Assignment 3

The assignment has been completed using Docker Desktop and Nodejs.
Github Link: https://github.com/prithvia28/ECC_S24/tree/main/Assignment3

Procedure:

Container Selection

-Docker Volumes: Two volumes are created, `servervol` and `clientvol`, to persist data on the server and client containers, respectively.
- Docker Network: A private network named `prithvi` is created to connect the server and client containers.
- Exposed Port: The server exposes port 8080, allowing external connections.

Server/Client Code

- **Server:**
  - Listens on port 8080.
  - Generates a 1KB text file containing repeated text.
  - Sends the generated file to the client upon connection.
  - Calculates the MD5 checksum for the generated file and logs it.
  - The server-side code is run with the command `CMD ["node", "server.js", "0.0.0.0", "8080"]`.

- **Client:**
  - Connects to the server at the specified port and address.
  - Receives data from the server, collecting it into a buffer.
  - Concatenates the buffer and writes it to a local file.
  - Calculates the MD5 checksum for the received file to confirm its integrity.
  - The client-side code is run with the command `CMD ["node", "client.js", "server", "8080"]`.

File Transfer Mechanism

- **Server Process:**
  - The server listens for client connections.
  - Upon connection, it generates a text-based file and calculates its MD5 checksum.
  - The server sends the file data through a socket to the client using a stream.

- **Client Process:**
  - The client connects to the server and listens for incoming data.
  - It collects the data into a buffer, concatenates it, and writes it to a local file.
  - It calculates the checksum to verify the data's integrity.

Docker Configuration

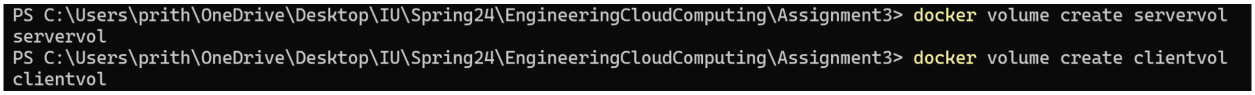
- **Build Docker Images:**
  - Build the server and client Docker images with `docker build -t server -f .\Dockerfile .` for the server and `docker build -t client -f .\Dockerfile .` for the client.

- **Run Docker Containers:**
  - The server container is run with `docker run -v servervol:/app/serverdata -p 8080:8080 -d --name server --network prithvi server`.
  - The client container is run with `docker run -v clientvol:/app/clientdata -d --name client --network prithvi client`.

With this setup, the server and client containers communicate through the private network `prithvi`, and the file transfer occurs as described above.

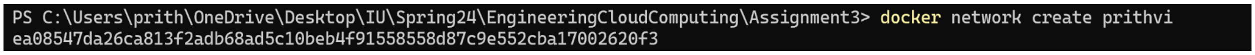
Detailed Steps:

Step 1: Create 2 volumes: servervol and clientvol



```
PS C:\Users\prith\OneDrive\Desktop\IU\Spring24\EngineeringCloudComputing\Assignment3> docker volume create servervol
servervol
PS C:\Users\prith\OneDrive\Desktop\IU\Spring24\EngineeringCloudComputing\Assignment3> docker volume create clientvol
clientvol
```

Step 2: Creating a private network in the docker so both containers can be connected there.



```
PS C:\Users\prith\OneDrive\Desktop\IU\Spring24\EngineeringCloudComputing\Assignment3> docker network create prithvi
ea08547da26ca813f2adb68ad5c10beb4f91558558d87c9e552cba17002620f3
```

Step 3: Codes for server.js and client.js

Server.js:

```
const net = require('net');
const fs = require('fs');
const crypto = require('crypto');

const HOST = process.argv[2] || "localhost";
const PORT = process.argv[3] || 8080;


const server = net.createServer((socket) => {
  console.log(`Client connected: ${socket.remoteAddress}`);

  // Generate 1KB of text data
  const textData = "This is a text-based 1KB file. ".repeat(32); // Approx. 1KB

  // Write the text data to a file
  const fileName = 'text_file.txt';
  fs.writeFileSync(fileName, textData);

  // Calculate the MD5 checksum
  const hasher = crypto.createHash('md5');
  hasher.update(textData);
  const checksum = hasher.digest('hex');

  console.log(`Checksum for the generated file: ${checksum}`);

  // Send the file to the client
  const fileStream = fs.createReadStream(fileName);
  fileStream.pipe(socket);

  socket.on('end', () => {
    console.log(`Client disconnected: ${socket.remoteAddress}`);
  });

  socket.on('error', (err) => {
    console.error(`Error: ${err.message}`);
  });
});

server.on("error", (err) => {
  console.error("Server error:", err);
});

server.listen(PORT, HOST, () => {
  console.log(`Server started on ${HOST}:${PORT}`);
});
```

Client.js:

```
const net = require('net');
const fs = require('fs');
const crypto = require('crypto');

const SERVER_ADDR = process.argv[2] || "localhost";
const PORT = process.argv[3] || 8080;

const client = new net.Socket();

client.connect(PORT, SERVER_ADDR, () => {
  console.log(`Connected to server at ${SERVER_ADDR}:${PORT}`);

  const receivedData = [];

  client.on('data', (data) => {
    receivedData.push(data); // Collect received data
    console.log(`Received data from server: ${data.toString()}`); // Log received data
  });

  client.on('end', () => {
    const buffer = Buffer.concat(receivedData); // Concatenate all received data

    // Save the data to a file
    fs.writeFileSync('received_file.txt', buffer);

    // Calculate the MD5 checksum
    const hasher = crypto.createHash('md5');
    hasher.update(buffer);
    const checksum = hasher.digest('hex');

    console.log(`Received a file with checksum: ${checksum}`);
  });
  client.on("end", () => {
    console.log("Disconnected from server");
  });

  client.on('error', (err) => {
    console.error(`Error: ${err.message}`);
  });
});
```

We will be sending a 1Kb file which says “This is a text-based 1KB file”.

```
const textData = "This is a text-based 1KB file. ".repeat(32); // Approx. 1KB
```

Step 4: Create Dockerfiles for server and client

Server:

```
# Use the official Node.js 14.17.0 image
FROM node:14.17.0-alpine

# Set the working directory inside the Docker container
WORKDIR /app

# Copy the server script into the working directory
COPY server.js /app/server.js

# Set executable permissions for the server.js file
RUN chmod +x /app/server.js

# create a dir for serverdata
RUN mkdir /app/serverdata

# Expose the port on which your server will run
EXPOSE 8080

# Command to run the server
CMD ["node", "server.js" , "0.0.0.0", "8080"]
```

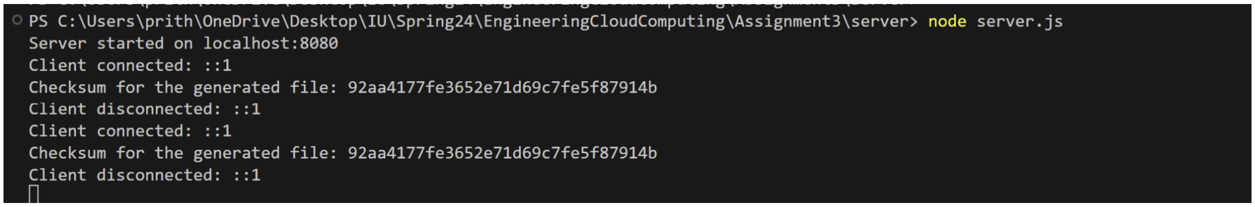
Client:

```
# Use the official Node.js 14.17.0 image
FROM node:14.17.0-alpine

# Set the working directory inside the Docker container
WORKDIR /app

# Copy the client script into the working directory
COPY client.js /app/client.js

# Set executable permissions for the client.js file
RUN chmod +x /app/client.js

# Create a directory for the client
```

```
RUN mkdir /app/clientdata

# Command to run the client
CMD ["node", "client.js", "server", "8080"]
```

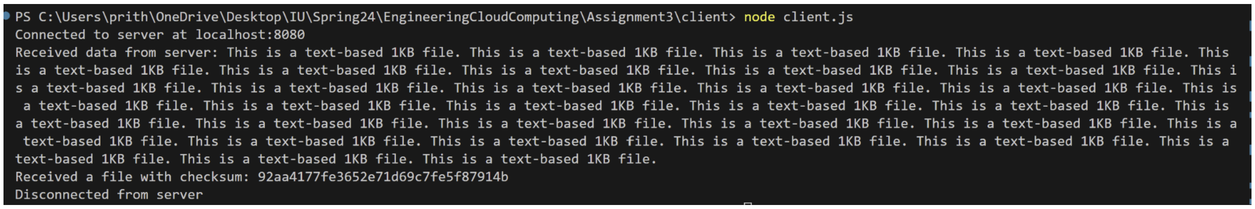
Step 5: Testing on local file system:

Server:



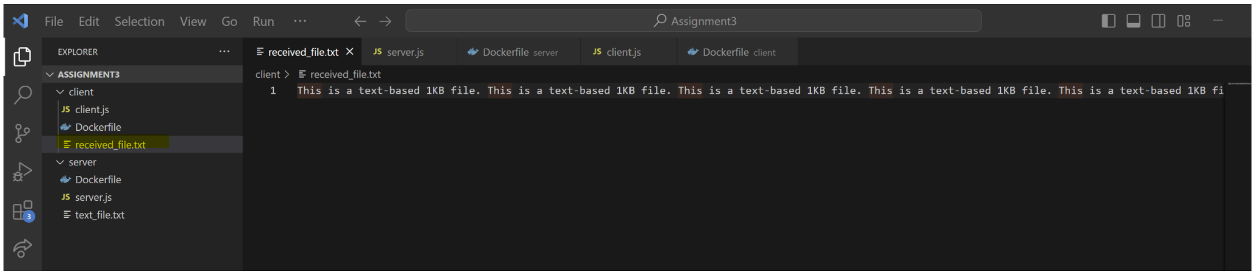
```
PS C:\Users\prith\OneDrive\Desktop\IU\Spring24\EngineeringCloudComputing\Assignment3\server> node server.js
Server started on localhost:8080
Client connected: ::1
Checksum for the generated file: 92aa4177fe3652e71d69c7fe5f87914b
Client disconnected: ::1
Client connected: ::1
Checksum for the generated file: 92aa4177fe3652e71d69c7fe5f87914b
Client disconnected: ::1
```

Client:



```
PS C:\Users\prith\OneDrive\Desktop\IU\Spring24\EngineeringCloudComputing\Assignment3\client> node client.js
Connected to server at localhost:8080
Received data from server: This is a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. This
is a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. This i
s a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. This is
 a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. This is
a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. This is a
 text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. This is a
text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file.
Received a file with checksum: 92aa4177fe3652e71d69c7fe5f87914b
Disconnected from server
```

Received file from server:



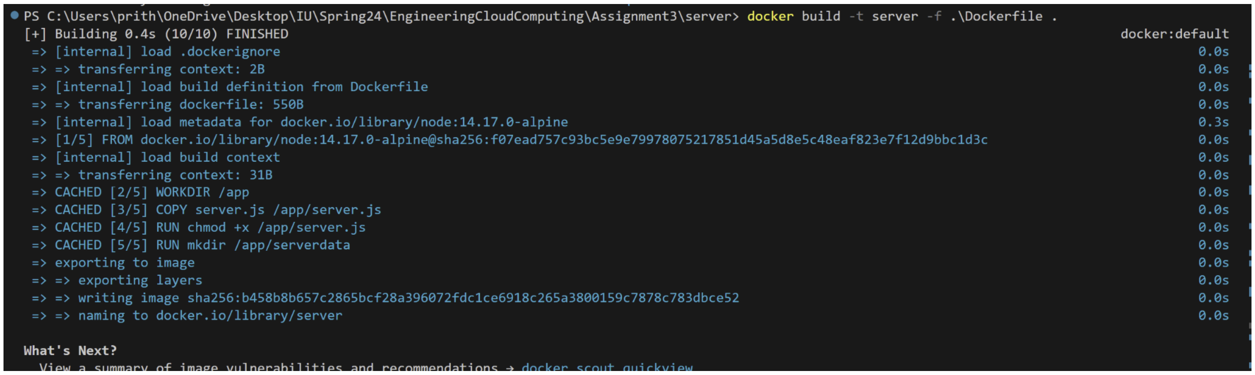
```
1  This is a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB fi
```

Step 6: Build the docker image

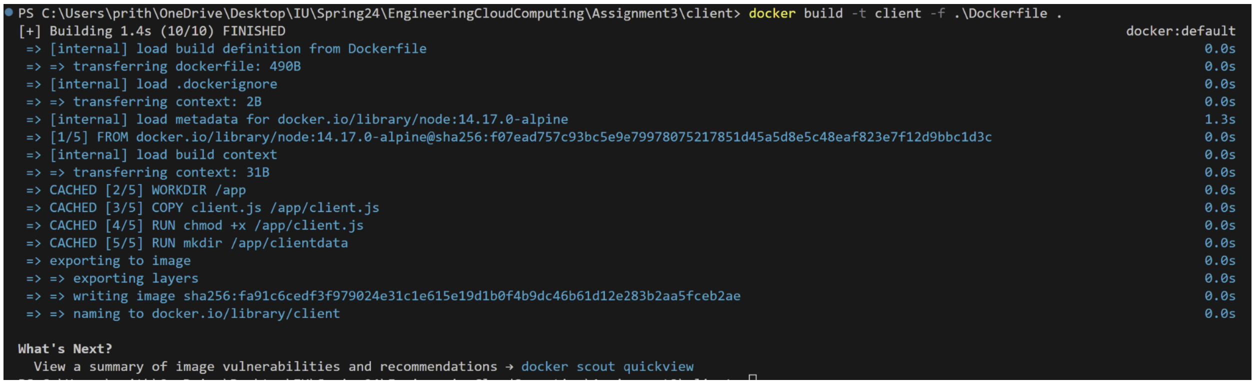
Commands:

Server: docker build -t server -f .\Dockerfile .

Client: docker build -t client -f .\Dockerfile .

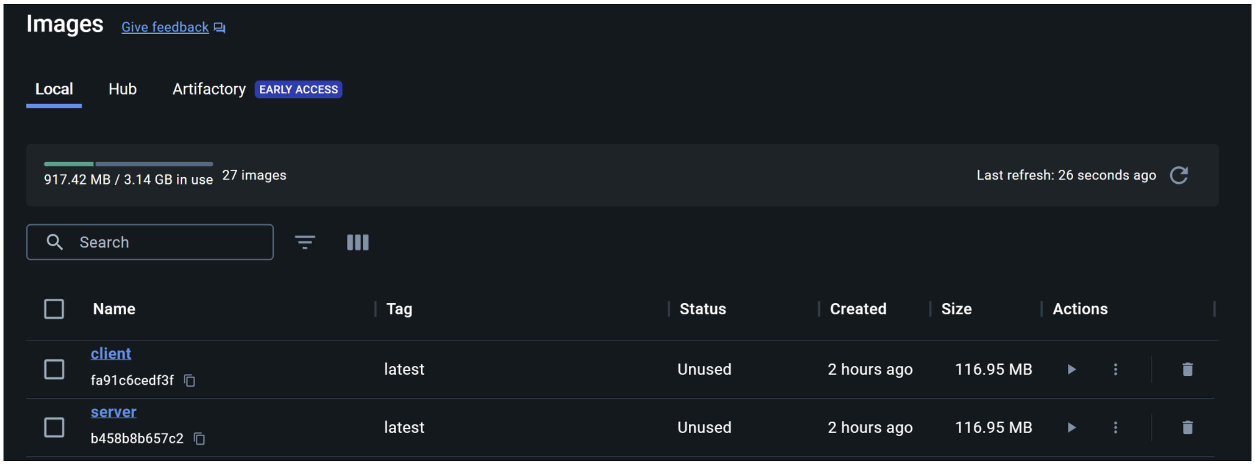


```
PS C:\Users\prith\OneDrive\Desktop\IU\Spring24\EngineeringCloudComputing\Assignment3\server> docker build -t server -f .\Dockerfile .
 [+] Building 0.4s (10/10) FINISHED                                                                                              docker:default
  => [internal] load .dockerignore                                                                                                          0.0s
  => => transferring context: 2B                                                                                                            0.0s
  => [internal] load build definition from Dockerfile                                                                                       0.0s
  => => transferring dockerfile: 550B                                                                                                       0.0s
  => [internal] load metadata for docker.io/library/node:14.17.0-alpine                                                                     0.3s
  => [1/5] FROM docker.io/library/node:14.17.0-alpine@sha256:f07ead757c93bc5e9e79978075217851d45a5d8e5c48eaf823e7f12d9bbc1d3c               0.0s
  => [internal] load build context                                                                                                          0.0s
  => => transferring context: 31B                                                                                                           0.0s
  => CACHED [2/5] WORKDIR /app                                                                                                              0.0s
  => CACHED [3/5] COPY server.js /app/server.js                                                                                             0.0s
  => CACHED [4/5] RUN chmod +x /app/server.js                                                                                               0.0s
  => CACHED [5/5] RUN mkdir /app/serverdata                                                                                                 0.0s
  => exporting to image                                                                                                                     0.0s
  => => exporting layers                                                                                                                    0.0s
  => => writing image sha256:b458b8b657c2865bcf28a396072fdc1ce6918c265a3800159c7878c783dbce52                                               0.0s
  => => naming to docker.io/library/server                                                                                                  0.0s

 What's Next?
   View a summary of image vulnerabilities and recommendations → docker scout quickview
```



```
PS C:\Users\prith\OneDrive\Desktop\IU\Spring24\EngineeringCloudComputing\Assignment3\client> docker build -t client -f .\Dockerfile .
[+] Building 1.4s (10/10) FINISHED                                                                                     docker:default
 => [internal] load build definition from Dockerfile                                                                             0.0s
 => => transferring dockerfile: 490B                                                                                             0.0s
 => [internal] load .dockerignore                                                                                                0.0s
 => => transferring context: 2B                                                                                                  0.0s
 => [internal] load metadata for docker.io/library/node:14.17.0-alpine                                                           1.3s
 => [1/5] FROM docker.io/library/node:14.17.0-alpine@sha256:f07ead757c93bc5e9e79978075217851d45a5d8e5c48eaf823e7f12d9bbc1d3c     0.0s
 => [internal] load build context                                                                                                0.0s
 => => transferring context: 31B                                                                                                 0.0s
 => CACHED [2/5] WORKDIR /app                                                                                                    0.0s
 => CACHED [3/5] COPY client.js /app/client.js                                                                                   0.0s
 => CACHED [4/5] RUN chmod +x /app/client.js                                                                                     0.0s
 => CACHED [5/5] RUN mkdir /app/clientdata                                                                                       0.0s
 => exporting to image                                                                                                           0.0s
 => => exporting layers                                                                                                          0.0s
 => => writing image sha256:fa91c6cedf3f979024e31c1e615e19d1b0f4b9dc46b61d12e283b2aa5fceb2ae                                     0.0s
 => => naming to docker.io/library/client                                                                                        0.0s

What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview
```

Check the images on Docker:



917.42 MB / 3.14 GB in use 27 images

Last refresh: 26 seconds ago

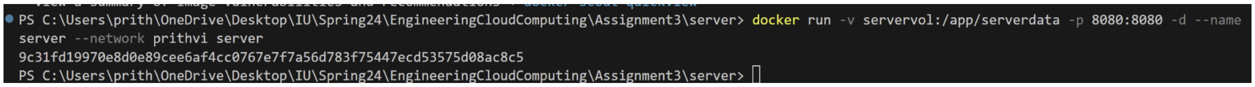
□	Name	Tag	Status	Created	Size	Actions
□	client fa91c6cedf3f	latest	Unused	2 hours ago	116.95 MB	▶ ⋮ 🗑
□	server b458b8b657c2	latest	Unused	2 hours ago	116.95 MB	▶ ⋮ 🗑

Step 7: Create container and run on the docker

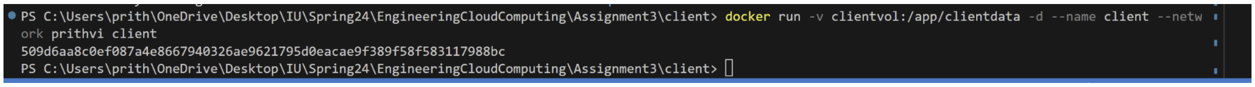
Commands:

Server: docker run -v servervol:/app/serverdata -p 8080:8080 -d --name server --network prithvi server

Client: docker run -v clientvol:/app/clientdata -d --name client --network prithvi client

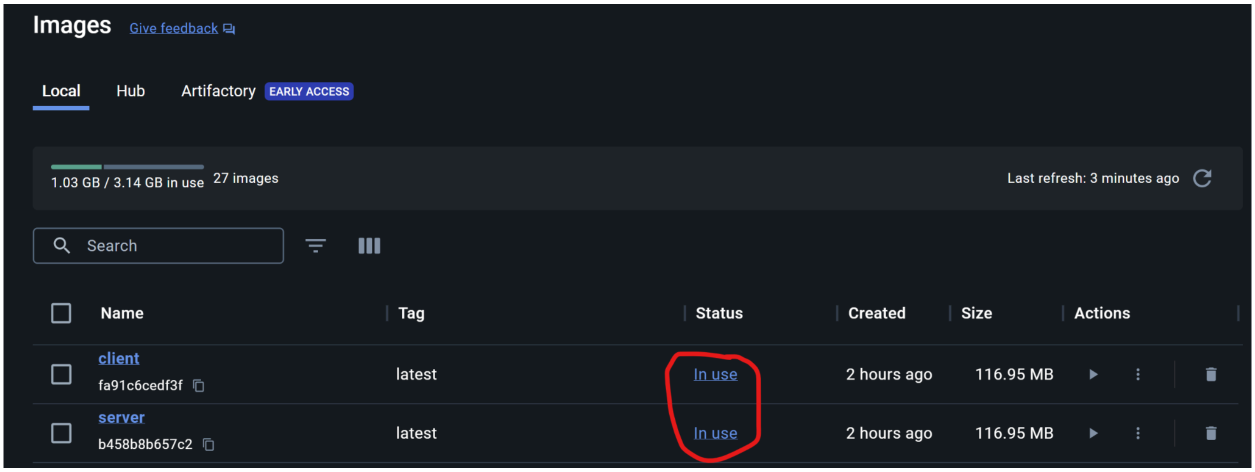


```
PS C:\Users\prith\OneDrive\Desktop\IU\Spring24\EngineeringCloudComputing\Assignment3\server> docker run -v servervol:/app/serverdata -p 8080:8080 -d --name
server --network prithvi server
9c31fd19970e8d0e89cee6af4cc0767e7f7a56d783f75447ecd53575d08ac8c5
PS C:\Users\prith\OneDrive\Desktop\IU\Spring24\EngineeringCloudComputing\Assignment3\server>
```



```
PS C:\Users\prith\OneDrive\Desktop\IU\Spring24\EngineeringCloudComputing\Assignment3\client> docker run -v clientvol:/app/clientdata -d --name client --netw
ork prithvi client
509d6aa8c0ef087a4e8667940326ae9621795d0eacae9f389f58f583117988bc
PS C:\Users\prith\OneDrive\Desktop\IU\Spring24\EngineeringCloudComputing\Assignment3\client>
```

Check the images on Docker:

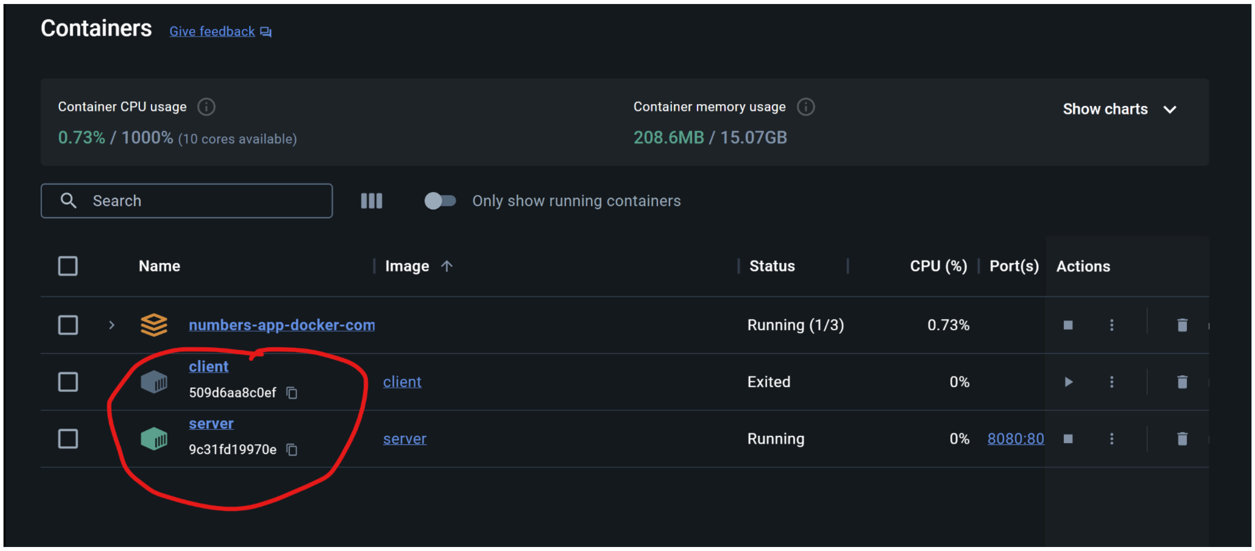


1.03 GB / 3.14 GB in use 27 images

Last refresh: 3 minutes ago

	Name	Tag	Status	Created	Size	Actions
□	client fa91c6cedf3f	latest	In use	2 hours ago	116.95 MB	
□	server b458b8b657c2	latest	In use	2 hours ago	116.95 MB	

Check the containers on Docker:



Container CPU usage
0.73% / 1000% (10 cores available)

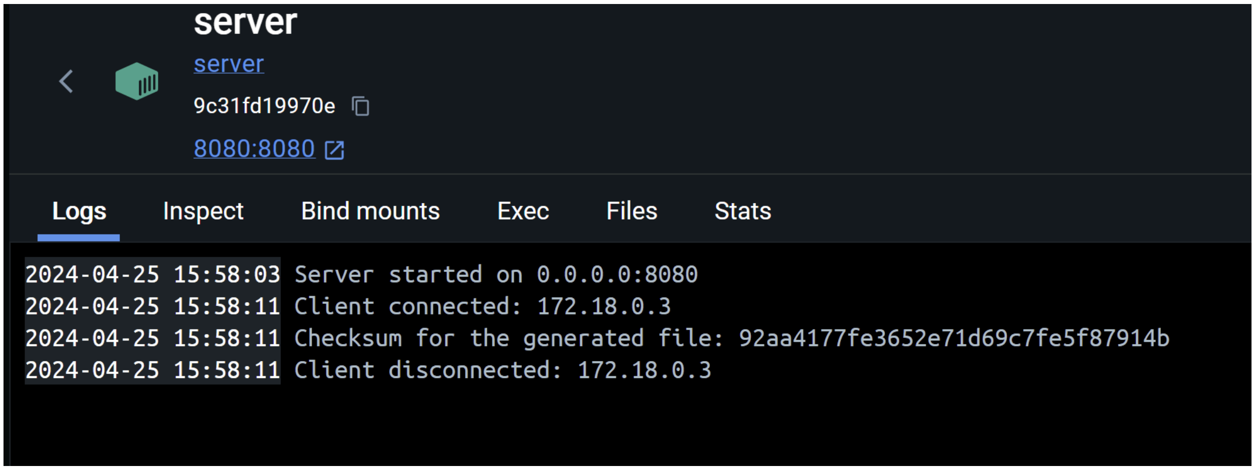
Container memory usage
208.6MB / 15.07GB

	Name	Image	Status	CPU (%)	Port(s)	Actions
□	numbers-app-docker-com		Running (1/3)	0.73%		
□	client 509d6aa8c0ef	client	Exited	0%		
□	server 9c31fd19970e	server	Running	0%	8080:80	

Client has status exited because the execution is complete.

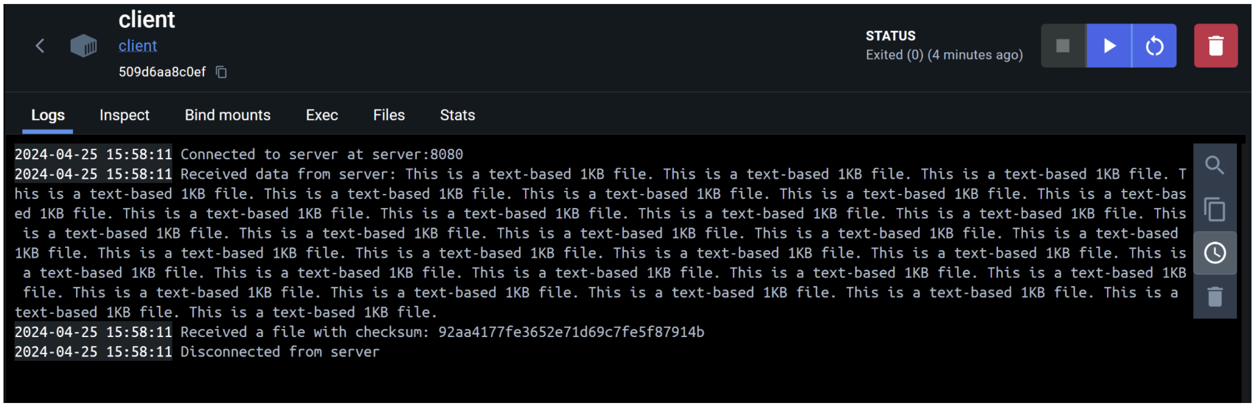
Step 8: Check the logs on Docker

Server:



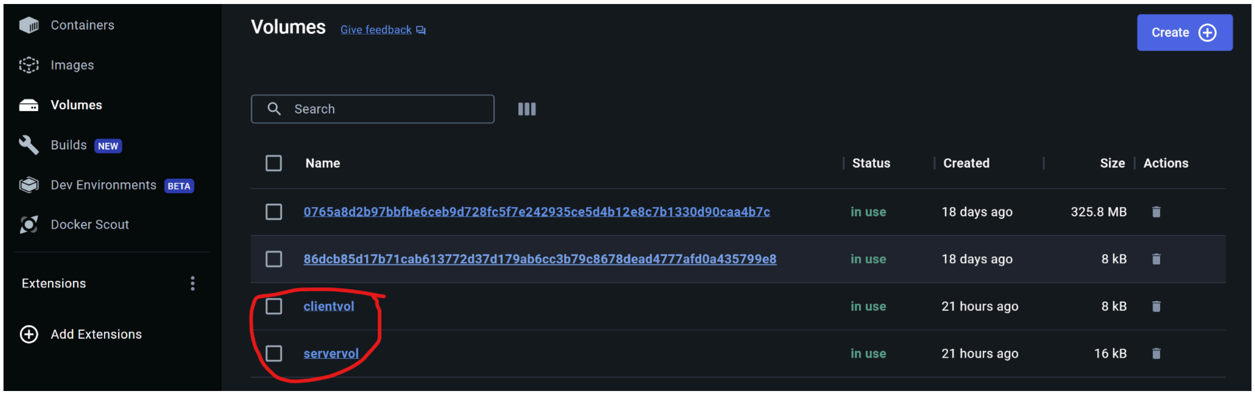
```
2024-04-25 15:58:03 Server started on 0.0.0.0:8080
2024-04-25 15:58:11 Client connected: 172.18.0.3
2024-04-25 15:58:11 Checksum for the generated file: 92aa4177fe3652e71d69c7fe5f87914b
2024-04-25 15:58:11 Client disconnected: 172.18.0.3
```

Client:

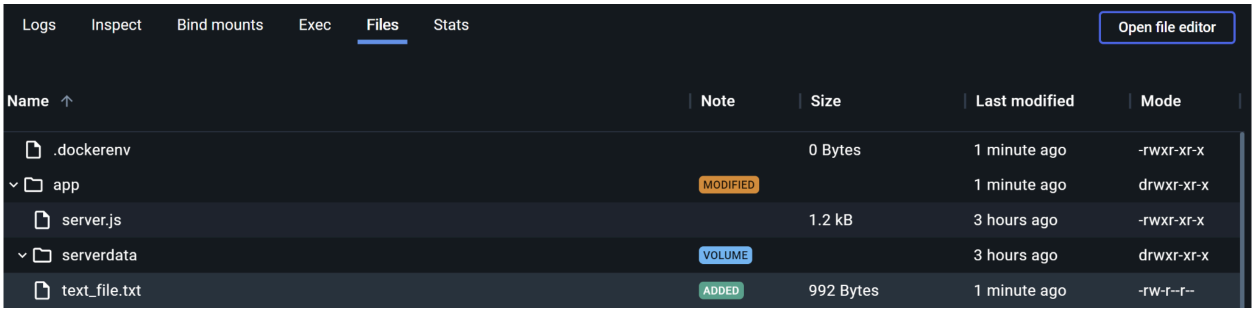


```
2024-04-25 15:58:11 Connected to server at server:8080
2024-04-25 15:58:11 Received data from server: This is a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. T
his is a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. This is a text-bas
ed 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. This
 is a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. This is a text-based
1KB file. This is a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. This is
 a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB
 file. This is a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. This is a
text-based 1KB file. This is a text-based 1KB file.
2024-04-25 15:58:11 Received a file with checksum: 92aa4177fe3652e71d69c7fe5f87914b
2024-04-25 15:58:11 Disconnected from server
```

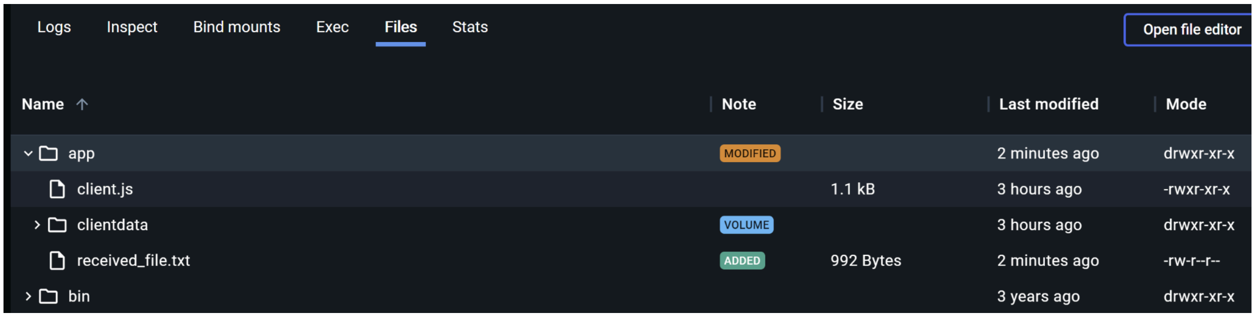
Step 9: Check the output in Files and Volumes:



☐	Name	Status	Created	Size	Actions
☐	0765a8d2b97bbfbe6ceb9d728fc5f7e242935ce5d4b12e8c7b1330d90caa4b7c	in use	18 days ago	325.8 MB	🗑
☐	86dcb85d17b71cab613772d37d179ab6cc3b79c8678dead4777afd0a435799e8	in use	18 days ago	8 kB	🗑
☐	clientvol	in use	21 hours ago	8 kB	🗑
☐	servervol	in use	21 hours ago	16 kB	🗑



Name	Note	Size	Last modified	Mode
.dockerenv		0 Bytes	1 minute ago	-rwxr-xr-x
app	MODIFIED		1 minute ago	drwxr-xr-x
server.js		1.2 kB	3 hours ago	-rwxr-xr-x
serverdata	VOLUME		3 hours ago	drwxr-xr-x
text_file.txt	ADDED	992 Bytes	1 minute ago	-rw-r--r--



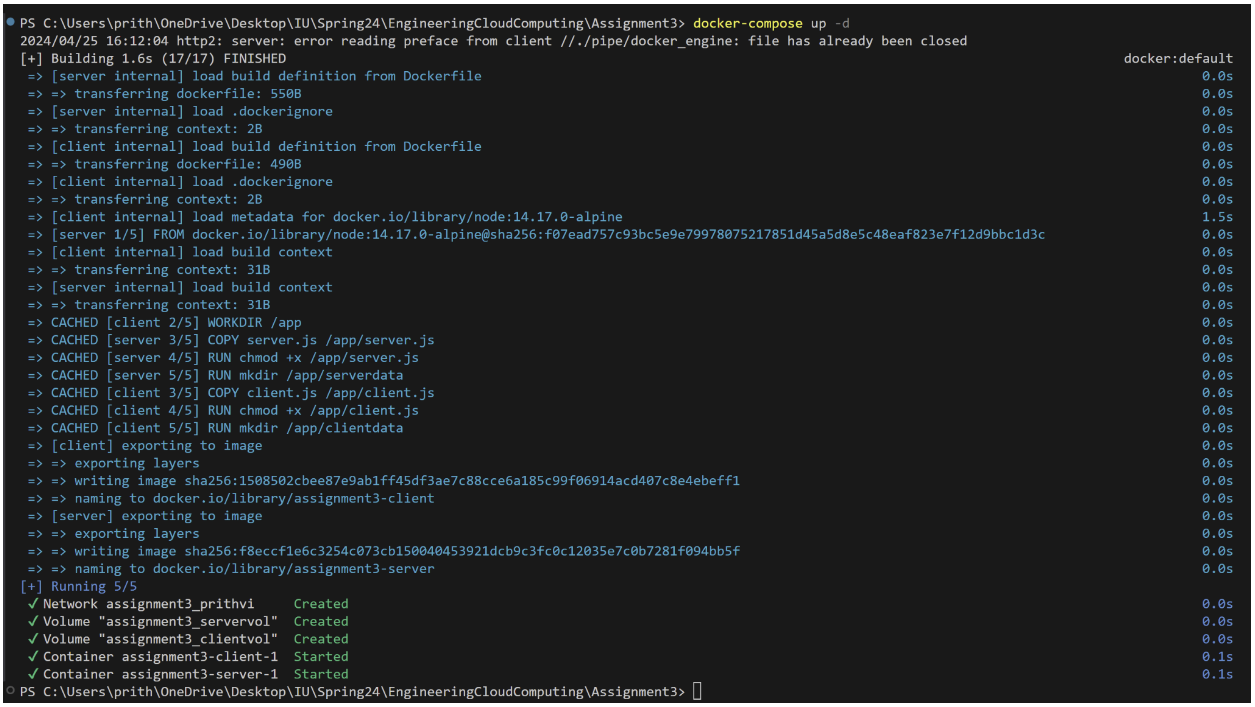
Name	Note	Size	Last modified	Mode
app	MODIFIED		2 minutes ago	drwxr-xr-x
client.js		1.1 kB	3 hours ago	-rwxr-xr-x
clientdata	VOLUME		3 hours ago	drwxr-xr-x
received_file.txt	ADDED	992 Bytes	2 minutes ago	-rw-r--r--
bin			3 years ago	drwxr-xr-x

Step 10: Automating using docker-compose.yml

Docker-compose.yml:

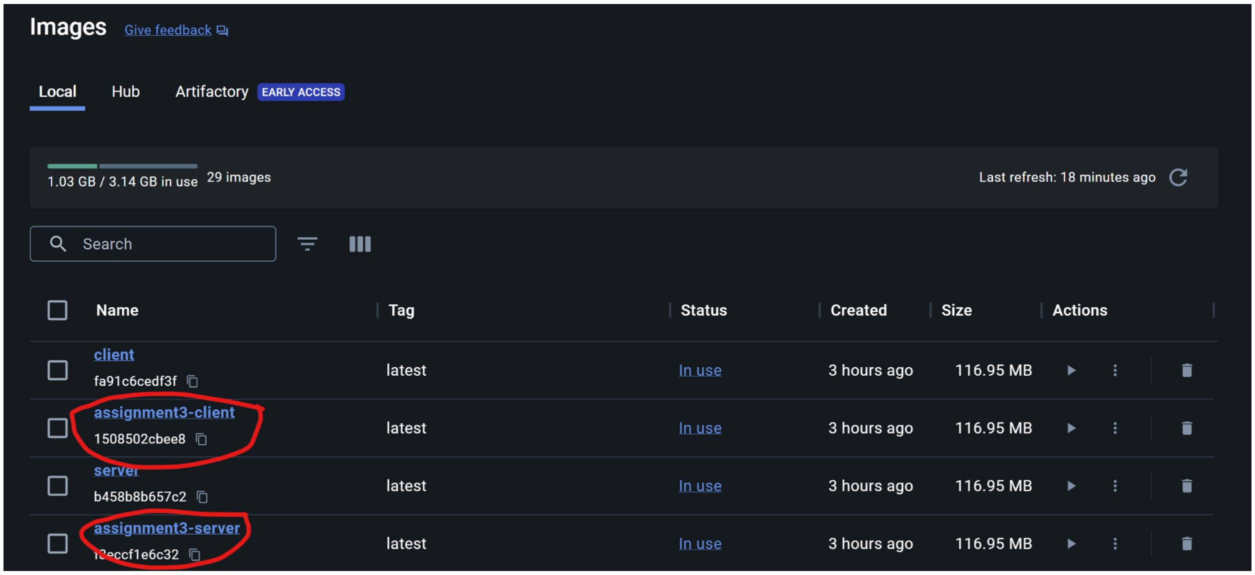
```
version: "3"

services:
  server:
    build:
      context: ./server
      dockerfile: Dockerfile
    networks:
      - prithvi
    volumes:
      - servervol:/app/serverdata

  client:
    build:
      context: ./client
      dockerfile: Dockerfile
    volumes:
      - clientvol:/app/clientdata
    networks:
      - prithvi

volumes:
  servervol:
  clientvol:

networks:
  prithvi:
```

Command: *docker-compose up -d*



```
PS C:\Users\prith\OneDrive\Desktop\IU\Spring24\EngineeringCloudComputing\Assignment3> docker-compose up -d
2024/04/25 16:12:04 http2: server: error reading preface from client //./pipe/docker_engine: file has already been closed
[+] Building 1.6s (17/17) FINISHED                                                                                                           docker:default
 => [server internal] load build definition from Dockerfile                                                                                            0.0s
 => => transferring dockerfile: 550B                                                                                                                   0.0s
 => [server internal] load .dockerignore                                                                                                               0.0s
 => => transferring context: 2B                                                                                                                        0.0s
 => [client internal] load build definition from Dockerfile                                                                                            0.0s
 => => transferring dockerfile: 490B                                                                                                                   0.0s
 => [client internal] load .dockerignore                                                                                                               0.0s
 => => transferring context: 2B                                                                                                                        0.0s
 => [client internal] load metadata for docker.io/library/node:14.17.0-alpine                                                                          1.5s
 => [server 1/5] FROM docker.io/library/node:14.17.0-alpine@sha256:f07ead757c93bc5e9e79978075217851d45a5d8e5c48eaf823e7f12d9bbc1d3c                    0.0s
 => [client internal] load build context                                                                                                               0.0s
 => => transferring context: 31B                                                                                                                       0.0s
 => [server internal] load build context                                                                                                               0.0s
 => => transferring context: 31B                                                                                                                       0.0s
 => CACHED [client 2/5] WORKDIR /app                                                                                                                   0.0s
 => CACHED [server 3/5] COPY server.js /app/server.js                                                                                                  0.0s
 => CACHED [server 4/5] RUN chmod +x /app/server.js                                                                                                    0.0s
 => CACHED [server 5/5] RUN mkdir /app/serverdata                                                                                                      0.0s
 => CACHED [client 3/5] COPY client.js /app/client.js                                                                                                  0.0s
 => CACHED [client 4/5] RUN chmod +x /app/client.js                                                                                                    0.0s
 => CACHED [client 5/5] RUN mkdir /app/clientdata                                                                                                      0.0s
 => [client] exporting to image                                                                                                                        0.0s
 => => exporting layers                                                                                                                                0.0s
 => => writing image sha256:1508502cbee87e9ab1ff45df3ae7c88cce6a185c99f06914acd407c8e4ebeff1                                                           0.0s
 => => naming to docker.io/library/assignment3-client                                                                                                  0.0s
 => [server] exporting to image                                                                                                                        0.0s
 => => exporting layers                                                                                                                                0.0s
 => => writing image sha256:f8eccf1e6c3254c073cb150040453921dcb9c3fc0c12035e7c0b7281f094bb5f                                                           0.0s
 => => naming to docker.io/library/assignment3-server                                                                                                  0.0s
[+] Running 5/5
 ✓ Network assignment3_prithvi     Created                                                                                                             0.0s
 ✓ Volume "assignment3_servervol"  Created                                                                                                             0.0s
 ✓ Volume "assignment3_clientvol"  Created                                                                                                             0.0s
 ✓ Container assignment3-client-1  Started                                                                                                             0.1s
 ✓ Container assignment3-server-1  Started                                                                                                             0.1s
PS C:\Users\prith\OneDrive\Desktop\IU\Spring24\EngineeringCloudComputing\Assignment3> 
```

Images:

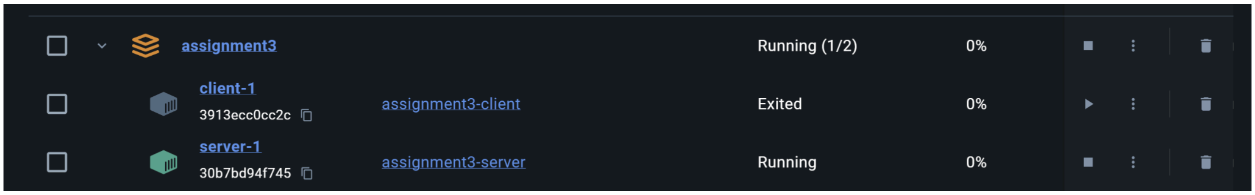


1.03 GB / 3.14 GB in use 29 images

Last refresh: 18 minutes ago

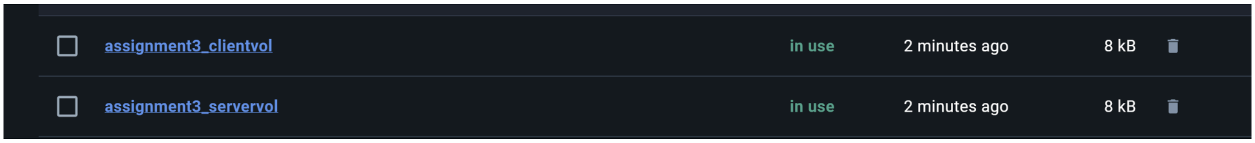
☐	Name	Tag	Status	Created	Size	Actions
☐	client fa91c6cedf3f	latest	In use	3 hours ago	116.95 MB	▶ ⋮ 🗑
☐	assignment3-client 1508502cbee8	latest	In use	3 hours ago	116.95 MB	▶ ⋮ 🗑
☐	server b458b8b657c2	latest	In use	3 hours ago	116.95 MB	▶ ⋮ 🗑
☐	assignment3-server f8eccf1e6c32	latest	In use	3 hours ago	116.95 MB	▶ ⋮ 🗑

Containers:



☐	assignment3		Running (1/2)	0%	■ ⋮ 🗑
☐	client-1 3913ecc0cc2c	assignment3-client	Exited	0%	▶ ⋮ 🗑
☐	server-1 30b7bd94f745	assignment3-server	Running	0%	■ ⋮ 🗑

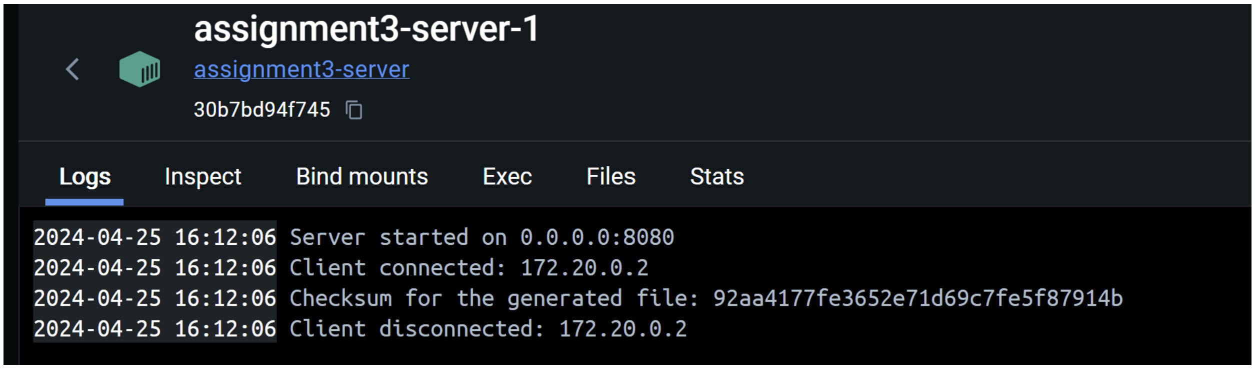
Volumes:



□	assignment3_clientvol	in use	2 minutes ago	8 kB
□	assignment3_servervol	in use	2 minutes ago	8 kB

Outputs:

Server:

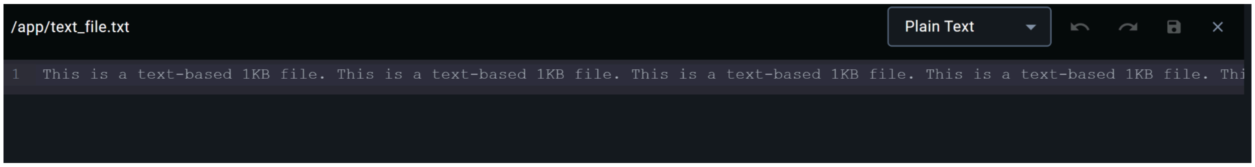


assignment3-server-1

assignment3-server

30b7bd94f745

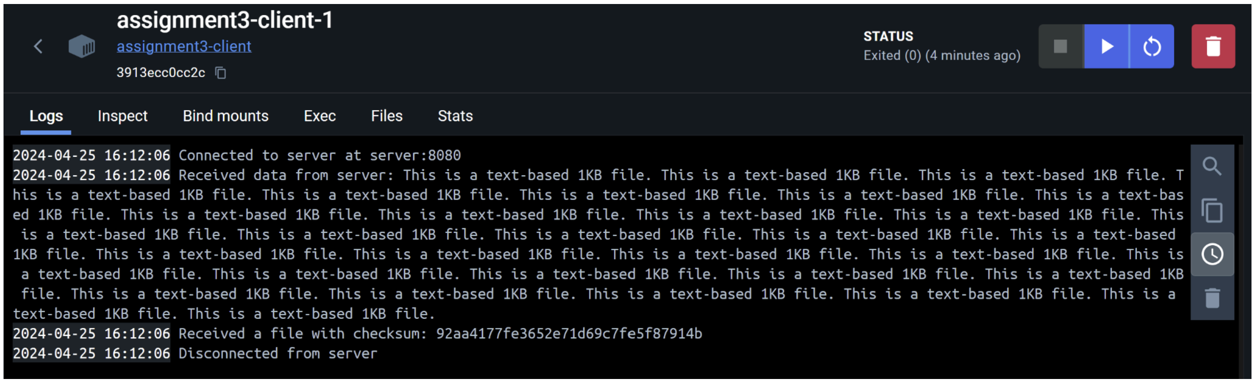
```
2024-04-25 16:12:06 Server started on 0.0.0.0:8080
2024-04-25 16:12:06 Client connected: 172.20.0.2
2024-04-25 16:12:06 Checksum for the generated file: 92aa4177fe3652e71d69c7fe5f87914b
2024-04-25 16:12:06 Client disconnected: 172.20.0.2
```



/app/text_file.txt

```
This is a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. Thi
```

Client:



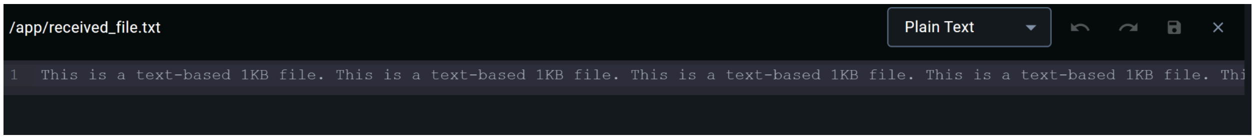
assignment3-client-1

assignment3-client

3913ecc0cc2c

STATUS
Exited (0) (4 minutes ago)

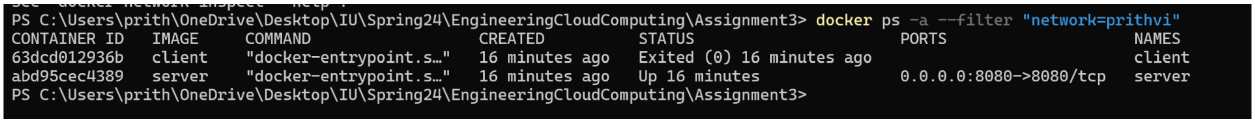
```
2024-04-25 16:12:06 Connected to server at server:8080
2024-04-25 16:12:06 Received data from server: This is a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. T
his is a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. This is a text-bas
ed 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. This
 is a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. This is a text-based
1KB file. This is a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. This is
 a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB
 file. This is a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. This is a
text-based 1KB file. This is a text-based 1KB file.
2024-04-25 16:12:06 Received a file with checksum: 92aa4177fe3652e71d69c7fe5f87914b
2024-04-25 16:12:06 Disconnected from server
```



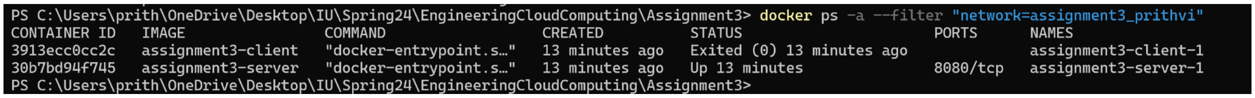
/app/received_file.txt

```
This is a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. This is a text-based 1KB file. Thi
```

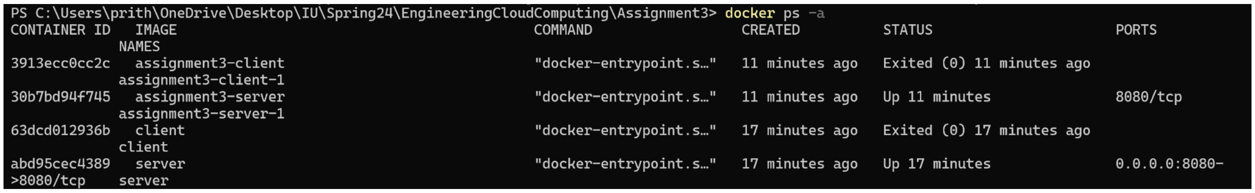
Step 11: Checking from CLI



```
PS C:\Users\prith\OneDrive\Desktop\IU\Spring24\EngineeringCloudComputing\Assignment3> docker ps -a --filter "network=prithvi"
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS                      PORTS                    NAMES
63dcd012936b   client    "docker-entrypoint.s…"   16 minutes ago   Exited (0) 16 minutes ago                            client
abd95cec4389   server    "docker-entrypoint.s…"   16 minutes ago   Up 16 minutes               0.0.0.0:8080->8080/tcp   server
PS C:\Users\prith\OneDrive\Desktop\IU\Spring24\EngineeringCloudComputing\Assignment3>
```



```
PS C:\Users\prith\OneDrive\Desktop\IU\Spring24\EngineeringCloudComputing\Assignment3> docker ps -a --filter "network=assignment3_prithvi"
CONTAINER ID   IMAGE                 COMMAND                  CREATED          STATUS                      PORTS      NAMES
3913ecc0cc2c   assignment3-client   "docker-entrypoint.s…"   13 minutes ago   Exited (0) 13 minutes ago              assignment3-client-1
30b7bd94f745   assignment3-server   "docker-entrypoint.s…"   13 minutes ago   Up 13 minutes               8080/tcp   assignment3-server-1
PS C:\Users\prith\OneDrive\Desktop\IU\Spring24\EngineeringCloudComputing\Assignment3>
```



```
PS C:\Users\prith\OneDrive\Desktop\IU\Spring24\EngineeringCloudComputing\Assignment3> docker ps -a
CONTAINER ID   IMAGE                                          COMMAND                  CREATED          STATUS                      PORTS
             NAMES
3913ecc0cc2c   assignment3-client                             "docker-entrypoint.s…"   11 minutes ago   Exited (0) 11 minutes ago
             assignment3-client-1
30b7bd94f745   assignment3-server                             "docker-entrypoint.s…"   11 minutes ago   Up 11 minutes               8080/tcp
             assignment3-server-1
63dcd012936b   client                                         "docker-entrypoint.s…"   17 minutes ago   Exited (0) 17 minutes ago
             client
abd95cec4389   server                                         "docker-entrypoint.s…"   17 minutes ago   Up 17 minutes               0.0.0.0:8080-
>8080/tcp    server
```