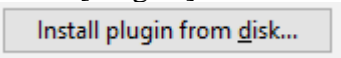




Introduction to building and running WebSocket Samples

Install Plugin

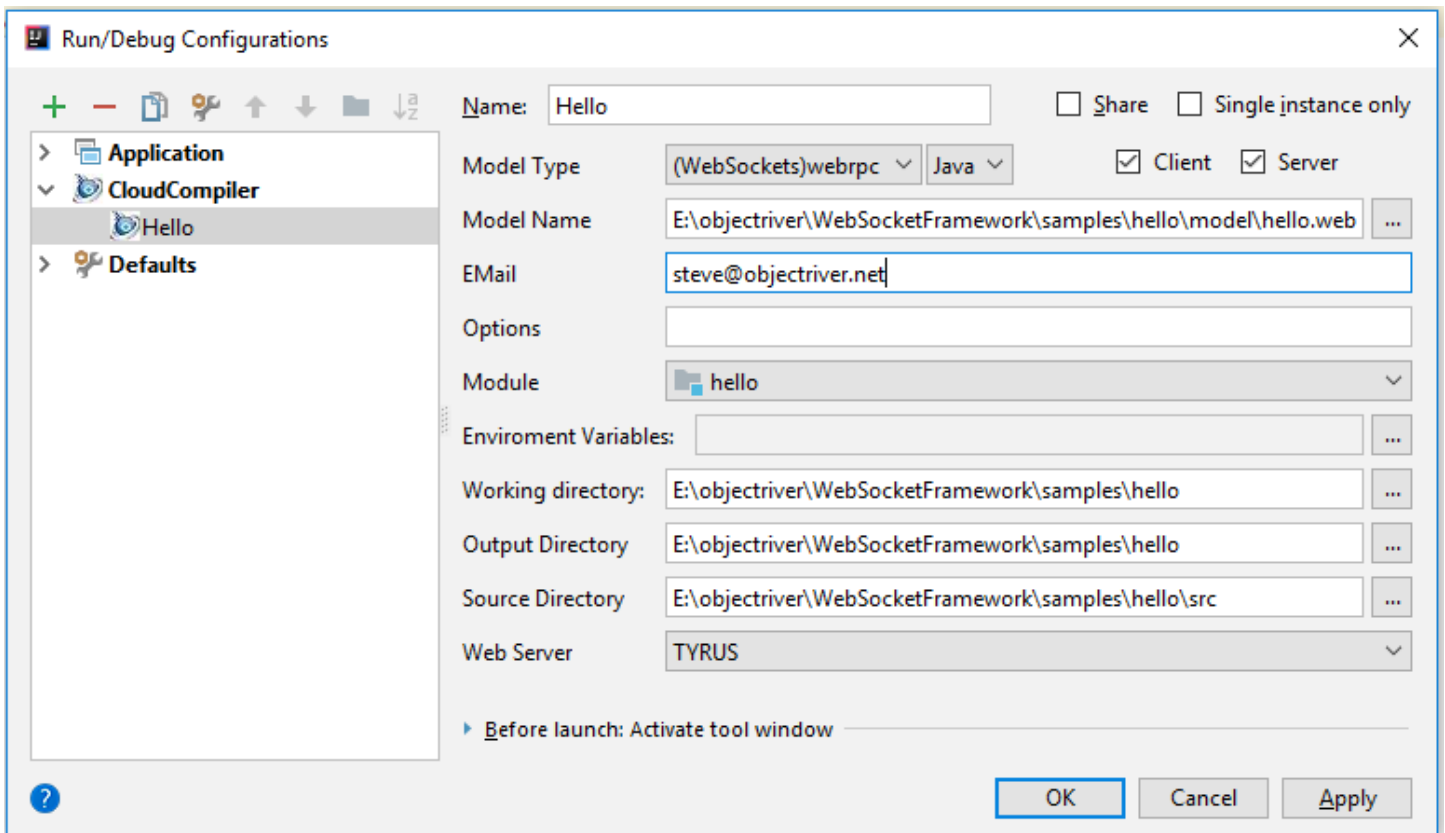
- IntelliJ Install CloudCompiler Plugin
 - File → Settings
 - Select [Plugins]
 - 
 - Directory
 <objectriver>/sdk/intellij/ObjectRiverCloudCompilerPlugin.zip
 - Restart IntelliJ

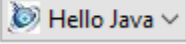
Set OBJECTRIVER_HOME env variable

- set OBJECTRIVER_HOME=C:/objectriver

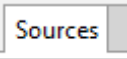
Build and Execute Sample

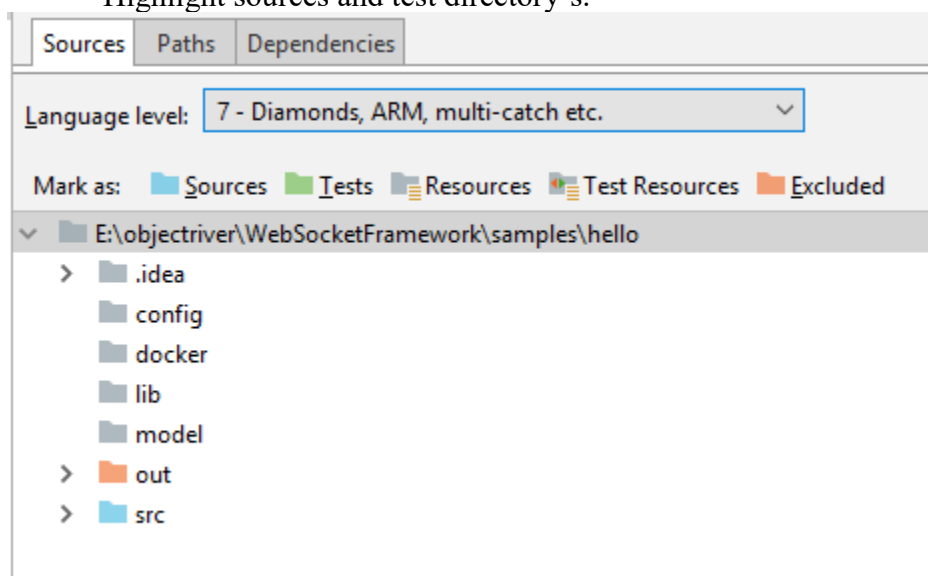
- New Project
 - File → New → Project...
 - Create a new IntelliJ project in a sample directory.
- Run → Edit Configurations...
 - [+] → CloudCompiler
 - Fill in dialog below



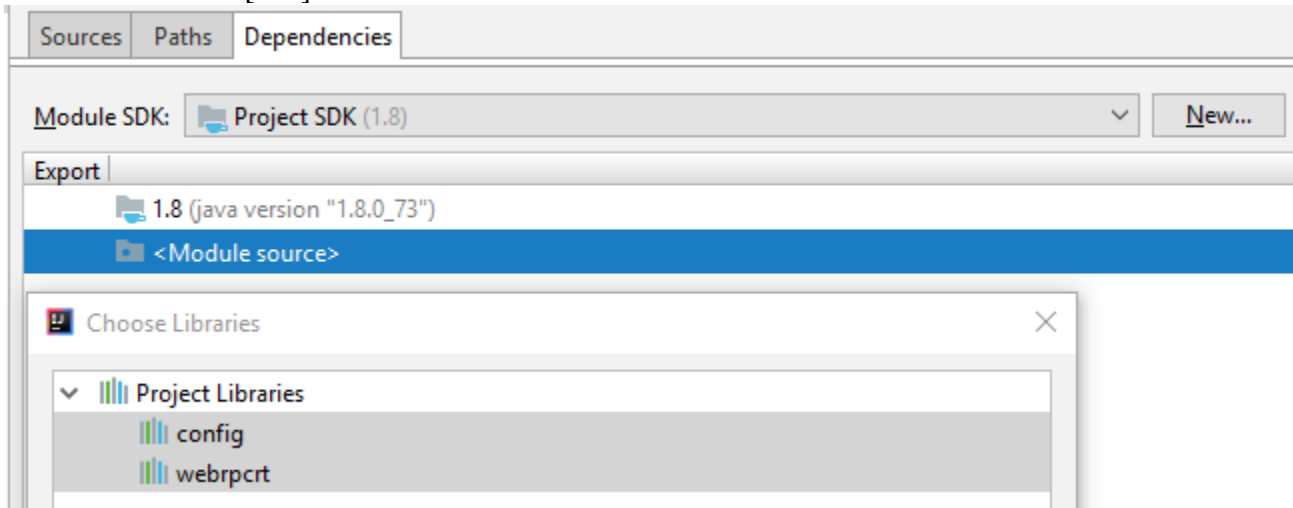
- Press (OK)
- Run CloudCompiler “Hello Java”
 - 
 - This will generate source code from the Hello.rest model.

- File → Project Structure 
 - [Modules]

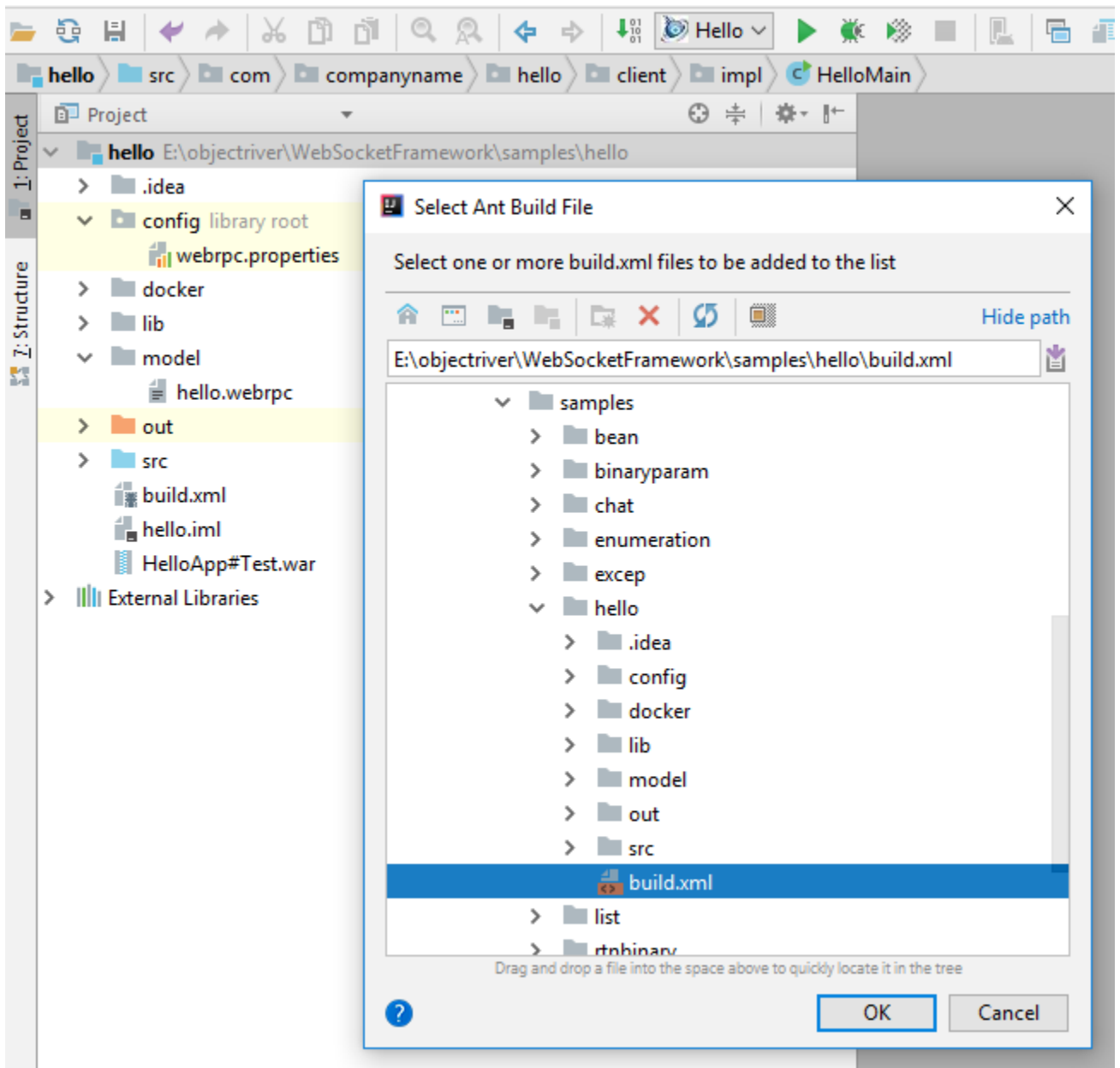
- Select Source Tab 
 - Highlight sources and test directory's.




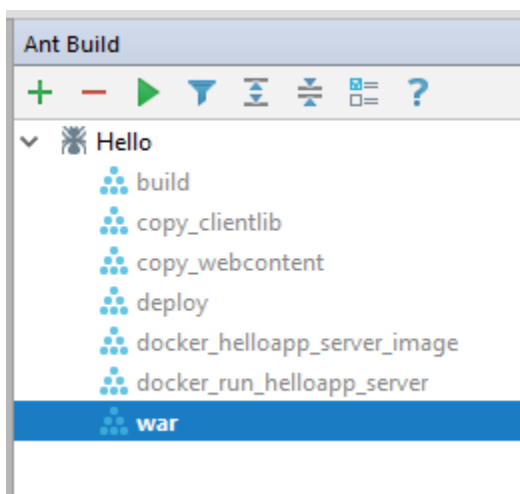
- Select Dependencies Tab
- [+] → Libraries
- Select libraries **config** and **webprcrt** and [Add Selected]
- [OK]



- View → Tools Window → Ant Build
 - [+]

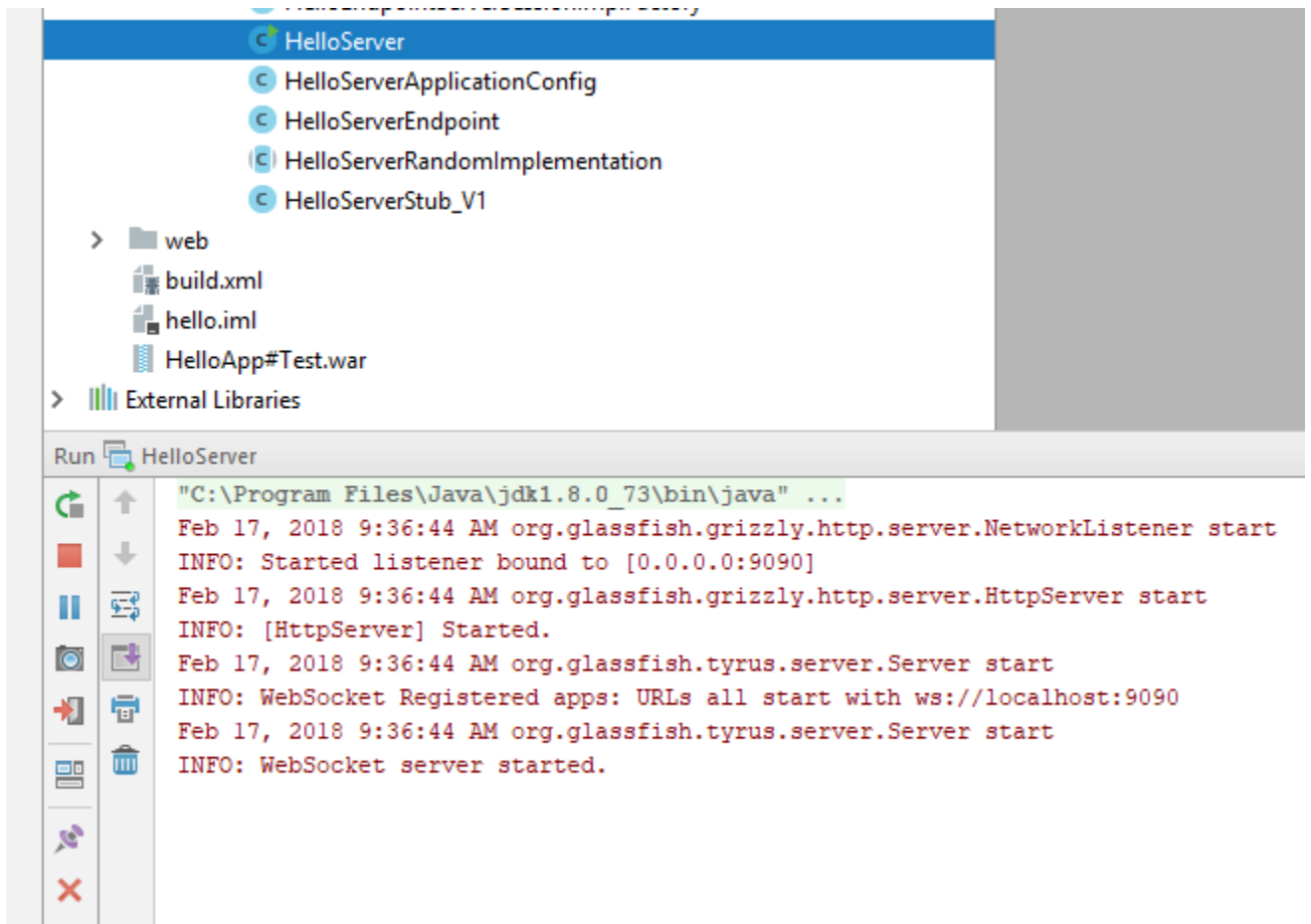


- Execute Ant war target.  war

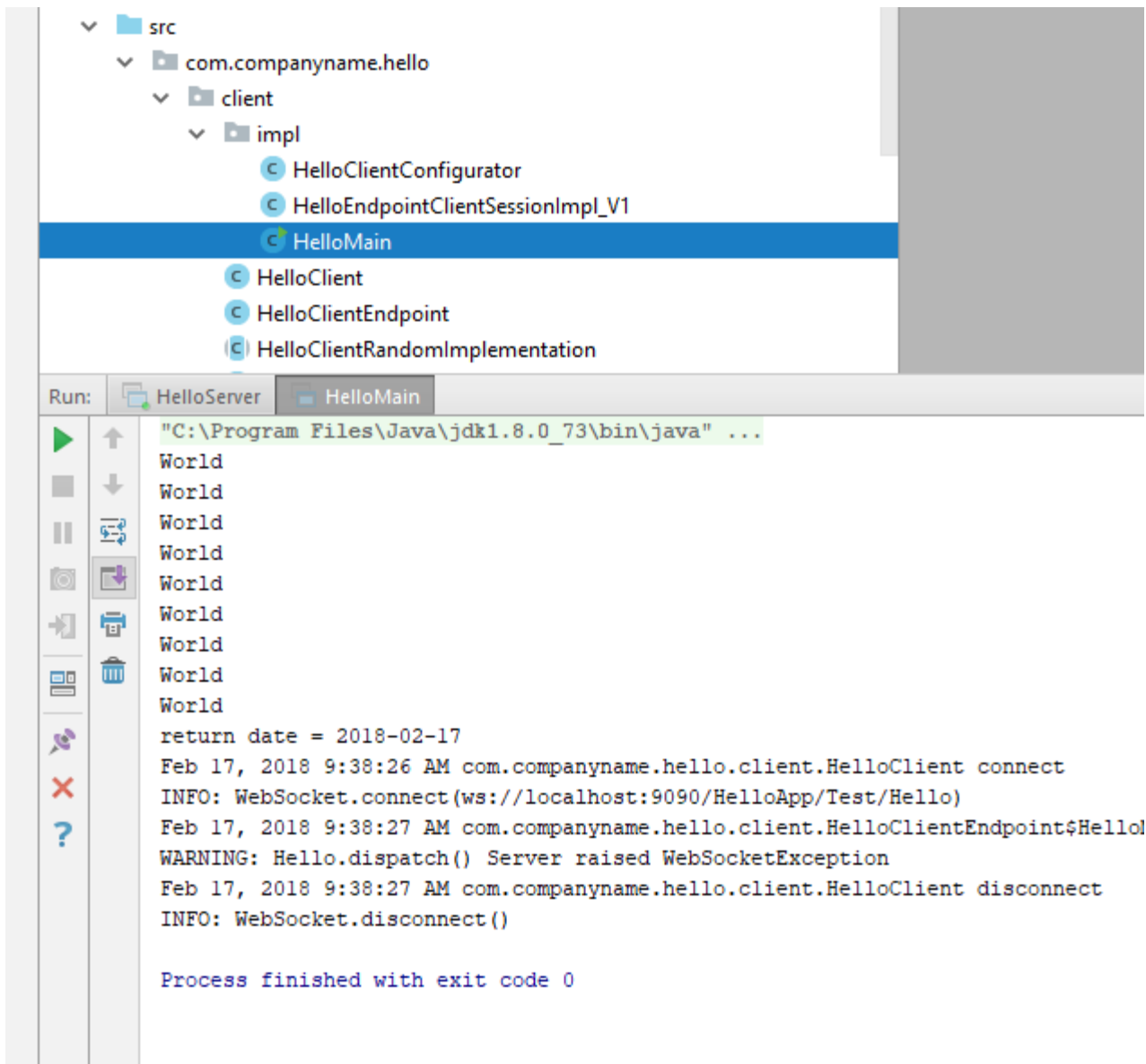


- This will create a HelloApp#Test.war file. WebSockets has compound war file name.
- Execute HelloServer

- Select the HelloServer, <RightClick>→Run
 - Server run on port 9090. See config directory.
 - By default the server will respond with random data. Ie. Server Mocking.
 - This server can be directly debugged.
 - This Server uses Oracle Tyrus/Grizzly for deploying to lightweight Internet Of Things servers. Docker/Alpine and Tomcat server uses JAXRS2 bundled in Tomcat





- Execute client test(s)
 - Select the HelloEndpointClientTest, <RightClick>→Run
 - Client uses on port 9090. See config directory.
 - By default the client will send random data. Ie. Client Mocking.
 -



- Stop HelloServer 

Build and Execute Docker Alpine-Tomcat Server

- Execute Ant Target  `docker_hello_server_image`
 - This will build a docker hello server image
- Execute Ant Target  `docker_run_hello_server`
 - This will load the hello docker image into a running docker container.
- Re-run client test(s)
 - Select the HelloEndpointClientTest, <RightClick>→Run
- Open CMD shell
- Execute **docker ps**

```
C:\Users\slemmo>docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS
5430fcb0bd30       hello_server:latest "/opt/tomcat/bin/cat..." 6 seconds ago       Up 5 seconds       8080/tcp, 0.0.0.0:9090->80/tcp
```

- Enjoy!