使用Dockerfile文件构建jdk的docker镜像

使用的jdk版本为1.8：jdk-8u321-linux-x64.tar.gz

新建文件夹，将jdk上传，然后vi Dockerfile

表格

描述已自动生成

屏幕的截图

描述已自动生成

编写生成镜像的文件

FROM centos

MAINTAINER Wujiale

WORKDIR /java\_docker/jdk

ADD jdk-8u321-linux-x64.tar.gz /java\_docker/jdk

ENV JAVA\_HOME /java\_docker/jdk/jdk1.8.0\_321

ENV CLASSPATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar

ENV PATH=$JAVA\_HOME/bin:$PATH

#CMD ["java","-version"]

[root@hecs-18197 jdk]# **vi Dockerfile**

[root@hecs-18197 jdk]# **docker build -t jdk8:v2.0 .**

Sending build context to Docker daemon 825.2 MB

Step 1/7 : FROM centos

---> 5d0da3dc9764

Step 2/7 : MAINTAINER Wujiale

---> Using cache

---> 6374bd49335c

Step 3/7 : WORKDIR /java\_docker/jdk

---> Using cache

---> 4b8bf0fa7c2b

Step 4/7 : ADD jdk-8u321-linux-x64.tar.gz /java\_docker/jdk

---> 7543744eeb2d

Removing intermediate container afb0098191ae

Step 5/7 : ENV JAVA\_HOME /java\_docker/jdk/jdk1.8.0\_321

---> Running in 590776263e9c

---> 189363d07db2

Removing intermediate container 590776263e9c

Step 6/7 : ENV CLASSPATH .:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar

---> Running in 15738b70ab99

---> c2735f888734

Removing intermediate container 15738b70ab99

Step 7/7 : ENV PATH $JAVA\_HOME/bin:$PATH

---> Running in 6f0fa4ea40f4

---> f2f726e5c28b

Removing intermediate container 6f0fa4ea40f4

Successfully built f2f726e5c28b

[root@hecs-18197 jdk]# docker image ls

REPOSITORY TAG IMAGE ID CREATED SIZE

**jdk8 v2.0 f2f726e5c28b 13 seconds ago 597 MB**

jdk8 v1.0 5433126c21b1 9 minutes ago 586 MB

jdk1.8 latest 92c064aa2091 39 minutes ago 586 MB

docker.io/tomcat latest 7a287e4562ea 6 weeks ago 680 MB

docker.io/alpine latest c059bfaa849c 4 months ago 5.59 MB

docker.io/centos latest 5d0da3dc9764 7 months ago 231 MB

docker.io/portainer/portainer latest 580c0e4e98b0 13 months ago 79.1 MB

[root@hecs-18197 jdk]# **docker run -it jdk8:v2.0 /bin/bash**

[root@ecfd15b46cdf jdk]# **java -version**

java version "1.8.0\_321"

Java(TM) SE Runtime Environment (build 1.8.0\_321-b07)

Java HotSpot(TM) 64-Bit Server VM (build 25.321-b07, mixed mode)

[root@ecfd15b46cdf jdk]# **javac**

Usage: javac <options> <source files>

where possible options include:

-g Generate all debugging info

-g:none Generate no debugging info

-g:{lines,vars,source} Generate only some debugging info

-nowarn Generate no warnings

-verbose Output messages about what the compiler is doing

-deprecation Output source locations where deprecated APIs are used

-classpath <path> Specify where to find user class files and annotation processors

-cp <path> Specify where to find user class files and annotation processors

-sourcepath <path> Specify where to find input source files

-bootclasspath <path> Override location of bootstrap class files

-extdirs <dirs> Override location of installed extensions

-endorseddirs <dirs> Override location of endorsed standards path

-proc:{none,only} Control whether annotation processing and/or compilation is done.

-processor <class1>[,<class2>,<class3>...] Names of the annotation processors to run; bypasses default discovery process

-processorpath <path> Specify where to find annotation processors

-parameters Generate metadata for reflection on method parameters

-d <directory> Specify where to place generated class files

-s <directory> Specify where to place generated source files

-h <directory> Specify where to place generated native header files

-implicit:{none,class} Specify whether or not to generate class files for implicitly referenced files

-encoding <encoding> Specify character encoding used by source files

-source <release> Provide source compatibility with specified release

-target <release> Generate class files for specific VM version

-profile <profile> Check that API used is available in the specified profile

-version Version information

-help Print a synopsis of standard options

-Akey[=value] Options to pass to annotation processors

-X Print a synopsis of nonstandard options

-J<flag> Pass <flag> directly to the runtime system

-Werror Terminate compilation if warnings occur

@<filename> Read options and filenames from file

至此，能够运行java的镜像和一个容器创建完毕

下面再使用jdk18创建镜像

文本

描述已自动生成

[root@hecs-18197 jdk18]# docker build -t jdk18:v2.0 .

Sending build context to Docker daemon 182.2 MB

Step 1/7 : FROM centos

---> 5d0da3dc9764

Step 2/7 : MAINTAINER Wujiale

---> Using cache

---> 6374bd49335c

Step 3/7 : WORKDIR /java\_docker/jdk

---> Using cache

---> 4b8bf0fa7c2b

Step 4/7 : ADD jdk-18\_linux-x64\_bin.tar.gz /java\_docker/jdk

---> 32ecbb879cfc

Removing intermediate container a256b2026224

Step 5/7 : ENV JAVA\_HOME /java\_docker/jdk/jdk-18

---> Running in c0b2a288570e

---> 124f8b4fa52b

Removing intermediate container c0b2a288570e

Step 6/7 : ENV CLASSPATH .:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar

---> Running in c6c813f3f63a

---> 9c501ca4c69e

Removing intermediate container c6c813f3f63a

Step 7/7 : ENV PATH $JAVA\_HOME/bin:$PATH

---> Running in 851c8ea16d7c

---> 32bc50c60ee7

Removing intermediate container 851c8ea16d7c

Successfully built 32bc50c60ee7

[root@hecs-18197 jdk18]# docker run -it jdk18:v2.0 /bin/bash

[root@f7d73b636805 jdk]# java -version

java version "18" 2022-03-22

Java(TM) SE Runtime Environment (build 18+36-2087)

Java HotSpot(TM) 64-Bit Server VM (build 18+36-2087, mixed mode, sharing)

[root@f7d73b636805 jdk]# javac

Usage: javac <options> <source files>

where possible options include:

@<filename> Read options and filenames from file

-Akey[=value] Options to pass to annotation processors

--add-modules <module>(,<module>)\*

Root modules to resolve in addition to the initial modules, or all modules

on the module path if <module> is ALL-MODULE-PATH.