**DevOps Lab 2.2: Docker Images**

Working with Docker images

**Step 1**

Go to the Cloud Platform Console at [https://console.cloud.google.com](https://console.cloud.google.com/). Go to *Compute Engine* and *VM Instances*. Start the VM if it isn’t running, and connect using *SSH*.

**Step 2**

Create an SSH client key pair (public and private). Check to see if you have SSH keys generated. ls ~/.ssh

If you see only the file “authorized\_keys,” then you need to perform the following step to generate an SSH client key.  
--> Accept the default file  
--> Do not enter a passphrase  
ssh-keygen -t rsa

**Step 3**

Prepare a Dockerfile to create a new Docker image.

Create a subdirectory to hold the context for the Docker build operation.  
mkdir build

Enter that directory.  
cd build

Copy the public part of the SSH key pair into the build directory.  
cp ~/.ssh/id\_rsa.pub authorized\_keys

Create a Dockerfile with the following contents. (Use *vi Dockerfile* or *nano Dockerfile*)

FROM alpine:latest  
RUN apk update  
RUN apk add openssh  
RUN adduser -g "Student User" -D student && mkdir /home/student/.ssh  
RUN echo "student:student" | chpasswd  
ADD authorized\_keys /home/student/.ssh  
RUN chown -R student.student /home/student  
RUN chmod 700 /home/student/.ssh && chmod 600 /home/student/.ssh/authorized\_keys  
RUN ssh-keygen -t rsa -f /etc/ssh/ssh\_host\_rsa\_key -q -N ""  
EXPOSE 22  
CMD ["/usr/sbin/sshd", "-D"]

**Step 4**

Create a new Docker image and test.

Create the docker image using the Dockerfile commands.  
docker build -t ssh:alpine

See that the image got created.  
docker images

Run the new container which contains your public SSH key, and run the SSH daemon.  
docker run -d -p 2022:22 --name ssh ssh:alpine

Make sure there are no errors.  
docker logs ssh

Connect to the Docker container using your Linux ssh client on your computer.  
ssh -p 2022 student@localhost

You are now inside the Alpine Linux container.  
Please explore!  
Exit by typing exit or control-D.

**Step 5**

Clean up docker stop ssh  
docker rm ssh  
docker rmi ssh:alpine  
docker rmi alpine:latest