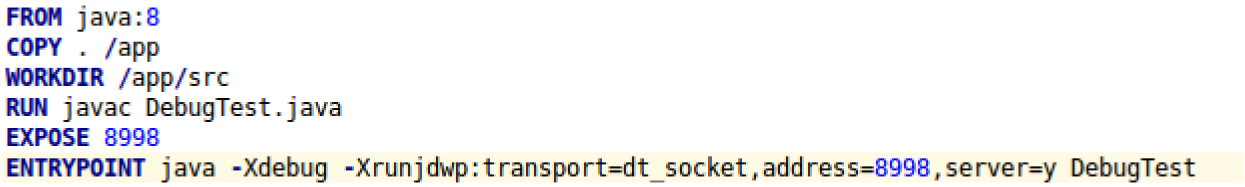
Docker to Host : Debugger

Step 1 : Docker file

EXPOSE <debug port>
Below is the Sample Dockerfile. Here 8998 is the debug port.It could be anything as your wish.



```
FROM java:8
COPY . /app
WORKDIR /app/src
RUN javac DebugTest.java
EXPOSE 8998
ENTRYPOINT java -Xdebug -Xrunjdwp:transport=dt_socket,address=8998,server=y DebugTest
```

Make sure to run the application/server in a debug mode

Step 2 : Build the Image

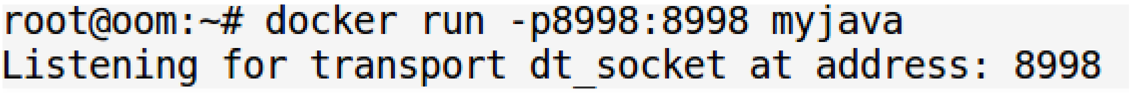
docker build -t <img name> .
Please verify the built image by docker image list

Step 3: Run the Image

docker run -p8998:8998 <img name>
So the request comes from the 8998 port to the host will be re directed to Docker
Please note : Host port could be anything , Docker port should be the one which you exposed

Step 4: Debug mode

Check whether the debug mode is enabled.



```
root@oom:~# docker run -p8998:8998 myjava
Listening for transport dt_socket at address: 8998
```

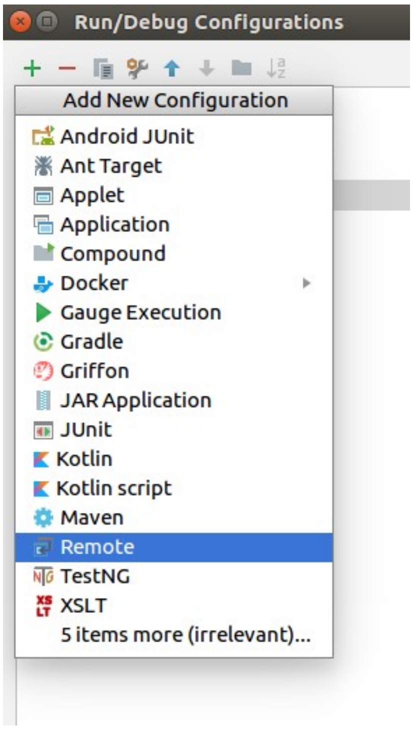
After running the container please confirm that the debug mode is enabled.
It should Listen for the mentioned address.

Step 5. Intelij IDE setup

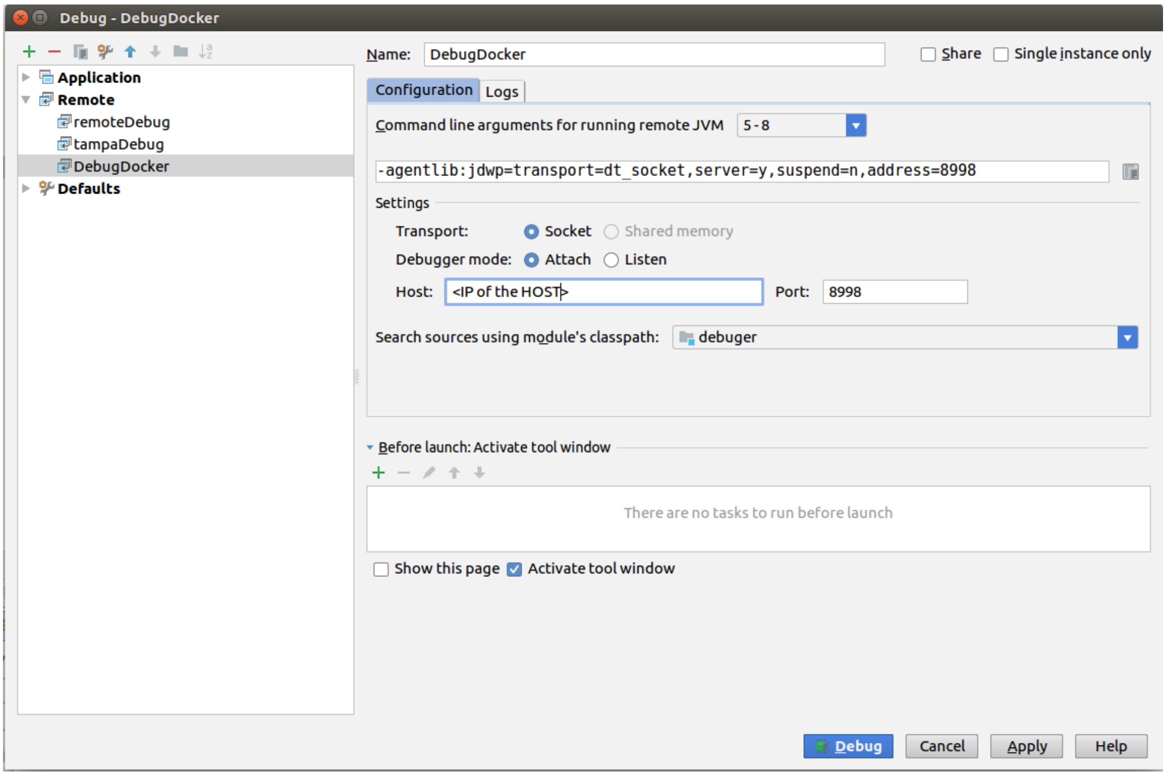
Please click

Run -> Debug -> Edit Configurations

Click the **green color plus** symbol -> Remote to add the remote debug configuration.



Please give a any name for this configuration.
IP address and port details should be mapped to the remote host.
Make sure that “Search sources using module’s class path” as your project root directory



Name: DebugDocker

□ Share □ Single instance only

Command line arguments for running remote JVM: 5 - 8

```
-agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=8998
```

Settings

Transport: ● Socket ○ Shared memory

Debugger mode: ● Attach ○ Listen

Host: <IP of the HOST> Port: 8998

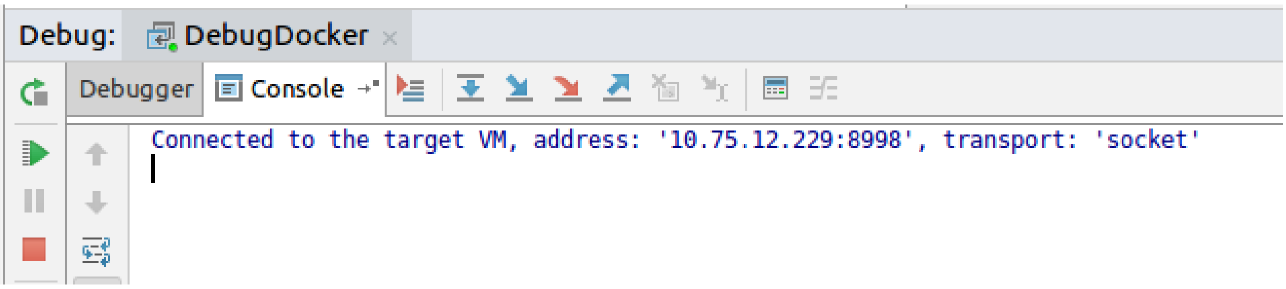
Search sources using module's classpath: debuger

Before launch: Activate tool window

There are no tasks to run before launch

□ Show this page ☑ Activate tool window

Please click apply and Debug.



```
Connected to the target VM, address: '10.75.12.229:8998', transport: 'socket'
```

It will connect with your remote host machine to debug.