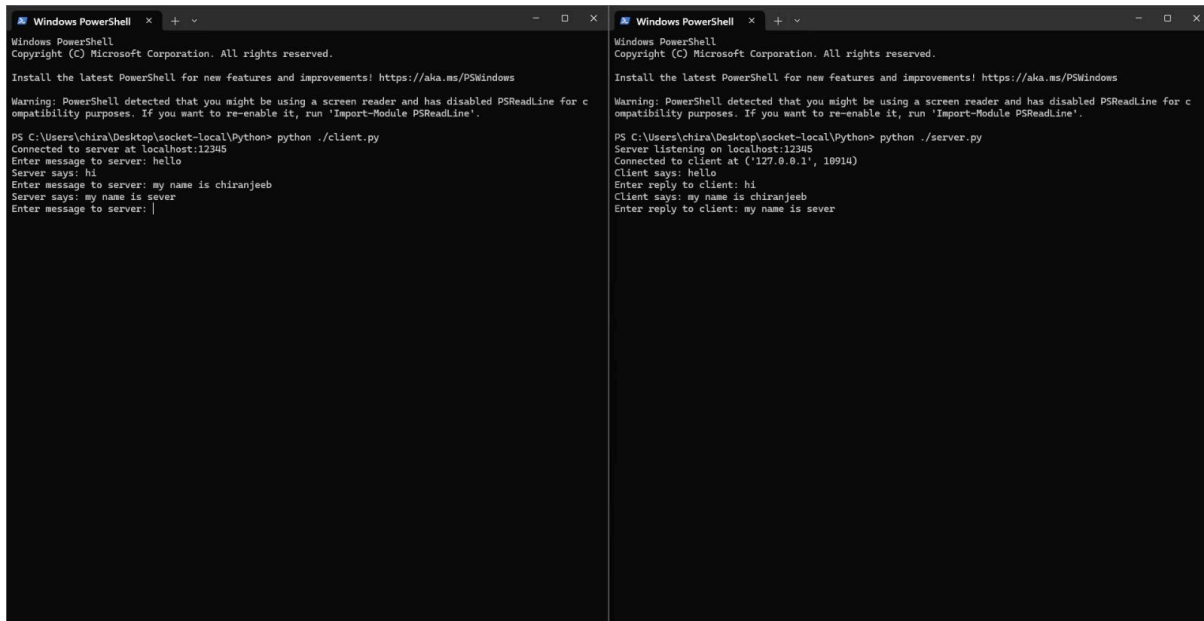


PYTHON LOCAL-



The image shows two side-by-side Windows PowerShell windows. The left window shows the execution of a Python client script (client.py) which connects to a server at localhost:12345 and sends the message 'hello'. The right window shows the execution of a Python server script (server.py) which listens on localhost:12345, receives the message 'hello' from the client, and responds with 'hi'. Both windows show the standard PowerShell header and a warning about PSReadLine.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

Warning: PowerShell detected that you might be using a screen reader and has disabled PSReadLine for compatibility purposes. If you want to re-enable it, run 'Import-Module PSReadLine'.

PS C:\Users\chira\Desktop\socket-local\Python> python .\client.py
Connected to server at localhost:12345
Enter message to server: hello
Server says: hi
Enter message to server: my name is chiranjeeb
Server says: my name is sever
Enter message to server: |

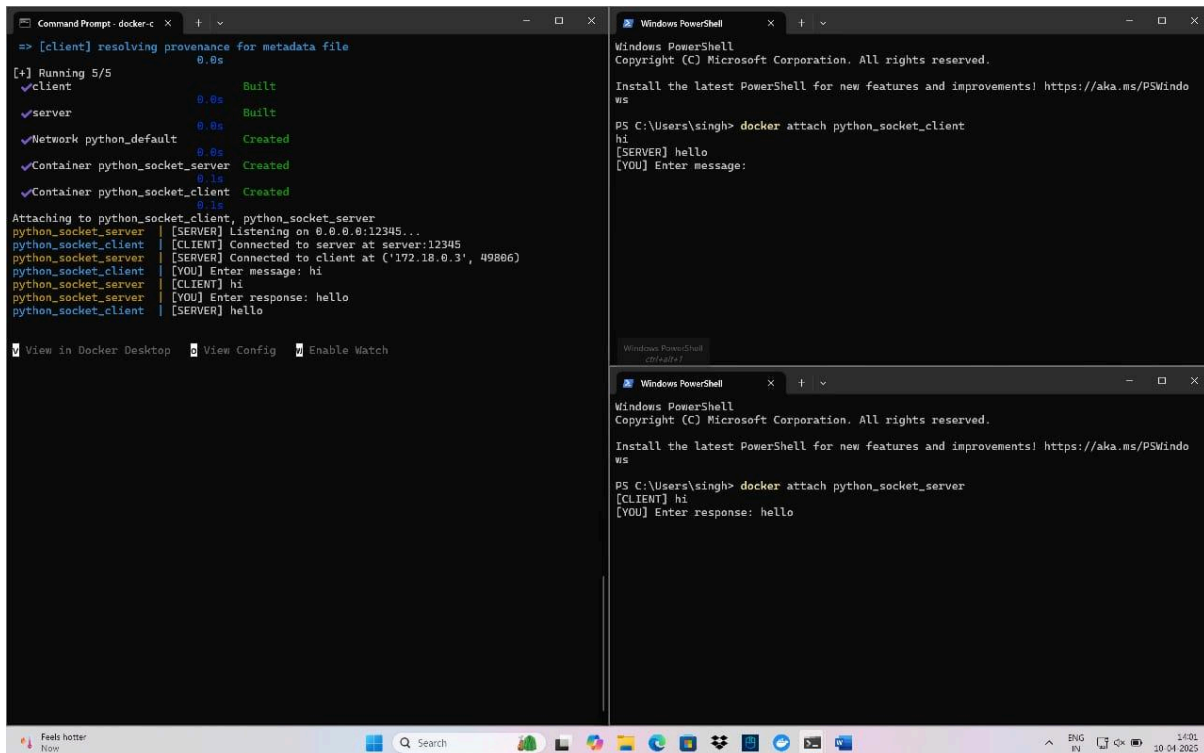
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

Warning: PowerShell detected that you might be using a screen reader and has disabled PSReadLine for compatibility purposes. If you want to re-enable it, run 'Import-Module PSReadLine'.

PS C:\Users\chira\Desktop\socket-local\Python> python .\server.py
Server listening on localhost:12345
Connected to client at ('127.0.0.1', 18914)
Client says: hello
Enter reply to client: hi
Client says: my name is chiranjeeb
Enter reply to client: my name is sever
```

PYTHON DOCKER-



The image shows two side-by-side windows. The left window is a Docker Desktop terminal showing the successful build and creation of two containers: 'python_socket_server' and 'python_socket_client'. It also shows the command to attach to the server container. The right window shows two PowerShell windows. The top window is attached to the 'python_socket_client' container, showing the client sending 'hello' and receiving 'hi'. The bottom window is attached to the 'python_socket_server' container, showing the server receiving 'hi' and responding with 'hello'.

```
Command Prompt - docker-c
=> [client] resolving provenance for metadata file
0.0s
[+] Running 5/5
  ✓ client      0.0s   Built
  ✓ server      0.0s   Built
  ✓ Network python_default 0.0s   Created
  ✓ Container python_socket_server 0.0s   Created
  ✓ Container python_socket_client 0.1s   Created
Attaching to python_socket_client, python_socket_server
python_socket_server | [SERVER] Listening on 0.0.0.0:12345...
python_socket_client | [CLIENT] Connected to server at server:12345
python_socket_server | [SERVER] Connected to client at ('172.18.0.3', 49886)
python_socket_client | [YOU] Enter message: hi
python_socket_server | [CLIENT] hi
python_socket_server | [YOU] Enter response: hello
python_socket_client | [SERVER] hello

View in Docker Desktop View Config Enable Watch

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

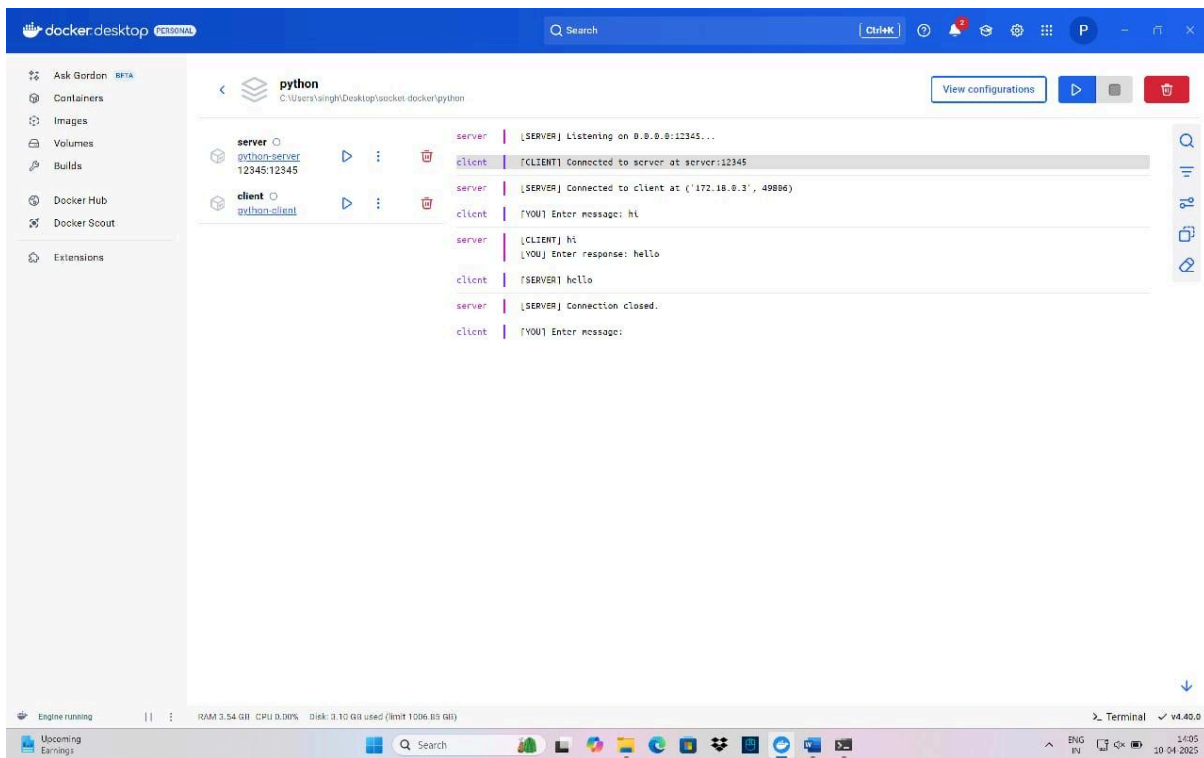
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\singh> docker attach python_socket_client
hi
[SERVER] hello
[YOU] Enter message:

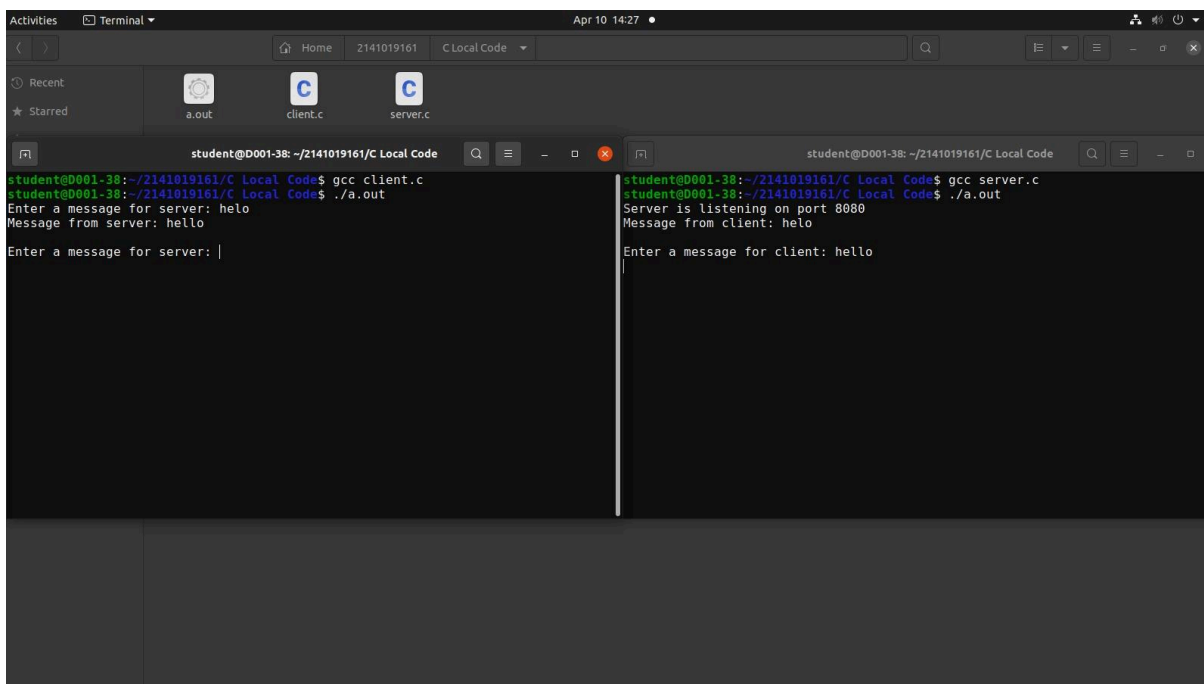
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\singh> docker attach python_socket_server
[CLIENT] hi
[YOU] Enter response: hello
```



C LOCAL-



C DOCKER-

The screenshot displays the Docker Desktop application window. The top section shows the 'C' container group with three containers: 'server', 'server', and 'client'. The 'server' containers are in a 'waiting' state, while the 'client' container is 'connected'. The bottom section shows the 'client' container's logs, which include the following text:

```
exporting layers 0.1s
exporting manifest sha256:eeb7d66f0f951988e29bcla7470222e93438bd191a2aaf2c9452c2
88cfb7d89b 0.0s
exporting config sha256:bab5589fd4a45acf9bc390d844c5688876f377f8eb2a2e5alc49bd61
f72fb3dd 0.0s
exporting attestation manifest sha256:b8ddale7b169915a7c6f4884732336f52fe0188d6f
b97548c52e6c7ddac68e99 0.0s
exporting manifest list sha256:6d3136f153a929c903ab69d0b5b11fc27e140b1d896f38a9e
felf08d73b81f8c 0.0s
naming to docker.io/library/c-client:latest 0.0s
unpacking to docker.io/library/c-client:latest 0.0s
[client] resolving provenance for metadata file 0.0s
Building 2/2
client Built 0.0s
```

The Docker Desktop interface also shows the 'server' container's logs, which include the following text:

```
Server waiting for connection...
Server waiting for connection...
Client connected:
client: Connected to server!
server: Client: hi
client: Client: hi
server: hello
```

The screenshot displays the Docker Desktop application window. The top section shows the 'C' container group with three containers: 'server', 'server', and 'client'. The 'server' containers are in a 'waiting' state, while the 'client' container is 'connected'. The bottom section shows the 'client' container's logs, which include the following text:

```
exporting layers 0.1s
exporting manifest sha256:eeb7d66f0f951988e29bcla7470222e93438bd191a2aaf2c9452c2
88cfb7d89b 0.0s
exporting config sha256:bab5589fd4a45acf9bc390d844c5688876f377f8eb2a2e5alc49bd61
f72fb3dd 0.0s
exporting attestation manifest sha256:b8ddale7b169915a7c6f4884732336f52fe0188d6f
b97548c52e6c7ddac68e99 0.0s
exporting manifest list sha256:6d3136f153a929c903ab69d0b5b11fc27e140b1d896f38a9e
felf08d73b81f8c 0.0s
naming to docker.io/library/c-client:latest 0.0s
unpacking to docker.io/library/c-client:latest 0.0s
[client] resolving provenance for metadata file 0.0s
Building 2/2
client Built 0.0s
```

The Docker Desktop interface also shows the 'server' container's logs, which include the following text:

```
Server waiting for connection...
Server waiting for connection...
Client connected:
client: Connected to server!
server: Client: hi
client: Client: hi
server: hello
```