

Title: Mapping ports in Compute Engine with Docker

Post Body:

I have a docker image running on Google Compute Engine. The image contains a Spring Boot application running on port 9000.

It is exposed on `http://<ip>:9000` and I can access it without any problems. I am trying to configure the exposed port to be 80 in order to configure DNS record to point just to the IP address.

My question is how to achieve that because the documentation left me confused. I am using the online GCP console (web interface) for the deployment and there is no field to specify `docker run -p 9000:80` command which would solve my issue (if there is, please correct me).

So do I need to remap the port in the Spring Boot settings? Or inside the Docker container? Or am I supposed to configure some forwarding rules in GCP? Thanks for the clarification!

My DOCKER file looks as follows:

```
FROM gcr.io/distroless/java VOLUME /tmp ADD build/libs/*.jar app.jar ENV JAVA_OPTS='' ENTRYPOINT ['java','-jar','app.jar']
```

I am using the Container optimized OS from GCP.

Accepted Answer:

I solved my issue by mapping the Spring Boot port to port 80 in the production profile.

This allows me to map the DNS records to raw IP address without having to do any extra configuration.

Highest Rated Answer:

tldr : you can't

When you deploy a container in compute engine in this way, docker network is in `host network` mode, which means that ([doc](#)): a container shares the host's network stack and all interfaces from the host are available to the container.

So compute engine will directly expose container port, on its own interface. So you have to configure your container to expose correct port.