

Title: Jenkins Docker container can't upload artifact to Nexus Docker container

Post Body:

I have the following setup for my DevOps environment, orchestrated with Docker Compose:

(simplified) docker-compose.yml:

```
version: '3.7'
services:
  nexus:
    build: ./nexus/
    expose:
      - 8081
    networks:
      - devops-network
  jenkins:
```

(simplified) nginx.conf:

```
http {
    upstream docker-jenkins {
        server jenkins:8080;
    }
    upstream docker-nexus {
        server nexus:8081;
    }
}
```

Jenkins can communicate with Nexus through a Docker network. If I enter the Jenkins container, then `ping nexus` and `curl http://nexus:8081` give me positive responses (I get feedback from Nexus).

But when I run a Jenkins pipeline with a nested Docker agent for Maven:

```
pipeline {
    agent {
        docker { // successfully pulls image
            image 'maven:3.6.3-openjdk-11'
        }
    }
}
```

and maven settings.xml with the above id:

```
<servers>
  <server>
    <id>nexus-snapshots</id>
    <username>jenkins</username>
    <password>${JENKINS_NEXUS_PASSWORD}</password>
  </server>
</servers>
```

and this in my pom.xml:

```
<distributionManagement>
  <snapshotRepository>
    <id>nexus-snapshots</id>
    <url>http://nexus:8081/repository/maven-snapshots/</url>
  </snapshotRepository>
</distributionManagement>
```

then it says:

```
[ERROR] Failed to execute goal org.apache.maven.plugins:maven-deploy-plugin:2.8.2:deploy (default-deploy) on project parent: Failed to retrieve remote metadata ... from/to nexus-snapshots (http://nexus:8081/repository/maven-snapshots/): Transfer failed for http://nexus:8081/repository/maven-snapshots/.../1.0.0-SNAPSHOT/maven-metadata.xml: Unknown host nexus: No address associated with hostname
```

When I change the repository url in the pom.xml to the network IP of the Nexus container:

```
<distributionManagement>
  <snapshotRepository>
    <id>nexus-snapshots</id>
    <url>http://172.26.0.2:8081/repository/maven-snapshots/</url>
  </snapshotRepository>
</distributionManagement>
```

it then times out:

```
[ERROR] Failed to execute goal org.apache.maven.plugins:maven-deploy-plugin:2.8.2:deploy (default-deploy) on project parent: ... Connect to 172.26.0.2:8081 [/172.26.0.2] failed: Connection timed out
```

What is going on? I mapped the host Docker daemon to the Jenkins container with `setfacl -m user:jenkins:rw /var/run/docker.sock`, so it should be able to reach the Nexus container? When I pull the Docker image in the pipeline, the image gets added to my host system and doesn't need to be downloaded again the next time so this seems to be in working order. Is it because the 'Docker inside Docker' doesn't have access to devop-network?

Ideally I would be able to set `http://nexus.homenetwork.dns/repository/maven-snapshots/` as url in my pom instead, but this doesn't seem possible. Does anyone have experience with this kind of setup?

## Update

To validate my suspicions, I used `agent any` instead of a Docker agent in the Jenkins pipeline and I installed maven in the Jenkins container. With that setup, running the Jenkins pipeline correctly uploads the snapshots with url `http://nexus:8081/repository/maven-snapshots/`.

This is not a bad solution, but I'm left wondering how to make it work with the Docker agent out of pure curiosity.

Accepted Answer: None

Highest Rated Answer:

Hey I had the very same problem and I was looking for a solution for quite some time now. I also didn't get it at first. However, I figured it out at the end to be a quite basic docker thing. The docker agent basically runs a new docker container. If you want it to access another already existing one with just 'nexus' you have to add it to the same bridge network. In your case you need to add

```
pipeline {
    agent {
        docker { ... args '--net="devops-network"' }
    }
}
```

This at least solved it for me.