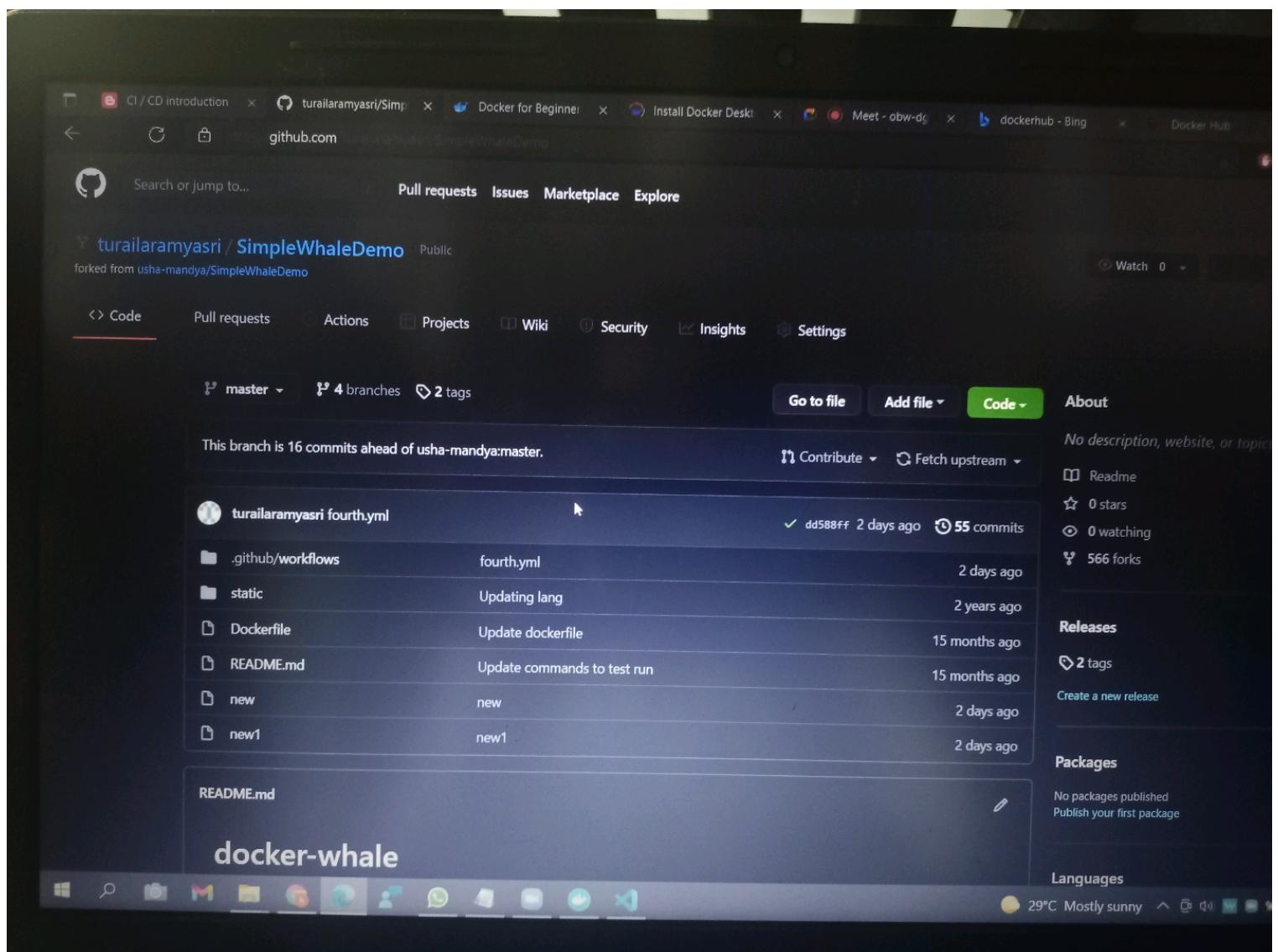
service "bb-entrypoint" deleted

PS C:\Users\himav\Desktop\kubernetes>

> OUTLINE

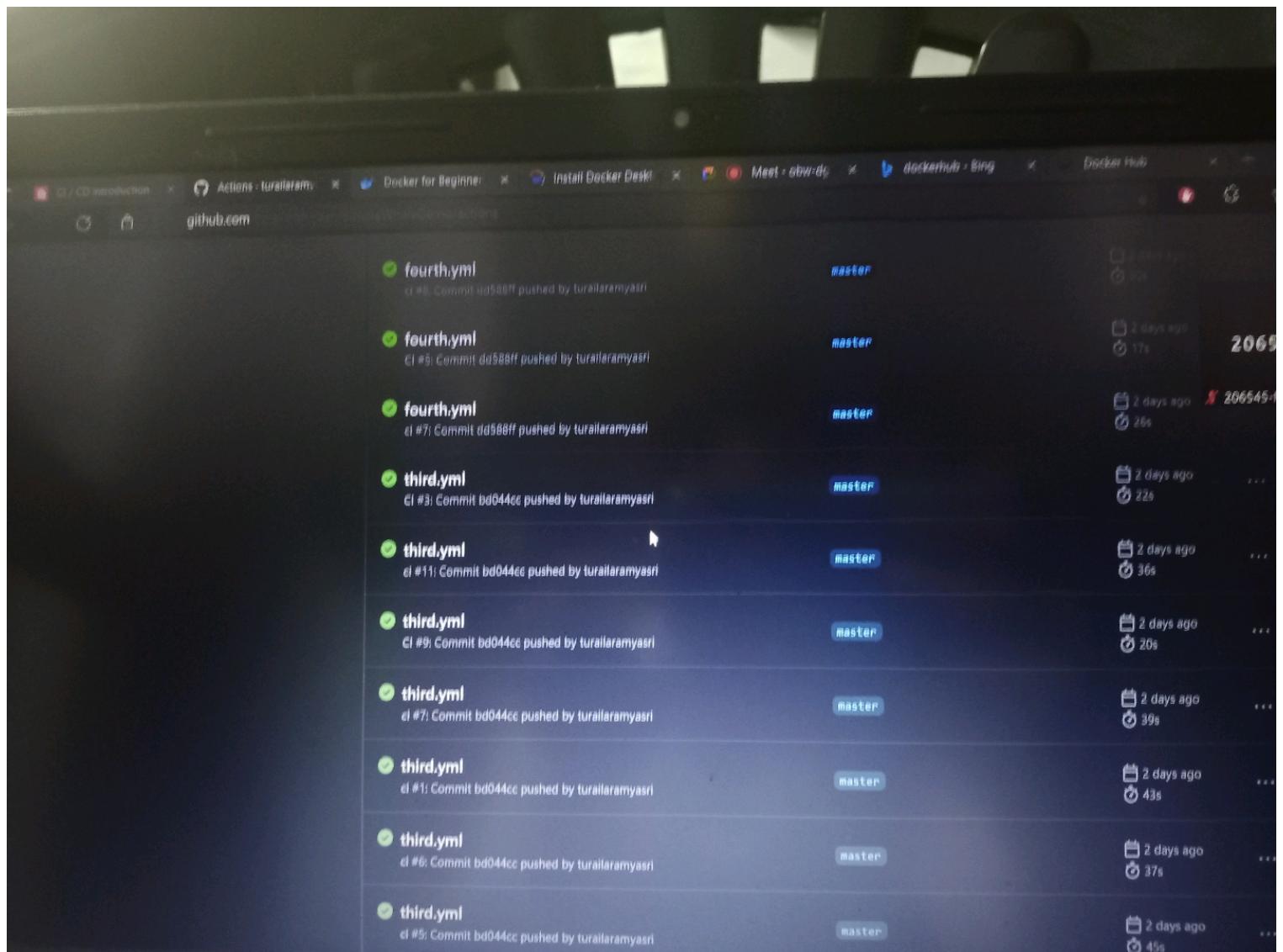
> TIMELINE

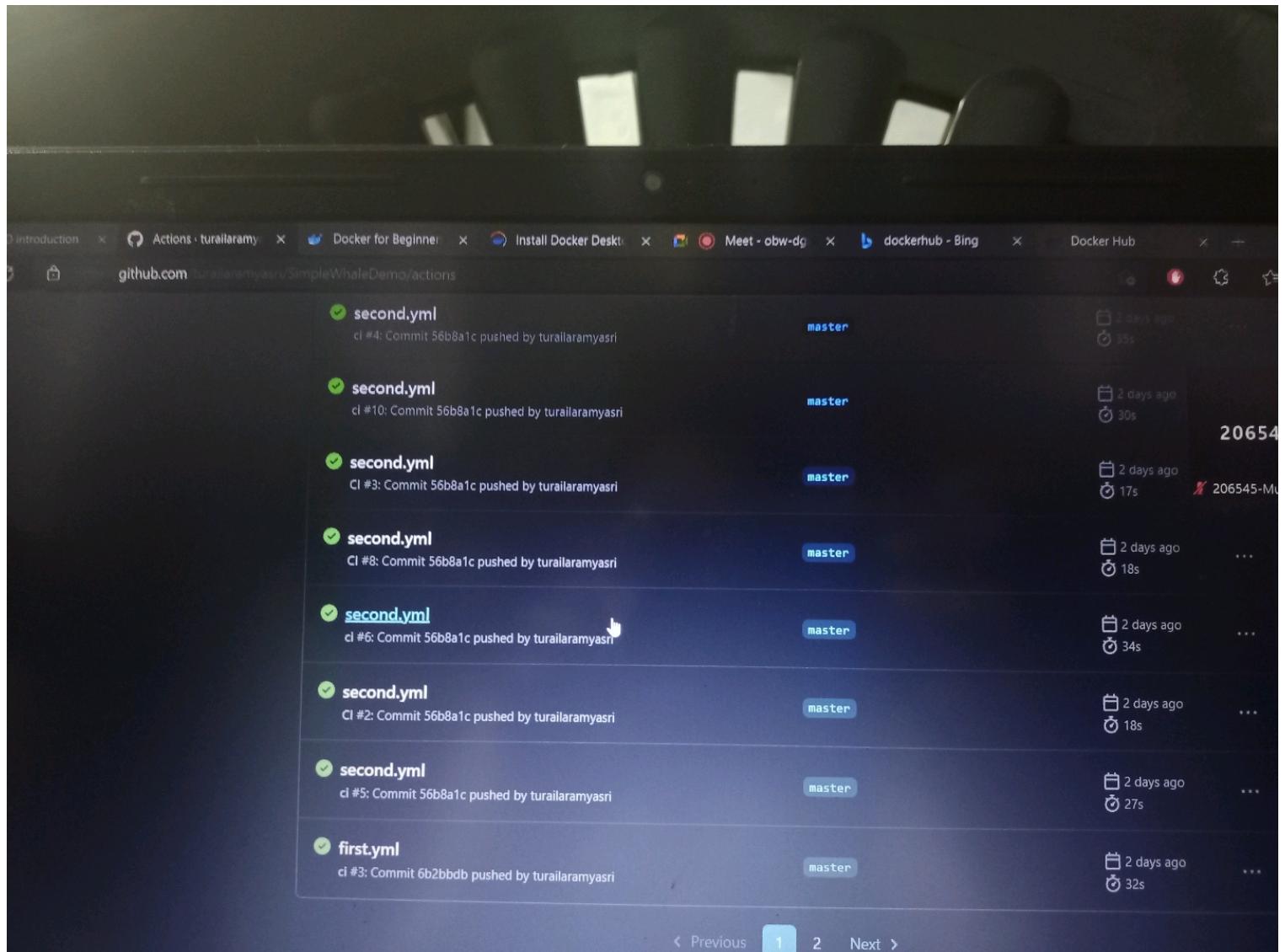




The screenshot shows a GitHub Actions workflow runs page. The top navigation bar includes tabs for Pull requests, Actions (which is active), Projects, Wiki, Security, Insights, and Settings. The main heading is "All workflows" with a subtitle "Showing runs from all workflows". A large blue button labeled "All workflows" is prominent. On the left, there's a sidebar with various CI logs and a "New workflow" button. The main content area displays "31 workflow runs" for the "fourth.yml" workflow. Each run is listed with a green checkmark icon, the workflow name, a commit message, the branch (master), and the run time (e.g., 2 days ago, 1m 0s). The last run shown is "ci #1: Commit dd588ff pushed by turailaramyasi". The bottom of the screen shows a dark dock with various application icons and a system status bar indicating "29°C Mostly sunny".

Run ID	Workflow	Commit Message	Branch	Time
ci #1	fourth.yml	ci #1: Commit dd588ff pushed by turailaramyasi	master	2 days ago 1m 0s
ci #2	fourth.yml	ci #2: Commit dd588ff pushed by turailaramyasi	master	2 days ago 34s
ci #12	fourth.yml	ci #12: Commit dd588ff pushed by turailaramyasi	master	2 days ago 34s
ci #6	fourth.yml	ci #6: Commit dd588ff pushed by turailaramyasi	master	2 days ago 28s
ci #10	fourth.yml	ci #10: Commit dd588ff pushed by turailaramyasi	master	2 days ago 19s
ci #4	fourth.yml	ci #4: Commit dd588ff pushed by turailaramyasi	master	2 days ago 16s
ci #1	fourth.yml		master	2 days ago





first.yml · turaillaram · GitHub Docker for Beginner · Install Docker Desktop · Meet - obw-dg · dockerhub - Bing · Docker Hub

github.com

```
1 name: ci
2 on:
3   push:
4     branches:
5       - 'master'
6     tags:
7       - 'v*'
8   jobs:
9     build:
10       runs-on: ubuntu-latest
11       steps:
12         -
13           name: Checkout
14           uses: actions/checkout@v2
15         -
16           name: Login to Docker Hub
17           uses: docker/login-action@v1
18           with:
19             username: ${{ secrets.DOCKER_HUB_USERNAME }}
20             password: ${{ secrets.DOCKER_HUB_ACCESS_TOKEN }}
21         -
22           name: Set up Docker Buildx
23           uses: docker/setup-buildx-action@v1
24         -
25           name: Build and push
26           uses: docker/build-push-action@v2
27           with:
28             context: .
29             file: ./Dockerfile
30             push: true
31             tags: ${{ secrets.DOCKER_HUB_USERNAME }}/simplewhale:latest
32             cache-from: type=registry,ref=${{ secrets.DOCKER_HUB_USERNAME }}/docker:buildcache
33             cache-to: type=registry,ref=${{ secrets.DOCKER_HUB_USERNAME }}/docker:buildcache,mode=max
```

```
1 This is a basic workflow to help you get started with Actions
2
3 name: CI
4
5 # Controls when the workflow will run
6 on:
7   # Triggers the workflow on push or pull request events but only for the master branch
8   push:
9     branches: [ master ]
10    pull_request:
11      branches: [ master ]
12
13    # Allows you to run this workflow manually from the Actions tab
14    workflow_dispatch:
15
16    # A workflow run is made up of one or more jobs that can run sequentially or in parallel
17    jobs:
18      # This workflow contains a single job called "build"
19      build:
20        # The type of runner that the job will run on
21        runs-on: ubuntu-latest
22
23        # Steps represent a sequence of tasks that will be executed as part of the job
24        steps:
25          # Checks-out your repository under $GITHUB_WORKSPACE, so your job can access it
26          - uses: actions/checkout@v2
27
28          # Runs a single command using the runners shell
29          - name: Run a one-line script
30            run: echo Hello, world!
31
32          # Runs a set of commands using the runners shell
33          - name: Run a multi-line script
34            run: |
35              echo Add other actions to build,
36              echo test, and deploy your project.
```

```
ub.com

      name: CI

      # Triggers the workflow whenever there's a push or pull request event.
      # The workflow does not run if there are merge conflicts.
      on:
        push:
          branches: [ master ]
        pull_request:
          branches: [ master ]

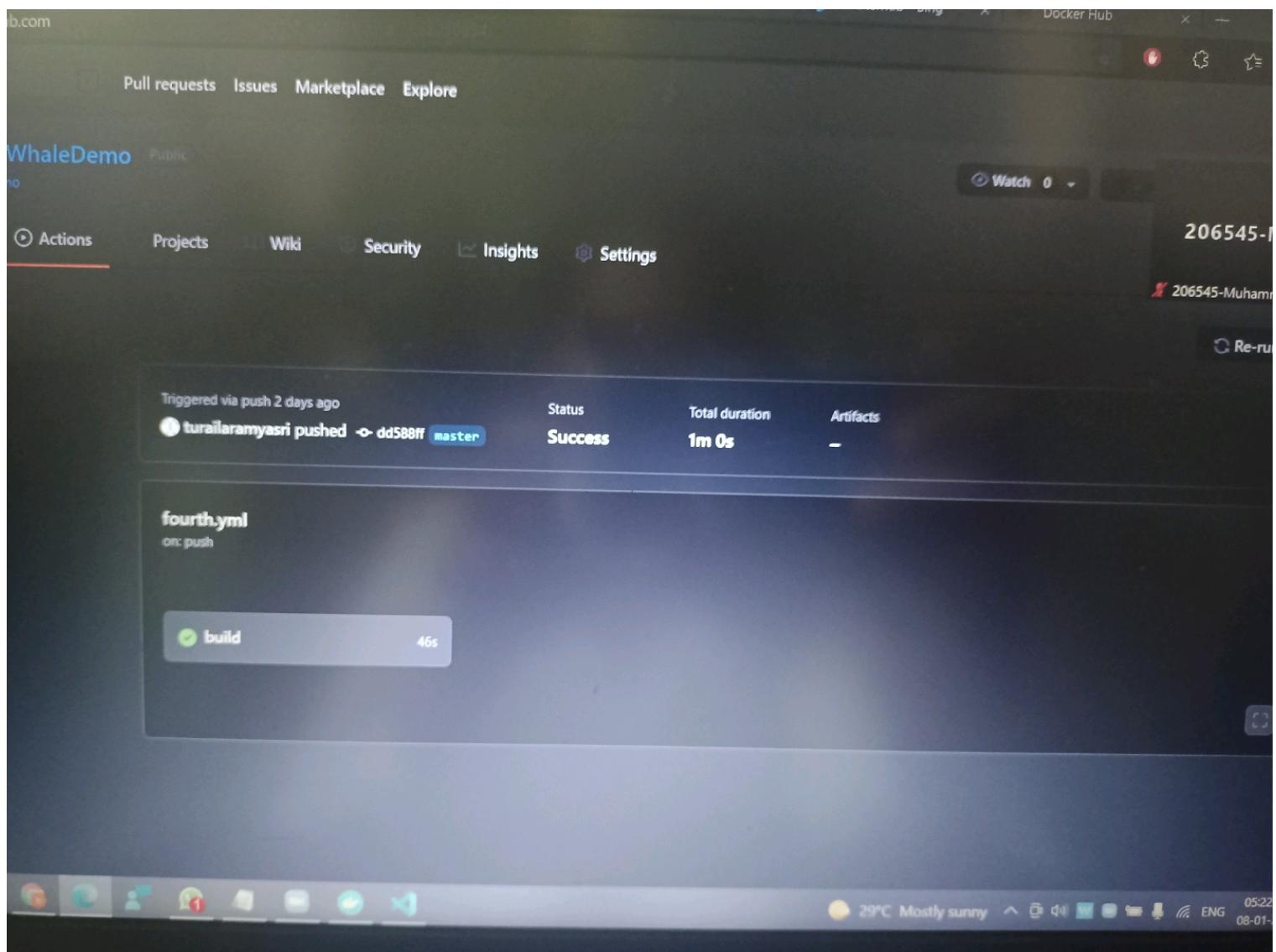
      # Allows you to run this workflow manually from the Actions tab
      workflow_dispatch:

      # A workflow run is made up of one or more jobs that can run sequentially or in parallel
      jobs:
        # This workflow contains a single job called "build"
        build:
          # The type of runner that the job will run on
          runs-on: ubuntu-latest

          # Steps represent a sequence of tasks that will be executed as part of the job
          steps:
            # Checks-out your repository under $GITHUB_WORKSPACE, so your job can access it
            - uses: actions/checkout@v2

            # Runs a single command using the runners shell
            - name: Run a one-line script
              run: echo Hello, world!

            # Runs a set of commands using the runners shell
            - name: Run a multi-line script
              run:
                echo Add other actions to build,
                echo test, and deploy your project.
```



A screenshot of a computer monitor displaying a GitHub repository settings page. The browser tab shows the URL <https://github.com/turaltaramyasi/SimpleWhaleDemo/settings/secrets/actions>. The page title is "Actions secrets". The "Settings" tab is active. On the left sidebar, the "Secrets" option is highlighted. The main content area displays information about Actions secrets, stating they are environment variables encrypted for collaborator access. It also mentions that secrets are not passed to workflows triggered by pull requests from forks. A section titled "Environment secrets" shows a message: "There are no secrets for this repository's environments." Below this, there is a link to "Manage your environments and add environment secrets". Another section titled "Repository secrets" lists two secrets: "DOCKER\_HUB\_ACCESS\_TOKEN" and "DOCKER\_HUB\_USERNAME", both updated 2 days ago, with "Update" and "Remove" buttons. The status bar at the bottom shows the URL <https://github.com/turaltaramyasi/SimpleWhaleDemo/settings/environments>.

