

# How To - Add Jenkins Master to Docker Group

## 1. Introduction -

While running Jenkins Master in Docker or other docker Jenkins Slave need a connectivity for docker.sock (/var/run/docker.sock).

Firstly, need to find GID of Docker group-

```
$ getent group docker
```

```
docker:x:999:xxxxxx
```

Using docker GID it's now possible to add jenkins user to it:

```
$ groupadd docker -g 999
```

```
$ usermod -a -G docker jenkins
```

## 2. Many ways to enable the Cloud >> Docker >> Test Connection -

### 2.1 Simple Way -

Assign the permission for docker socket (/var/run/docker.sock).

```
sudo chmod -R 777 /var/run/docker.sock
```

### 2.2 Various Ways to docker group -

**Build a another docker image on base image-**

```
#!/bin/bash -x
x=$(cat /etc/group | awk -F ":" '{ print $1,$3 }' | grep docker | awk
'{print $2}')
echo "
FROM jenkins/jenkins:2.176.2
USER root
RUN groupadd -g $x docker
RUN usermod -g docker jenkins
USER jenkins
" > /tmp/jenkinsDockerfile
```

**Another way-**

```

# In general, you should always provide exact version (eg. 2.89.3)
# rather than some more general tag (latest/lts)
FROM jenkins/jenkins:lts

ARG HOST_DOCKER_GROUP_ID

USER root

# Create 'docker' group with provided group ID
# and add 'jenkins' user to it
RUN groupadd docker -g ${HOST_DOCKER_GROUP_ID} && \
    usermod -a -G docker jenkins

# Install 'docker-ce' and it's dependencies
# https://docs.docker.com/engine/installation/linux/docker-ce/debian/
RUN apt-get update && \
    apt-get install -y --no-install-recommends \
        apt-transport-https \
        ca-certificates \
        curl \
        software-properties-common && \
    curl -fsSL https://download.docker.com/linux/${. /etc/os-release;
echo "$ID"}/gpg | apt-key add - && \
    add-apt-repository \
        "deb [arch=amd64] https://download.docker.com/linux/${.
/etc/os-release; echo "$ID") $(lsb_release -cs) stable" && \
    apt-get update && \
    apt-get install -y --no-install-recommends \
        docker-ce && \
    apt-get clean

# Run Jenkins as dedicated non-root user
USER jenkins

=====
=====
=====

docker build \
    --build-arg HOST_DOCKER_GROUP_ID=\
        "`getent group docker | cut -d':' -f3`" \
    -t jenkins-with-docker .

```

**One more good way-**

```

FROM jenkins:1.651.1
MAINTAINER xxx
USER root
RUN apt-get update \
    && apt-get install -y apt-transport-https ca-certificates \
    && echo "deb https://apt.dockerproject.org/repo debian-jessie main" >
/etc/apt/sources.list.d/docker.list \
    && apt-key adv --keyserver hkp://p80.pool.sks-keyservers.net:80
--recv-keys 58118E89F3A912897C070ADBF76221572C52609D \
    && apt-get update -y \
    && apt-get install -y docker-engine \
    && apt-get install sudo \
    && apt-get clean \
    && rm -rf /var/lib/apt/lists/* /tmp/* /var/tmp/*
ADD jenkins.sh /usr/local/bin/jenkins.sh
RUN chmod a+x /usr/local/bin/jenkins.sh
ENTRYPOINT ["/usr/local/bin/jenkins.sh"]

=====
=====
=====
jenkins.sh:

#!/bin/bash -x

# this only works if the docker group does not already exist

DOCKER_SOCKET=/var/run/docker.sock
DOCKER_GROUP=docker

if [ -S ${DOCKER_SOCKET} ]; then
    DOCKER_GID=$(stat -c '%g' ${DOCKER_SOCKET})
    groupadd -for -g ${DOCKER_GID} ${DOCKER_GROUP}
    usermod -aG ${DOCKER_GROUP} jenkins
fi

```

### 3. References -

- <https://blog.mkowalski.net/2018/01/22/build-docker-images-with-jenkins-running-in-docker/>
- <https://github.com/jenkinsci/docker/issues/263>