Introduction to Docker

Outline

- Review of software building automation tools
- Why Docker?
- What is Docker?
- Introduction to Docker
- Simple Demos
- Useful Resources

Software Building Automation Tools

- Make
- Ant
- Maven
- Docker

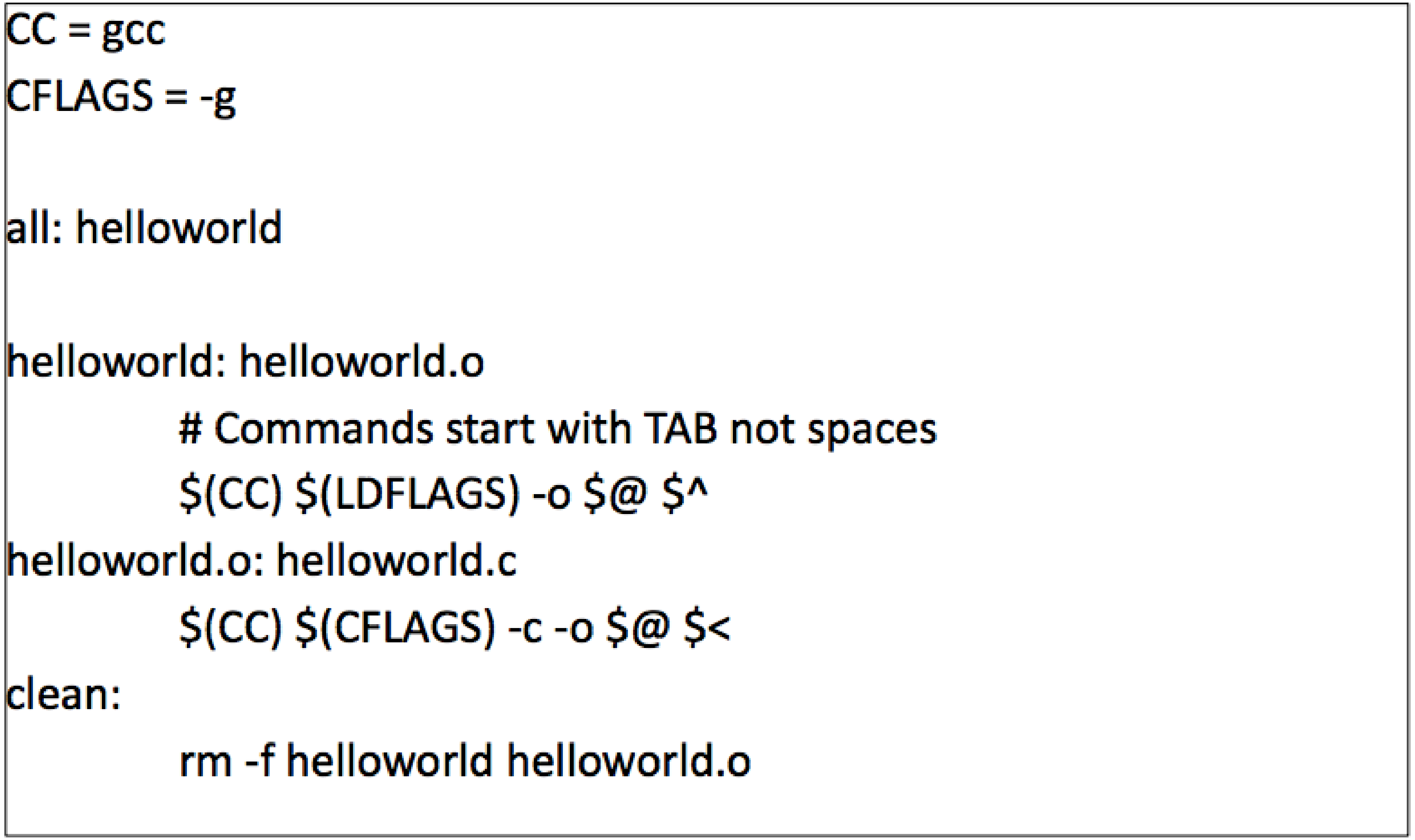
Make

- An utility that automatically builds executable programs and libraries from source code
- Take a ‘makefile’ as input
- Automatically support updating for each program
- Support dependences between programs (object code)
- ‘Make’ still remains widely used, especially in Unix

Many make implementations

- BSD make (pmake)
- GNU make
- A&T nmake
- Microsoft nmake

A simple makefile



```
CC = gcc
CFLAGS = -g

all: helloworld

helloworld: helloworld.o
	# Commands start with TAB not spaces
	$(CC) $(LDFLAGS) -o $@ $^
helloworld.o: helloworld.c
	$(CC) $(CFLAGS) -c -o $@ $<
clean:
	rm -f helloworld helloworld.o
```

How to use the makefile?

- make
    - Which is equivalent to:
        - make all
        - Make helloworld
- Or: nmake (on windows)

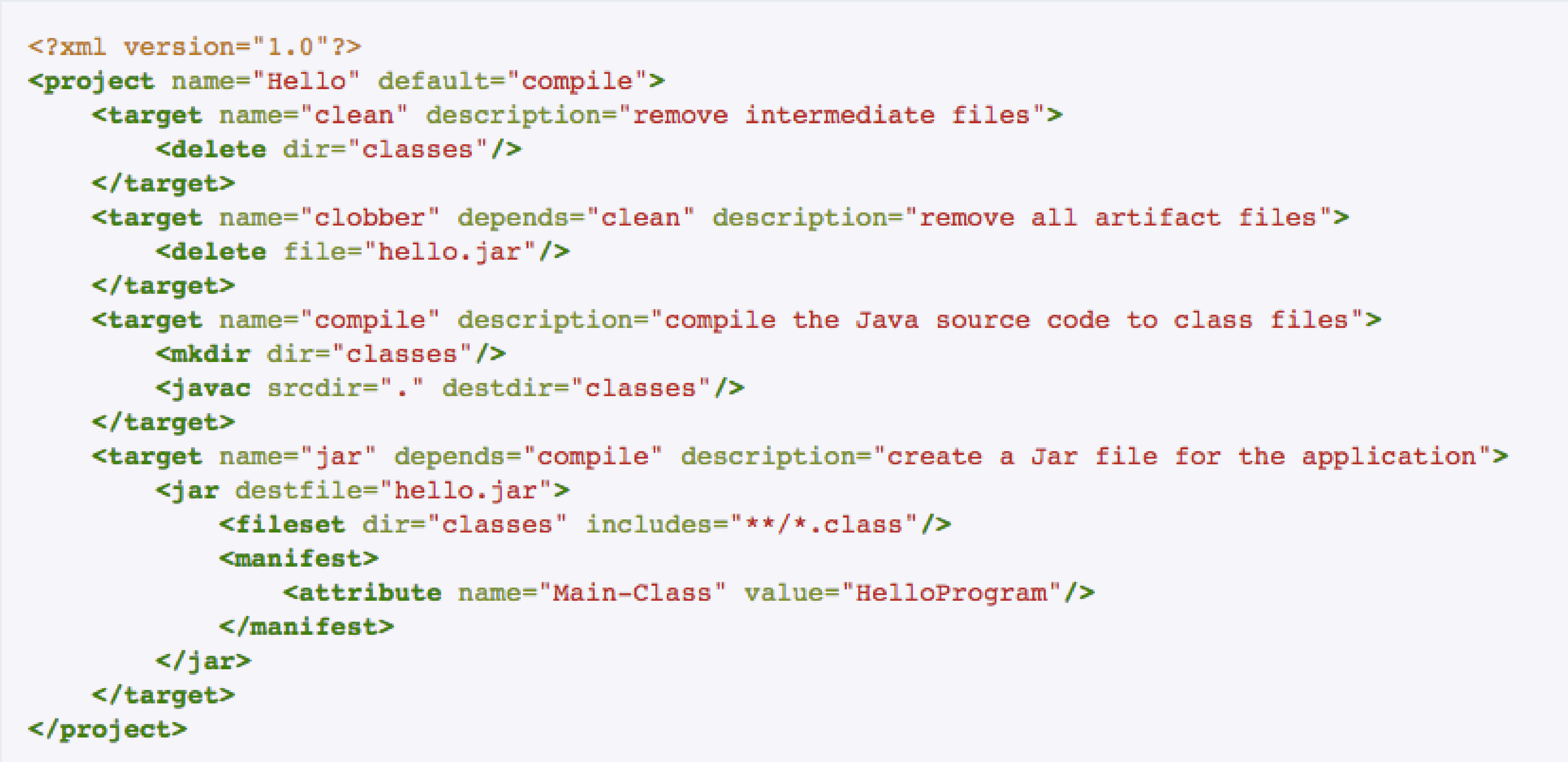
Tips for using make

- Usually for compiling C/C++ programs
- Carefully define $(INCLUDE) & $(LIB) for your: • *.h • *.lib
- Clearly define the dependency between these *.o
- Put ‘clean’ as the last task for a clean build

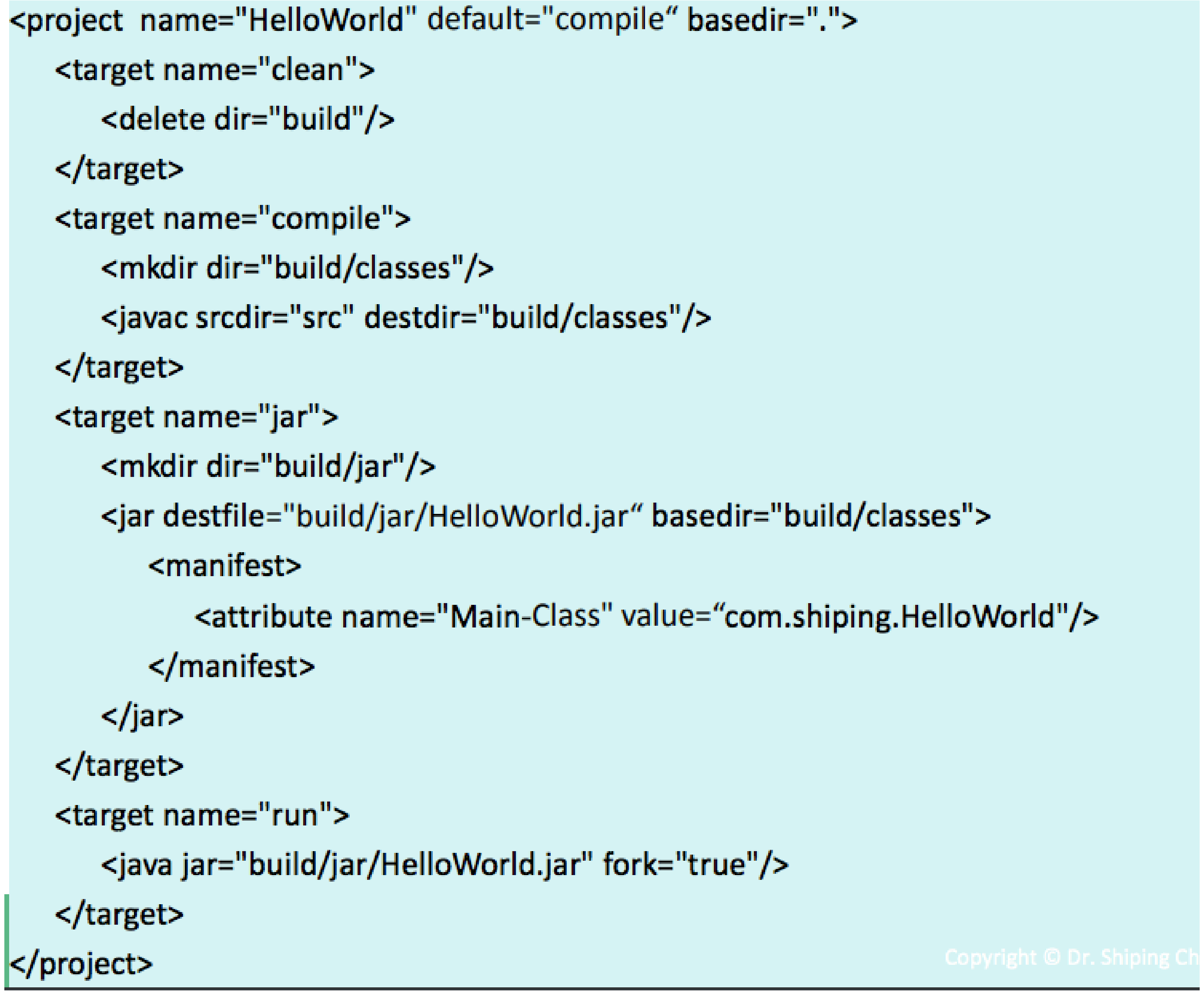
Ant – A Java build/package tool

- Another Neat Tool
- Apache Ant is a Java tool whose mission is to drive processes described in build files as targets and extension points dependent upon each other.
- The main usage of Ant is to compile, package/assemble and run java applications using built-in tasks
- Can also be used effectively to build non Java applications, e.g., C or C++ applications
- http://ant.apache.org/

Tasks	Description
<javac>	Compiles the specified source file(s)
<Jar>	Package a set of classes to a jar file
<java>	executes a Java class
delete	Deletes either a single file, all files and sub-directories in a specified directory
mkdir	Creates a directory. Non-existent parent directories are created, when necessar
echo	print a message to the current loggers and listeners
javadoc	Generates code documentation using the *javadoc* tool



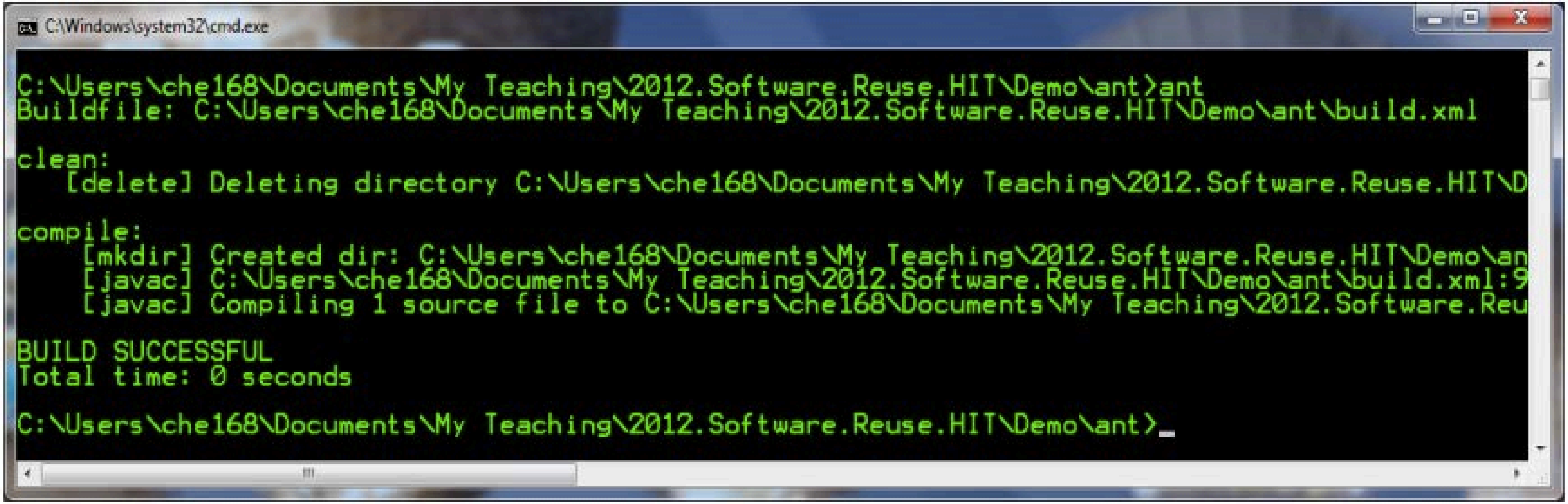
```xml
<?xml version="1.0"?>
<project name="Hello" default="compile">
    <target name="clean" description="remove intermediate files">
        <delete dir="classes"/>
    </target>
    <target name="clobber" depends="clean" description="remove all artifact files">
        <delete file="hello.jar"/>
    </target>
    <target name="compile" description="compile the Java source code to class files">
        <mkdir dir="classes"/>
        <javac srcdir="." destdir="classes"/>
    </target>
    <target name="jar" depends="compile" description="create a Jar file for the application">
        <jar destfile="hello.jar">
            <fileset dir="classes" includes="**/*.class"/>
            <manifest>
                <attribute name="Main-Class" value="HelloProgram"/>
            </manifest>
        </jar>
    </target>
</project>
```



```
<project  name="HelloWorld" default="compile“ basedir=".">
    <target name="clean">
        <delete dir="build"/>
    </target>
    <target name="compile">
        <mkdir dir="build/classes"/>
        <javac srcdir="src" destdir="build/classes"/>
    </target>
    <target name="jar">
        <mkdir dir="build/jar"/>
        <jar destfile="build/jar/HelloWorld.jar“ basedir="build/classes">
            <manifest>
                <attribute name="Main-Class" value=“com.shiping.HelloWorld"/>
            </manifest>
        </jar>
    </target>
    <target name="run">
        <java jar="build/jar/HelloWorld.jar" fork="true"/>
    </target>
</project>
```

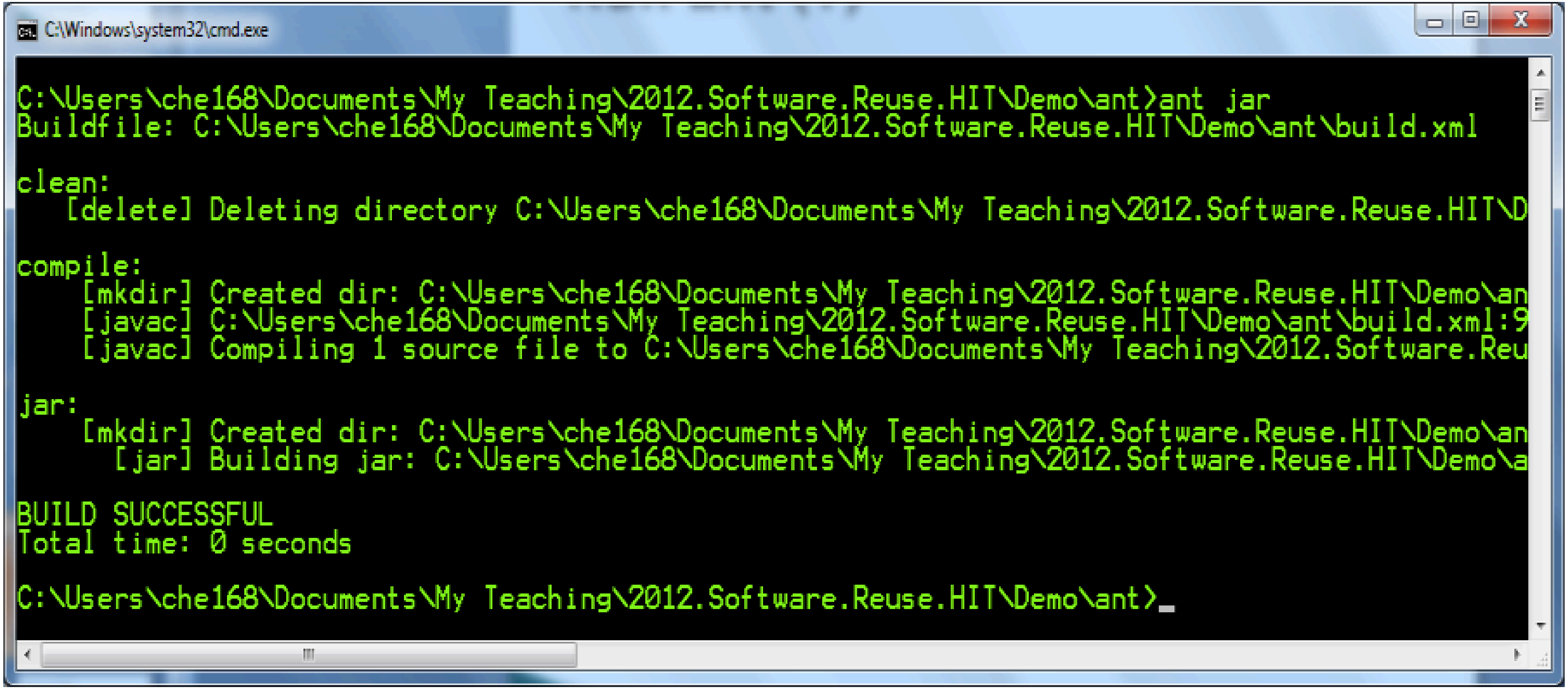
Run ant for compiling

- If you run ant with no argument, Ant will use the default build.xml and conduct the defaul task defined as the property of <project>



```
C:\Users\che168\Documents\My Teaching\2012.Software.Reuse.HIT\Demo\ant>ant
Buildfile: C:\Users\che168\Documents\My Teaching\2012.Software.Reuse.HIT\Demo\ant\build.xml

clean:
   [delete] Deleting directory C:\Users\che168\Documents\My Teaching\2012.Software.Reuse.HIT\D

compile:
    [mkdir] Created dir: C:\Users\che168\Documents\My Teaching\2012.Software.Reuse.HIT\Demo\an
    [javac] C:\Users\che168\Documents\My Teaching\2012.Software.Reuse.HIT\Demo\ant\build.xml:9
    [javac] Compiling 1 source file to C:\Users\che168\Documents\My Teaching\2012.Software.Reu

BUILD SUCCESSFUL
Total time: 0 seconds

C:\Users\che168\Documents\My Teaching\2012.Software.Reuse.HIT\Demo\ant>_
```

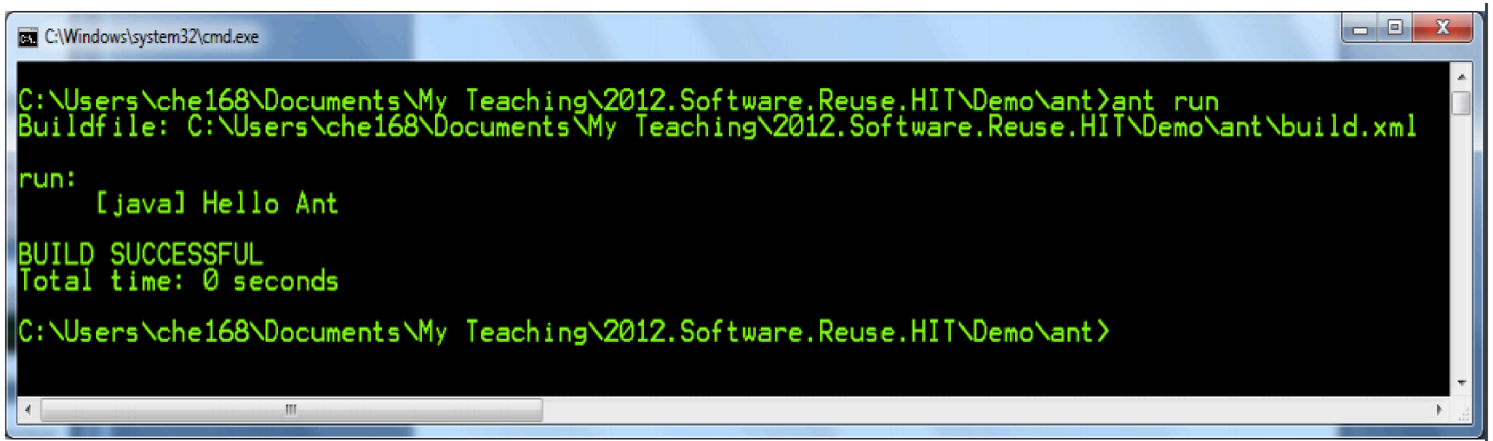
Run ant for packaging



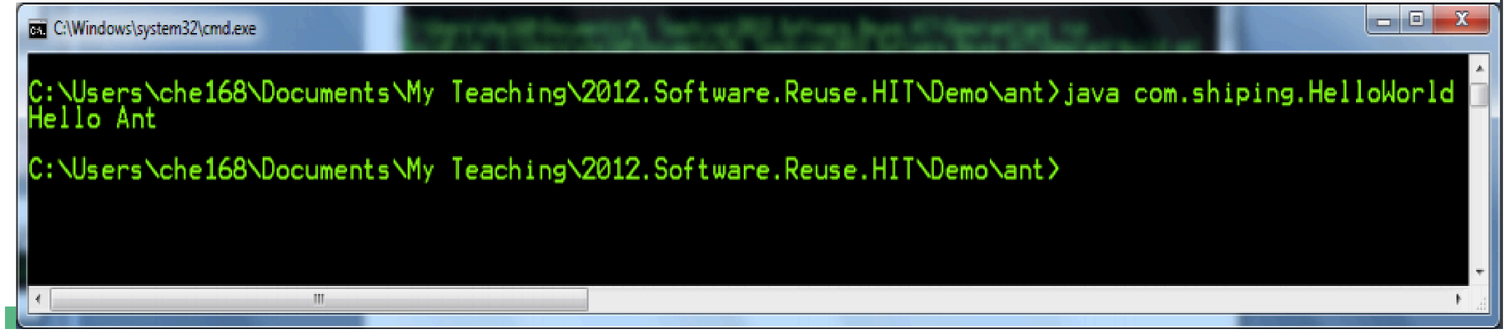
```
C:\Users\che168\Documents\My Teaching\2012.Software.Reuse.HIT\Demo\ant>ant jar
Buildfile: C:\Users\che168\Documents\My Teaching\2012.Software.Reuse.HIT\Demo\ant\build.xml

clean:
   [delete] Deleting directory C:\Users\che168\Documents\My Teaching\2012.Software.Reuse.HIT\D

compile:
    [mkdir] Created dir: C:\Users\che168\Documents\My Teaching\2012.Software.Reuse.HIT\Demo\an
    [javac] C:\Users\che168\Documents\My Teaching\2012.Software.Reuse.HIT\Demo\ant\build.xml:9
    [javac] Compiling 1 source file to C:\Users\che168\Documents\My Teaching\2012.Software.Reu

jar:
    [mkdir] Created dir: C:\Users\che168\Documents\My Teaching\2012.Software.Reuse.HIT\Demo\an
      [jar] Building jar: C:\Users\che168\Documents\My Teaching\2012.Software.Reuse.HIT\Demo\a

BUILD SUCCESSFUL
Total time: 0 seconds

C:\Users\che168\Documents\My Teaching\2012.Software.Reuse.HIT\Demo\ant>_
```

Run ant for execution

- You can use ant to run a java program as the same as using java directly:



```
C:\Users\che168\Documents\My Teaching\2012.Software.Reuse.HIT\Demo\ant>ant run
Buildfile: C:\Users\che168\Documents\My Teaching\2012.Software.Reuse.HIT\Demo\ant\build.xml

run:
     [java] Hello Ant

BUILD SUCCESSFUL
Total time: 0 seconds

C:\Users\che168\Documents\My Teaching\2012.Software.Reuse.HIT\Demo\ant>
```



```
C:\Users\che168\Documents\My Teaching\2012.Software.Reuse.HIT\Demo\ant>java com.shiping.HelloWorld
Hello Ant

C:\Users\che168\Documents\My Teaching\2012.Software.Reuse.HIT\Demo\ant>
```

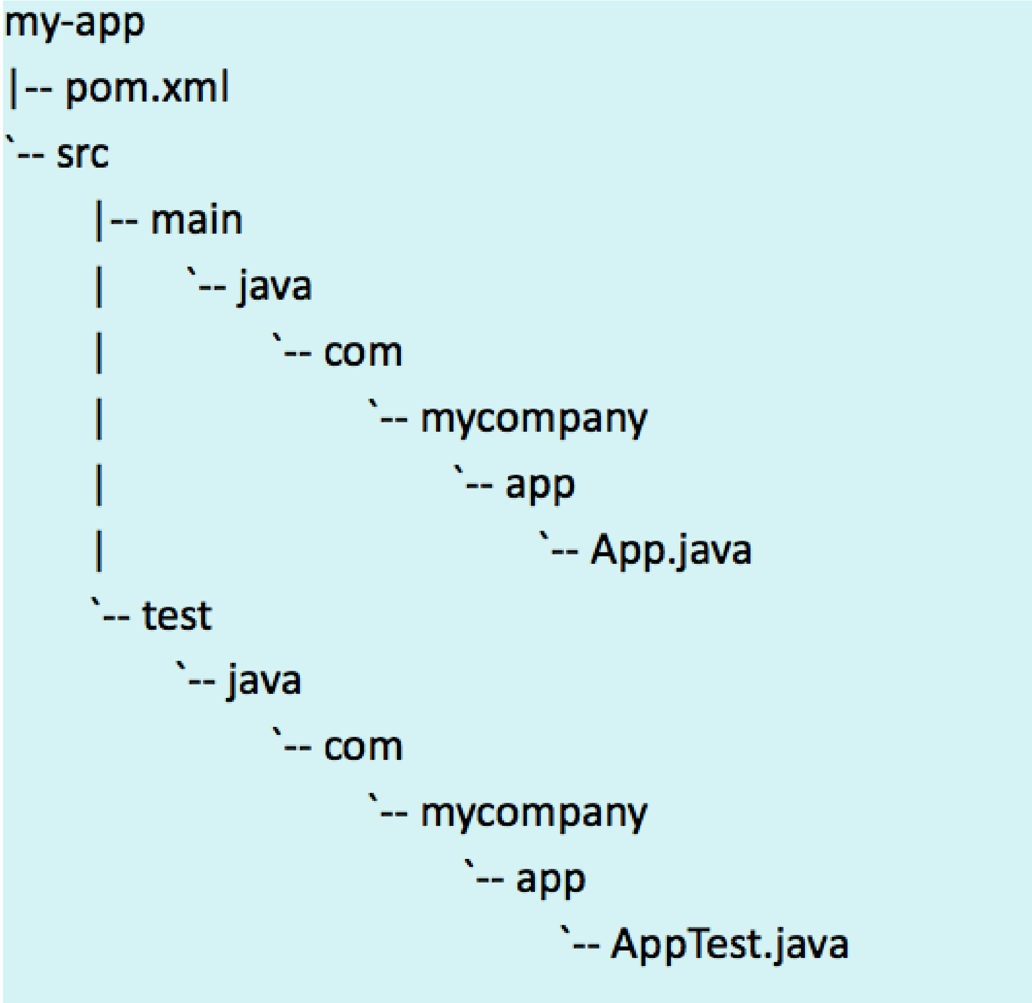
Maven: a java-based project management tool

- The name is from Yiddish (Old German Language) meaning one who understands.
- Based on the concept of a Project Object Model (POM)
- Can manage a java-based project's building, reporting and documentation.
- Suitable for managing large and complex java-based projects

What maven offers?

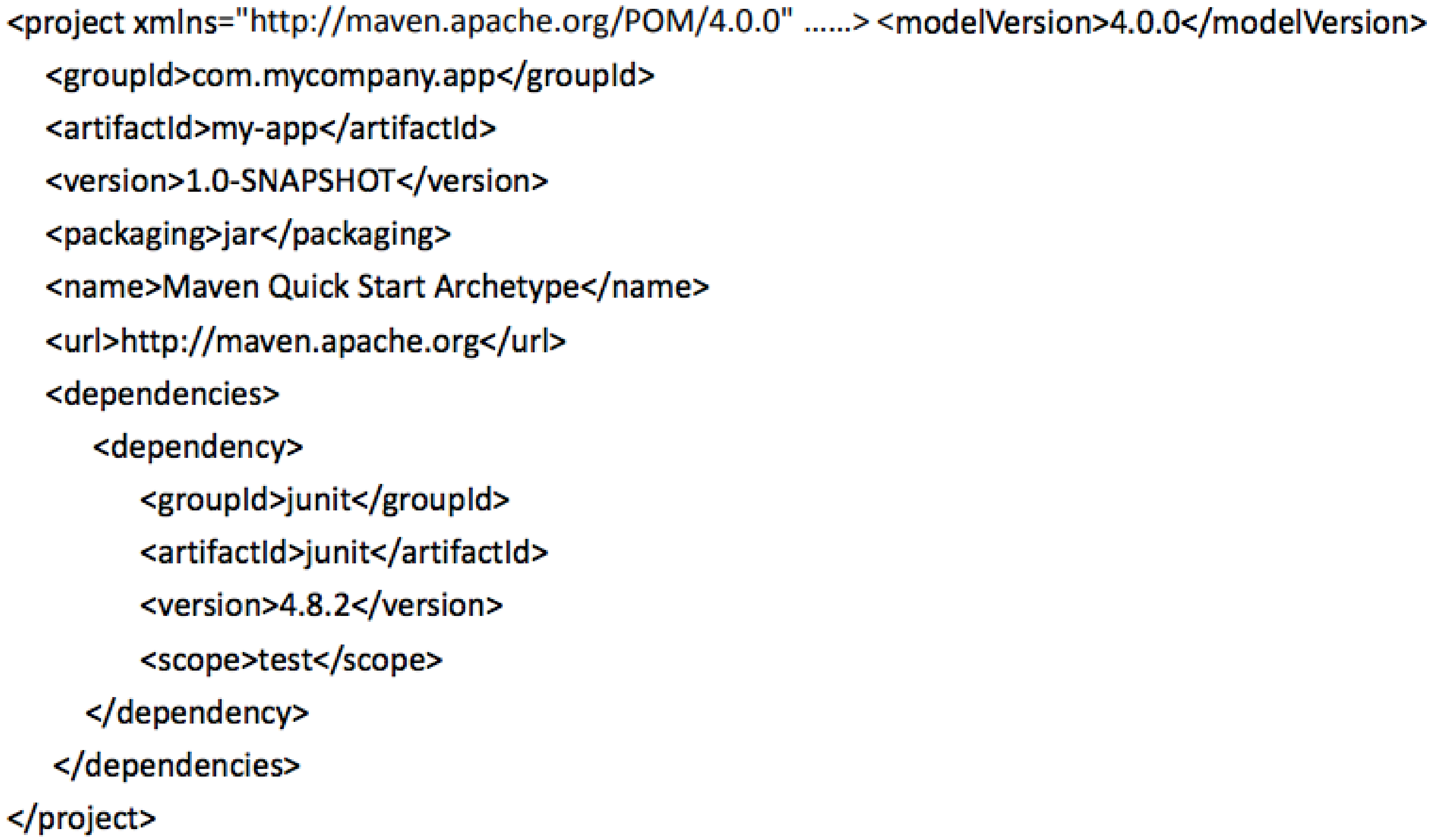
- Help manage your code across whole project via CVS
- Help maintain all dependencies (software, tools etc.) via runtime download
- Allow extensions via plug-in

Standard maven project structure



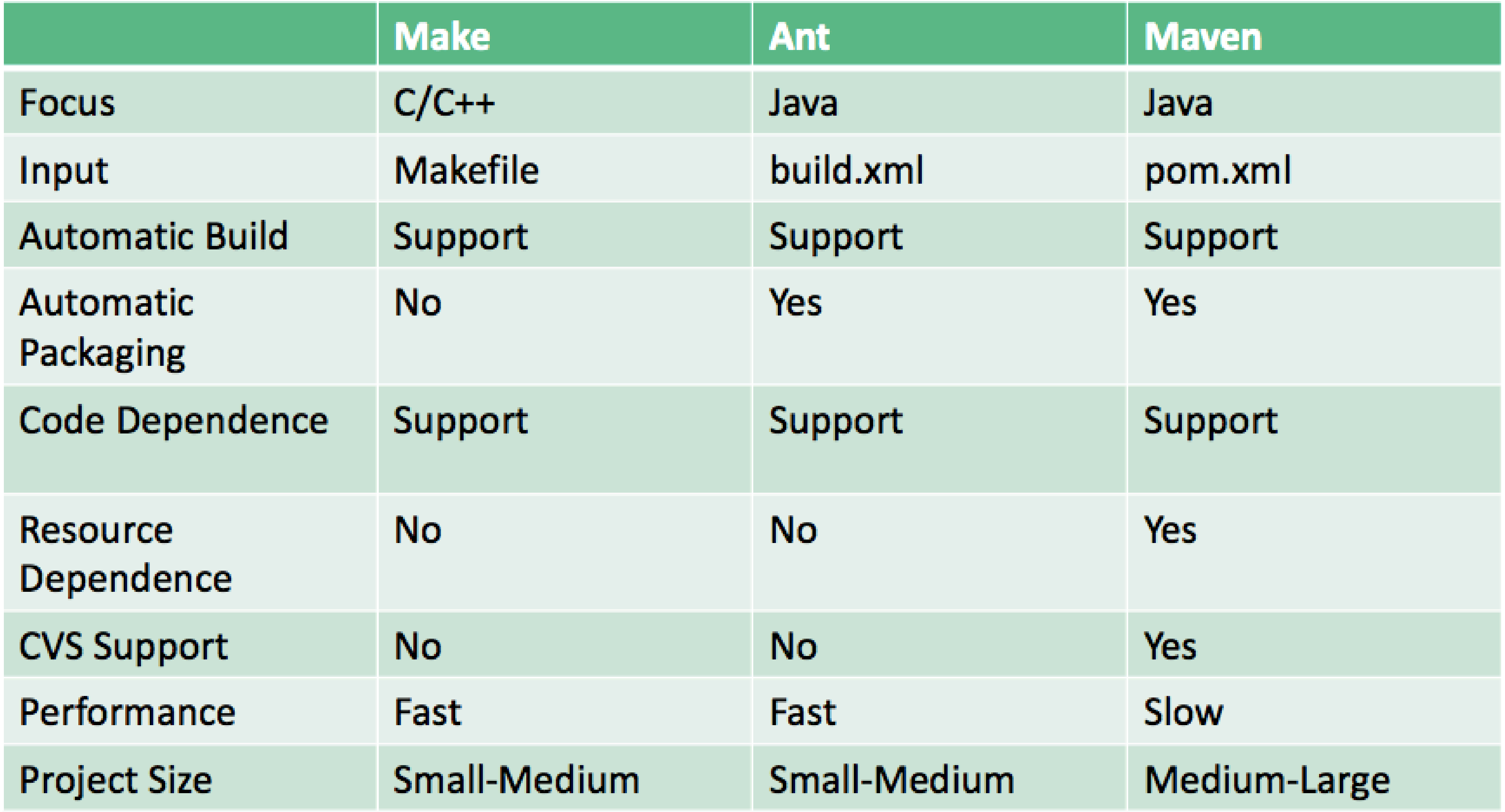
```
my-app
|-- pom.xml
`-- src
     |-- main
     |    `-- java
     |         `-- com
     |              `-- mycompany
     |                   `-- app
     |                        `-- App.java
     `-- test
          `-- java
               `-- com
                    `-- mycompany
                         `-- app
                              `-- AppTest.java
```

pom.xml



```
<project xmlns="http://maven.apache.org/POM/4.0.0" ......> <modelVersion>4.0.0</modelVersion>
   <groupId>com.mycompany.app</groupId>
   <artifactId>my-app</artifactId>
   <version>1.0-SNAPSHOT</version>
   <packaging>jar</packaging>
   <name>Maven Quick Start Archetype</name>
   <url>http://maven.apache.org</url>
   <dependencies>
      <dependency>
         <groupId>junit</groupId>
         <artifactId>junit</artifactId>
         <version>4.8.2</version>
         <scope>test</scope>
     </dependency>
    </dependencies>
</project>
```

Comparison

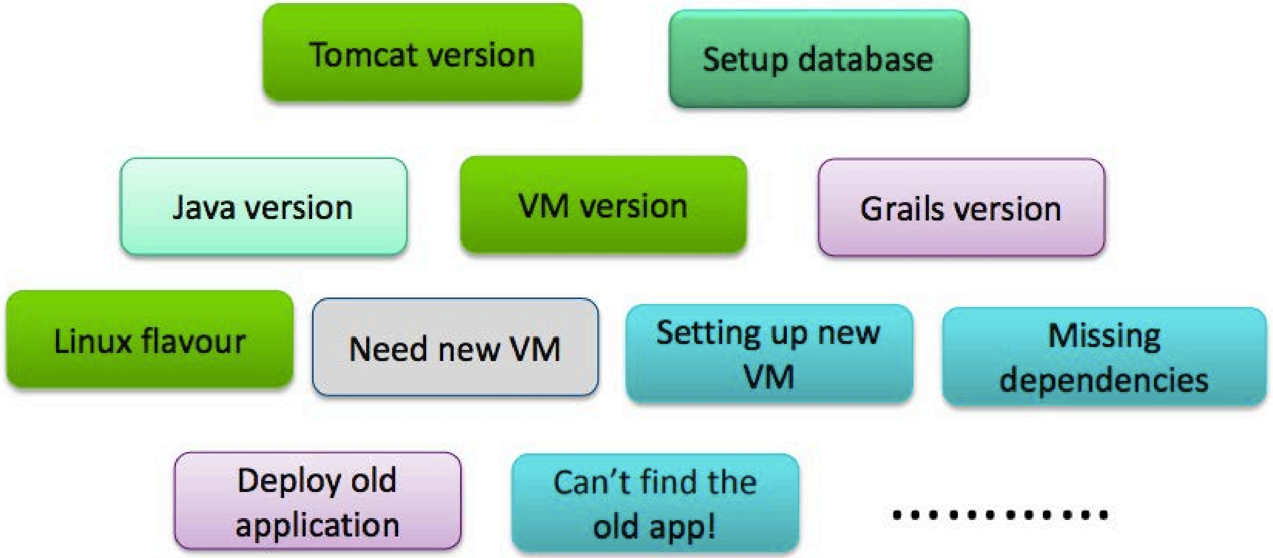


	Make	Ant	Maven
Focus	C/C++	Java	Java
Input	Makefile	build.xml	pom.xml
Automatic Build	Support	Support	Support
Automatic Packaging	No	Yes	Yes
Code Dependence	Support	Support	Support
Resource Dependence	No	No	Yes
CVS Support	No	No	Yes
Performance	Fast	Fast	Slow
Project Size	Small-Medium	Small-Medium	Medium-Large

docker

Why Docker?

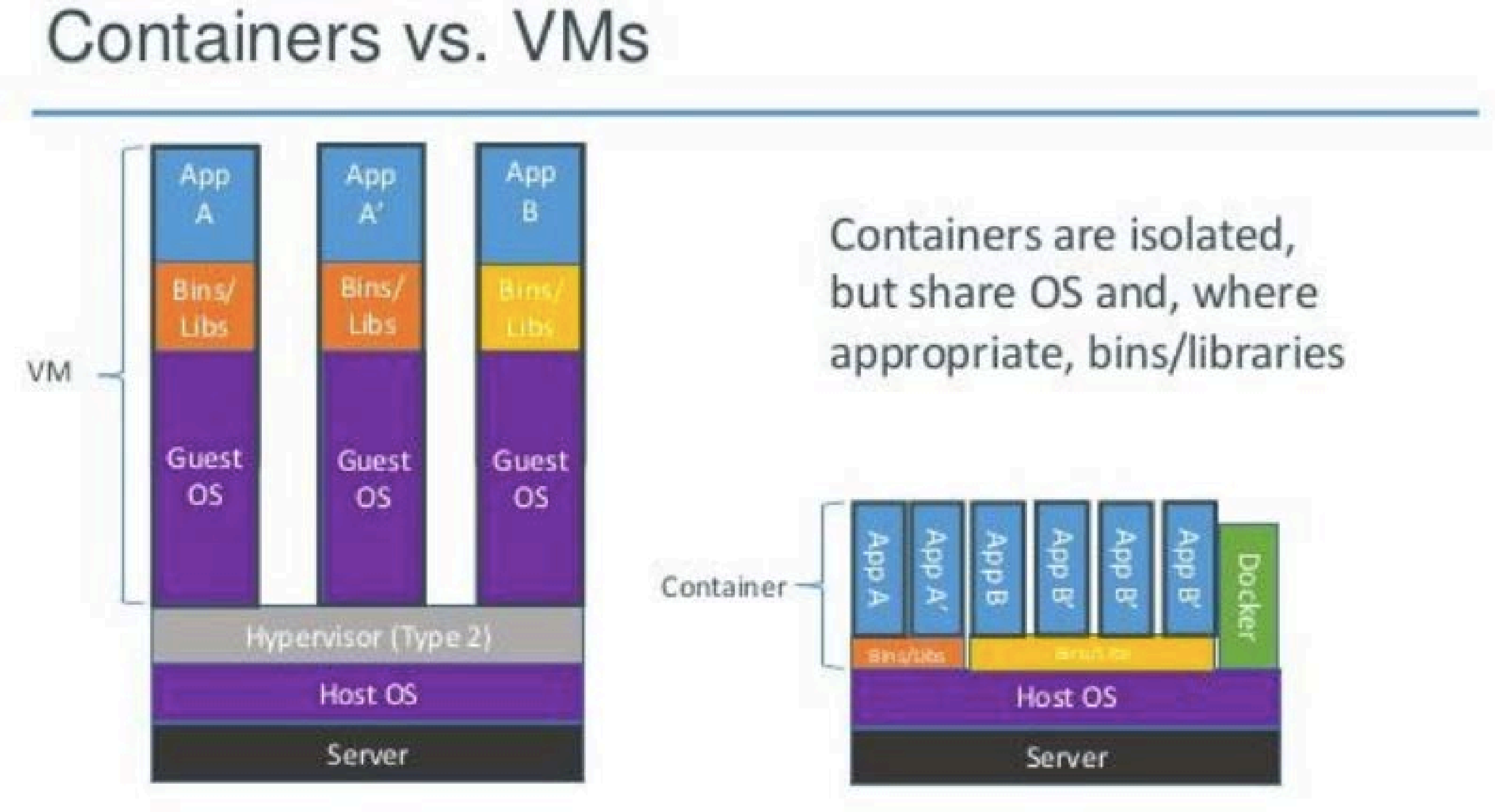
- It is becoming difficulty to manage (building, deploying, configuring, running) large and complex applications in dynamical operational environments, due to:



What is docker?

- www.docker.com
- a software containerization platform
- Allow you to build, ship, deploy, run your application everywhere
- Docker image repository: hub.docker.com
- Open source

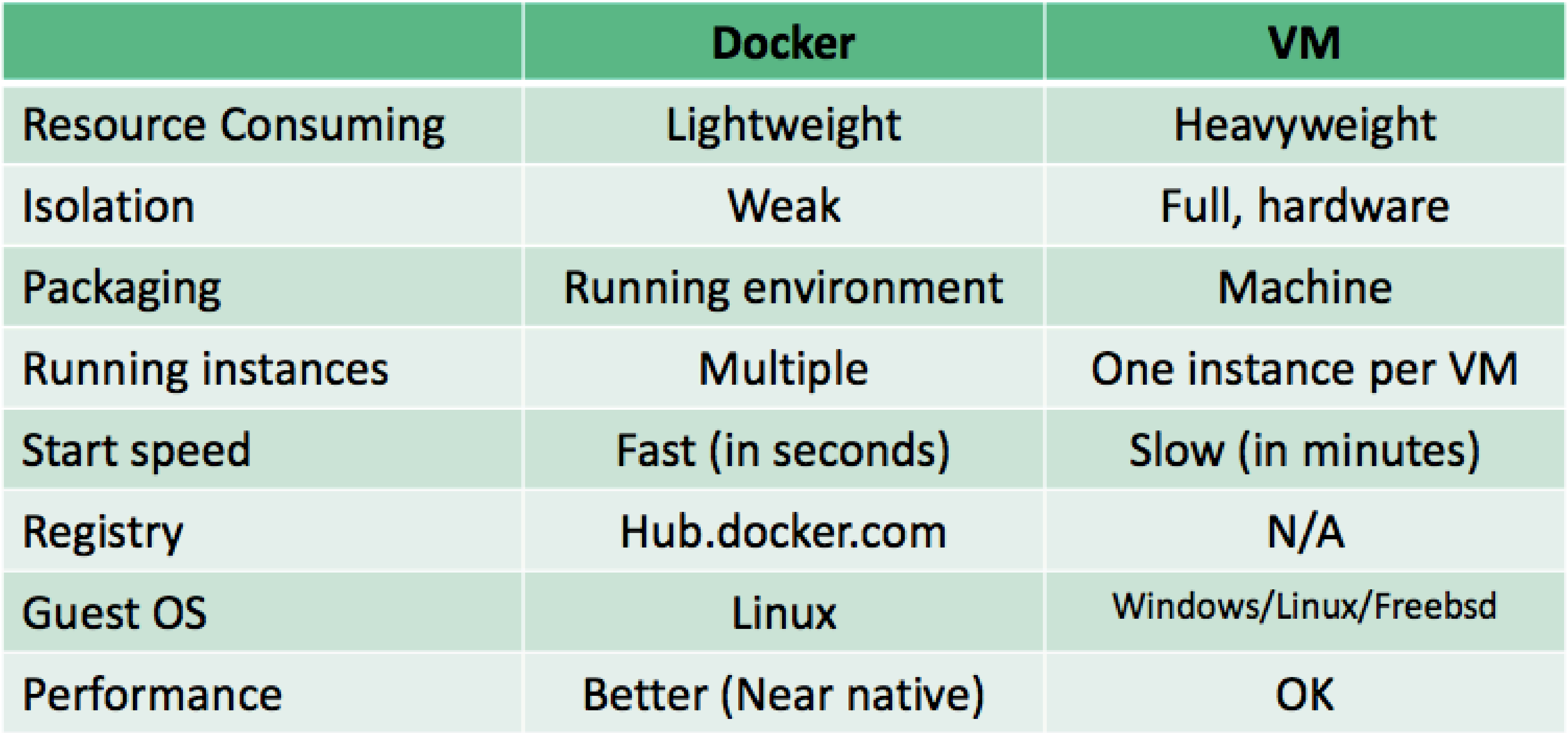
Compared with VM…



Containers vs. VMs

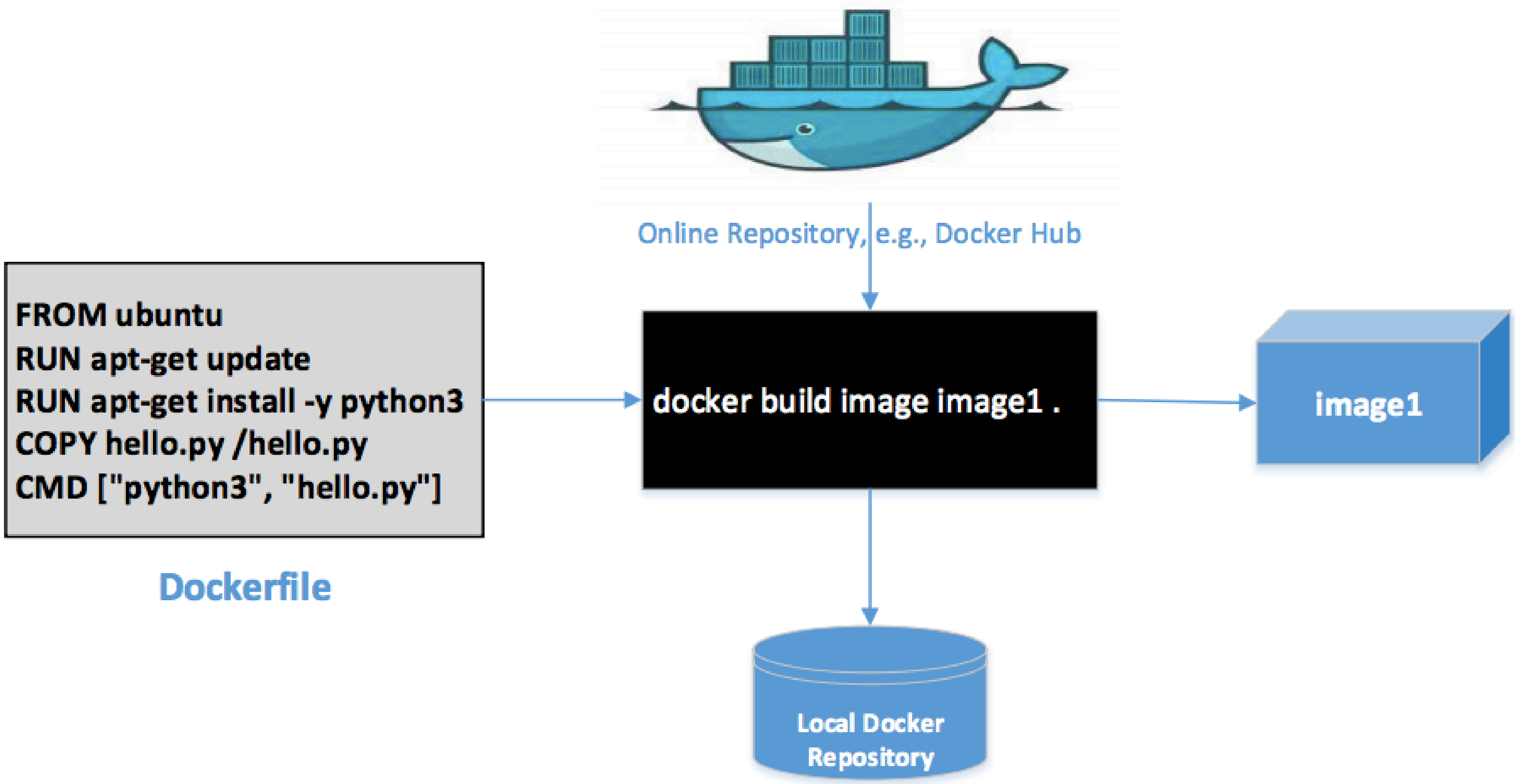
Containers are isolated, but share OS and, where appropriate, bins/libraries

Compared with VM…



	Docker	VM
Resource Consuming	Lightweight	Heavyweight
Isolation	Weak	Full, hardware
Packaging	Running environment	Machine
Running instances	Multiple	One instance per VM
Start speed	Fast (in seconds)	Slow (in minutes)
Registry	Hub.docker.com	N/A
Guest OS	Linux	Windows/Linux/Freebsd
Performance	Better (Near native)	OK

How Docker works



Online Repository, e.g., Docker Hub

```
FROM ubuntu
RUN apt-get update
RUN apt-get install -y python3
COPY hello.py /hello.py
CMD ["python3", "hello.py"]
```

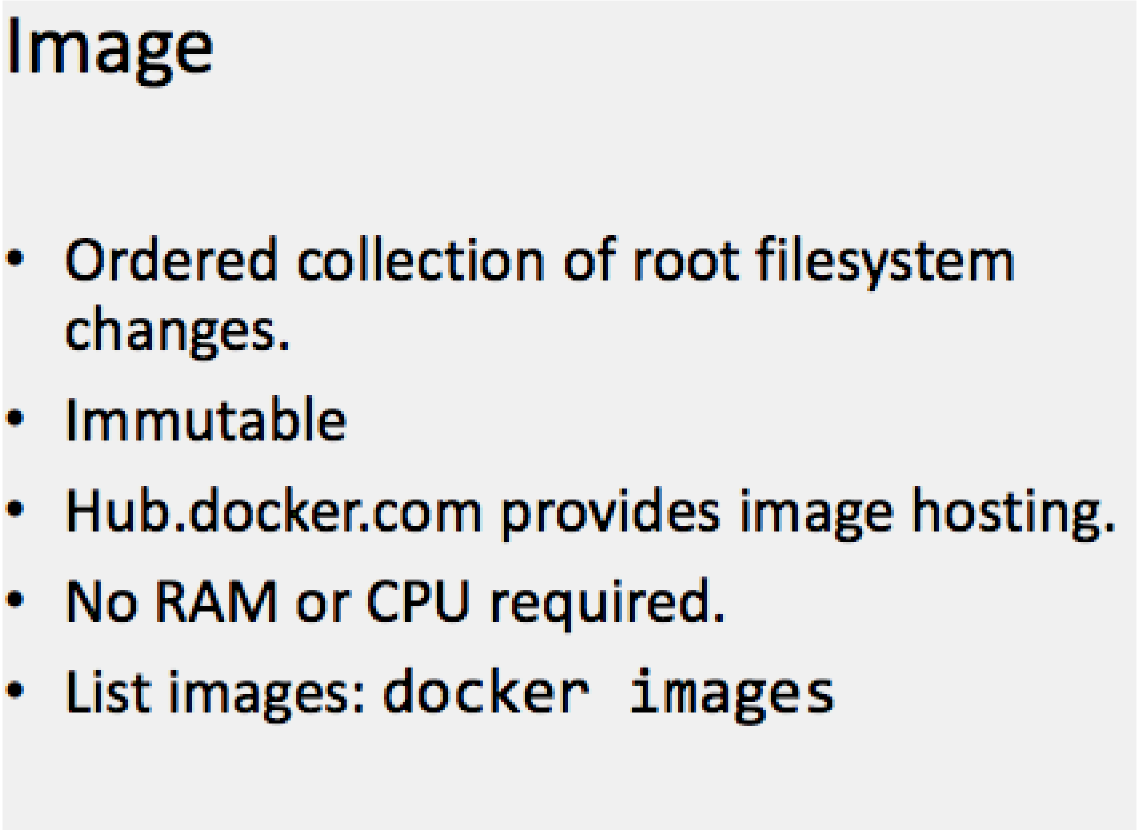
Dockerfile

```
docker build image image1 .
```

image1

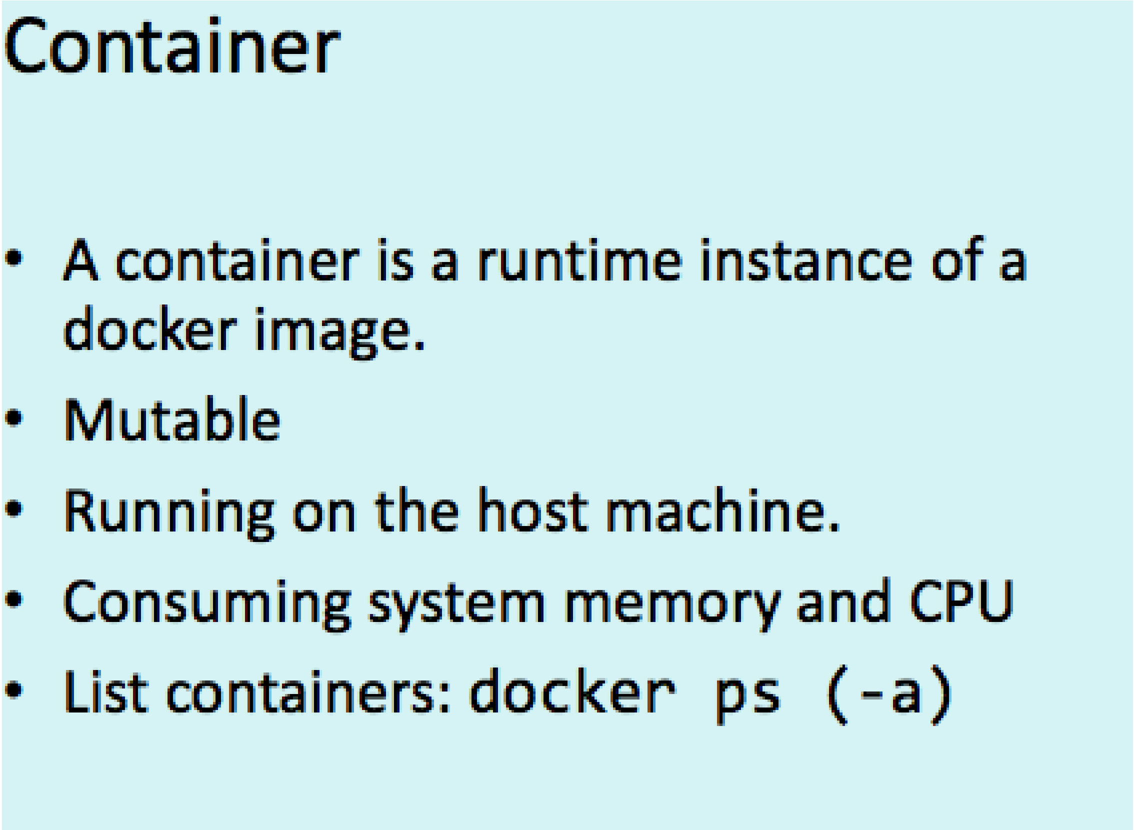
Local Docker Repository

Docker image and container



Image

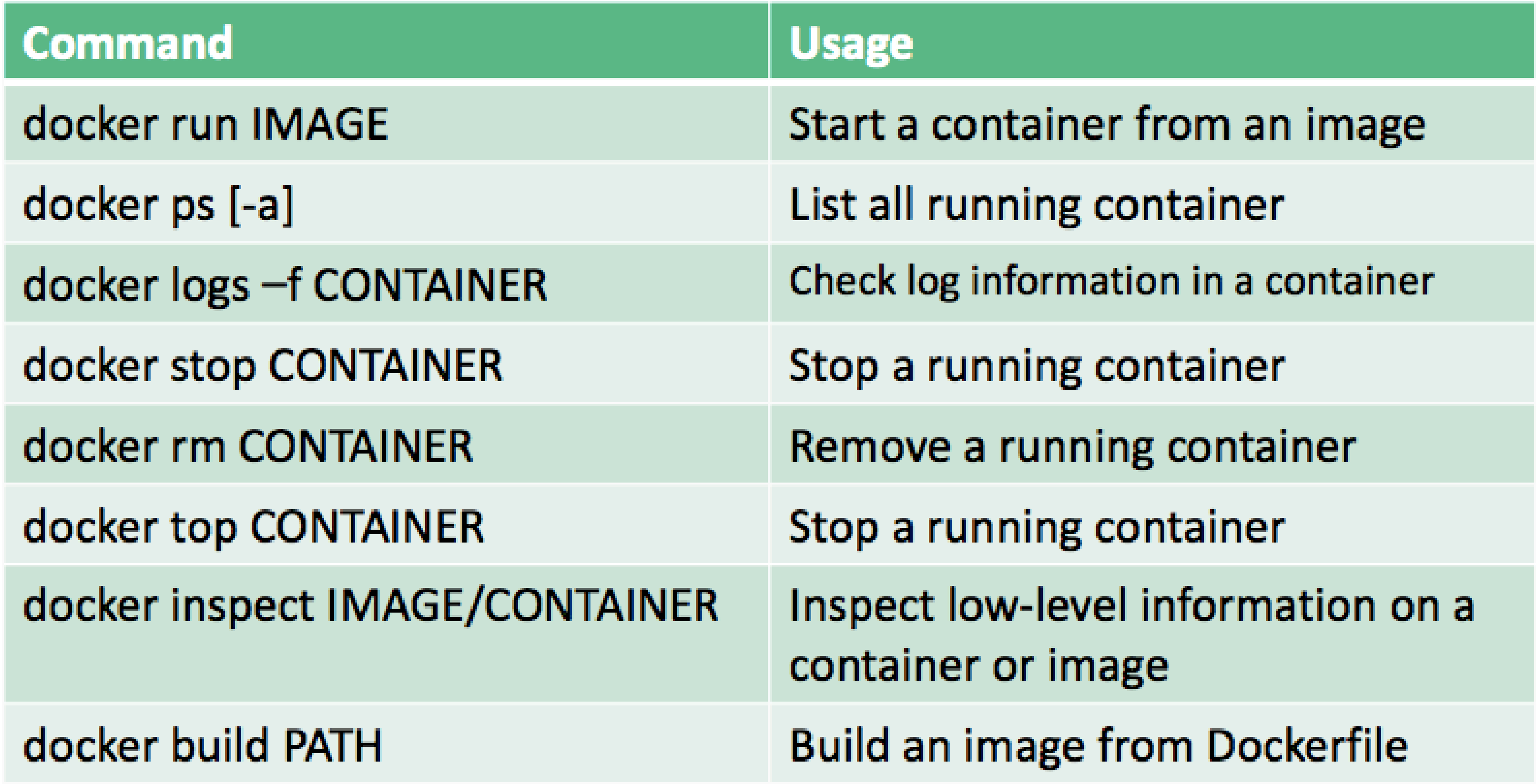
- Ordered collection of root filesystem changes.
- Immutable
- Hub.docker.com provides image hosting.
- No RAM or CPU required.
- List images: `docker images`



Container

- A container is a runtime instance of a docker image.
- Mutable
- Running on the host machine.
- Consuming system memory and CPU
- List containers: `docker ps (-a)`

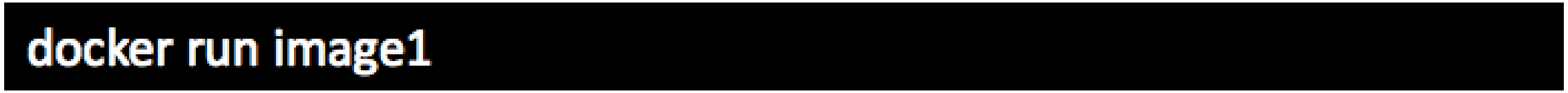
Docker commands



Command	Usage
docker run IMAGE	Start a container from an image
docker ps [-a]	List all running container
docker logs –f CONTAINER	Check log information in a container
docker stop CONTAINER	Stop a running container
docker rm CONTAINER	Remove a running container
docker top CONTAINER	Stop a running container
docker inspect IMAGE/CONTAINER	Inspect low-level information on a container or image
docker build PATH	Build an image from Dockerfile

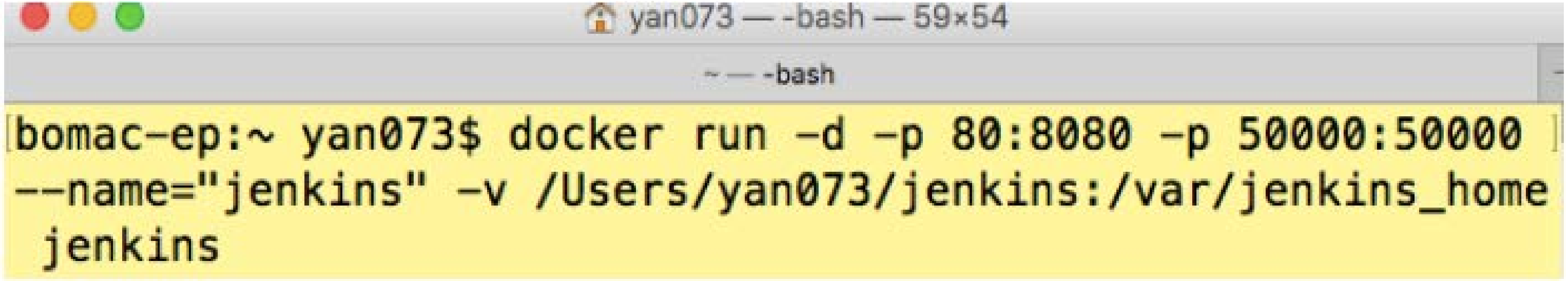
How to run docker image?

- Simple version:



```
docker run image1
```

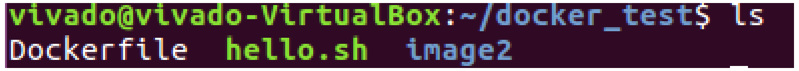
- Complex version



```
bomac-ep:~ yan073$ docker run -d -p 80:8080 -p 50000:50000
--name="jenkins" -v /Users/yan073/jenkins:/var/jenkins_home
 jenkins
```

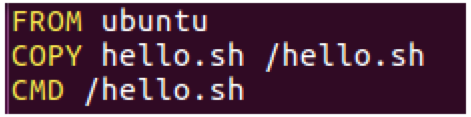
Example 1

- Directory (ls):



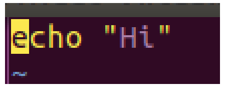
```
vivado@vivado-VirtualBox:~/docker_test$ ls
Dockerfile  hello.sh  image2
```

- Dockerfile:



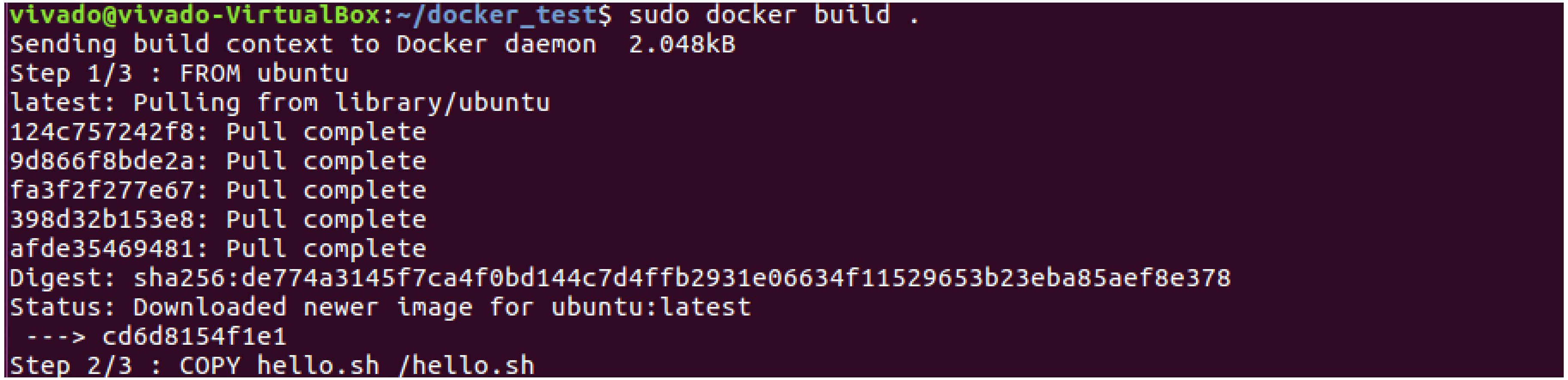
```
FROM ubuntu
COPY hello.sh /hello.sh
CMD /hello.sh
```

- Hello.sh

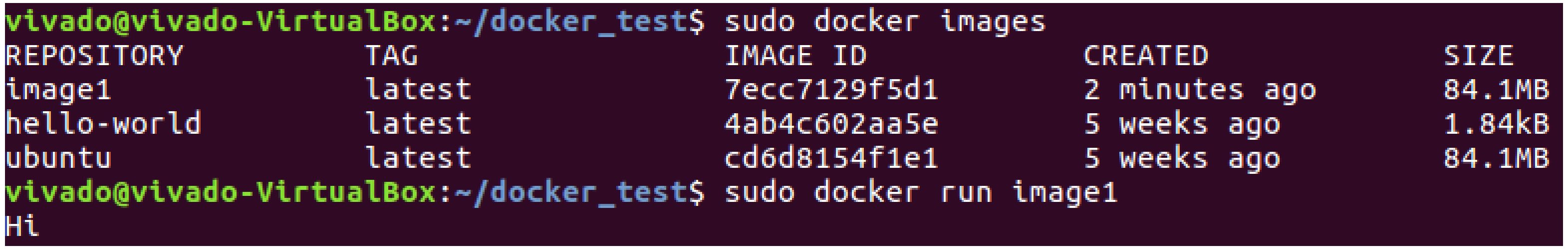


```
echo "Hi"
~
```

Example 1 build/run



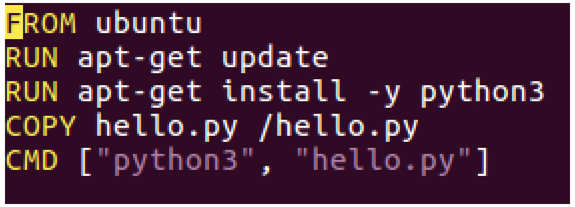
```
vivado@vivado-VirtualBox:~/docker_test$ sudo docker build .
Sending build context to Docker daemon  2.048kB
Step 1/3 : FROM ubuntu
latest: Pulling from library/ubuntu
124c757242f8: Pull complete
9d866f8bde2a: Pull complete
fa3f2f277e67: Pull complete
398d32b153e8: Pull complete
afde35469481: Pull complete
Digest: sha256:de774a3145f7ca4f0bd144c7d4ffb2931e06634f11529653b23eba85aef8e378
Status: Downloaded newer image for ubuntu:latest
 ---> cd6d8154f1e1
Step 2/3 : COPY hello.sh /hello.sh
```



```
vivado@vivado-VirtualBox:~/docker_test$ sudo docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
image1              latest              7ecc7129f5d1        2 minutes ago       84.1MB
hello-world         latest              4ab4c602aa5e        5 weeks ago         1.84kB
ubuntu              latest              cd6d8154f1e1        5 weeks ago         84.1MB
vivado@vivado-VirtualBox:~/docker_test$ sudo docker run image1
Hi
```

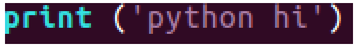
Example 2

- Dockerfile:



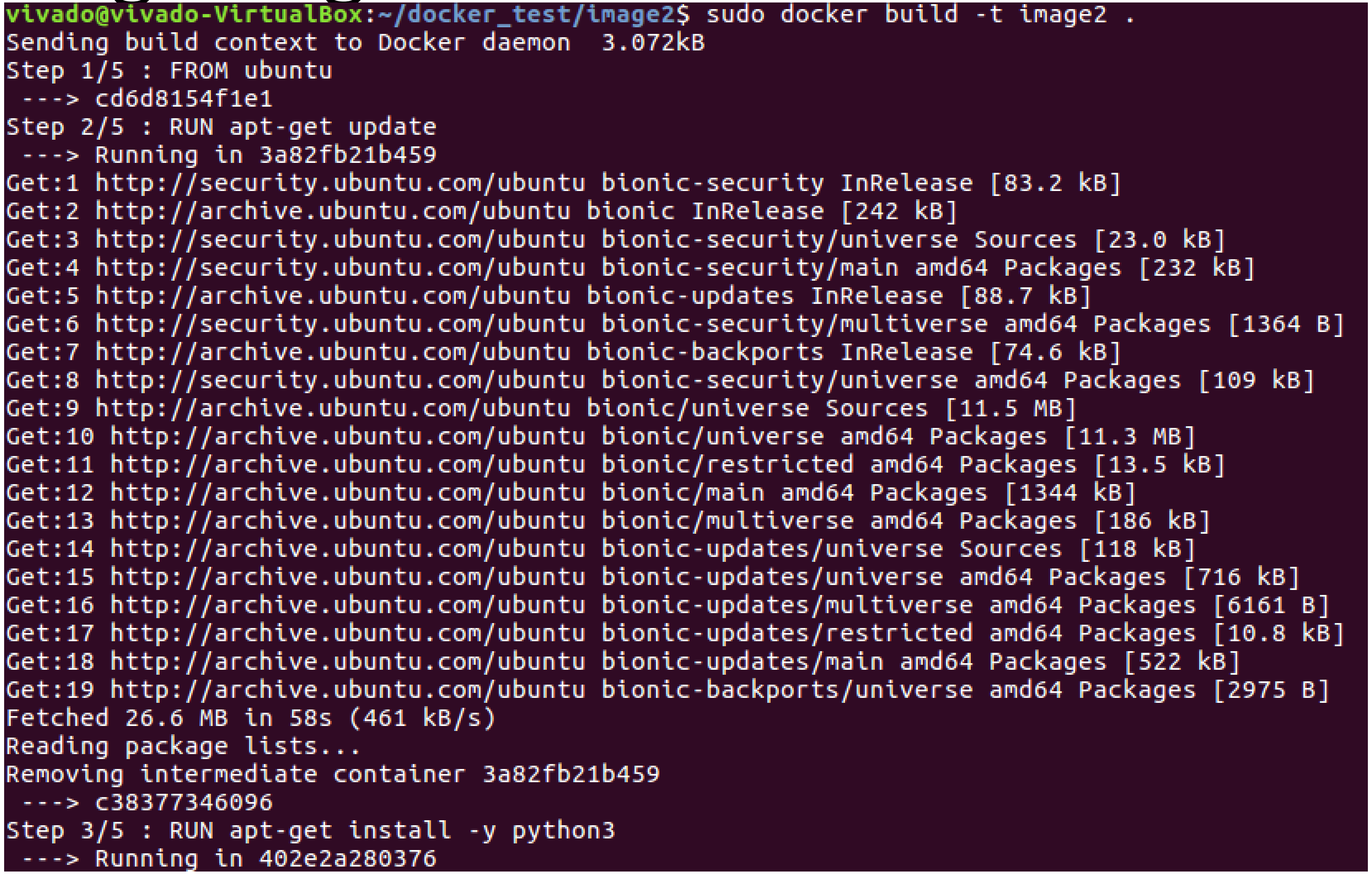
```
FROM ubuntu
RUN apt-get update
RUN apt-get install -y python3
COPY hello.py /hello.py
CMD ["python3", "hello.py"]
```

Vim hello.py



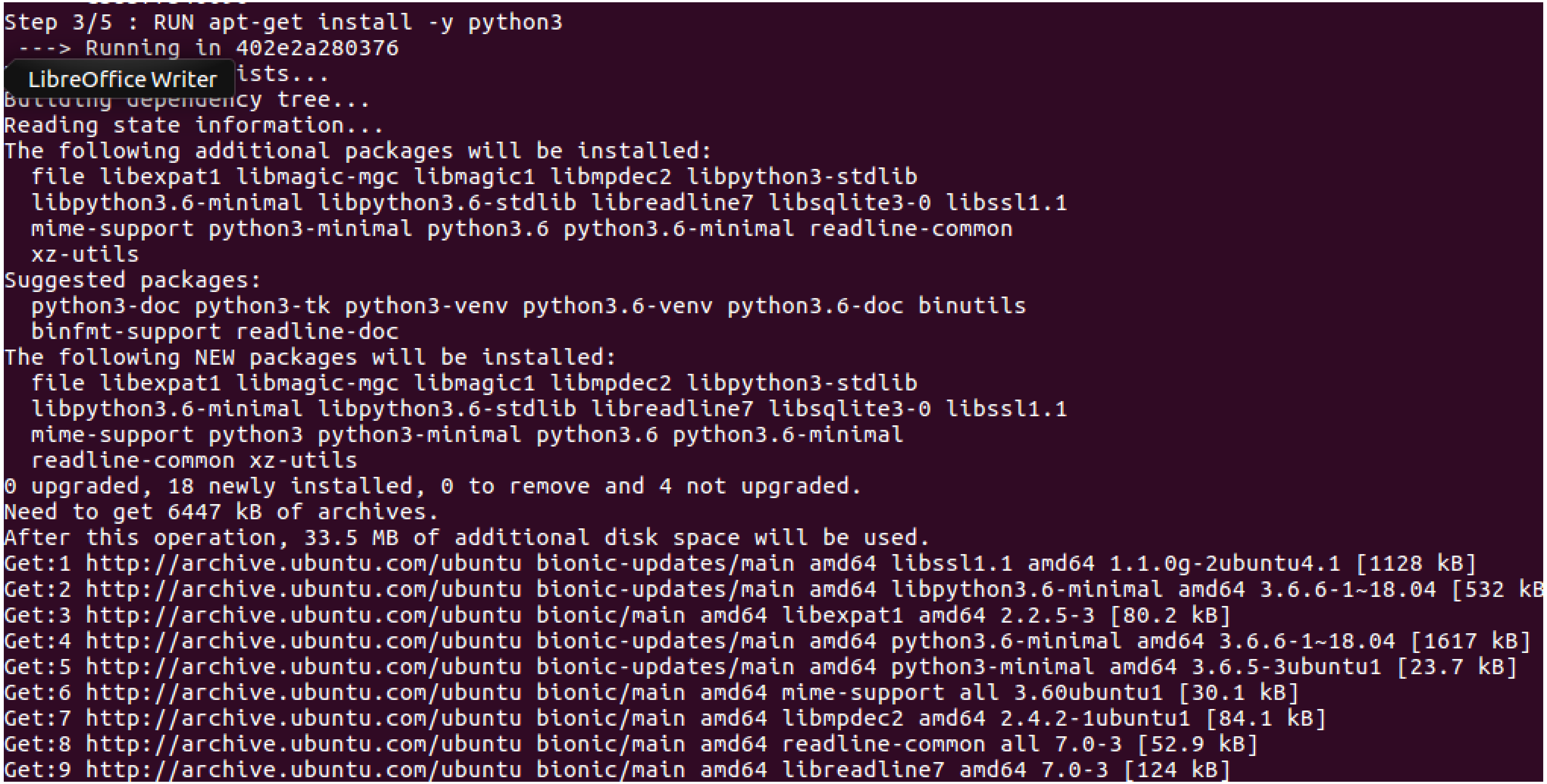
```
print ('python hi')
```

Building image 2



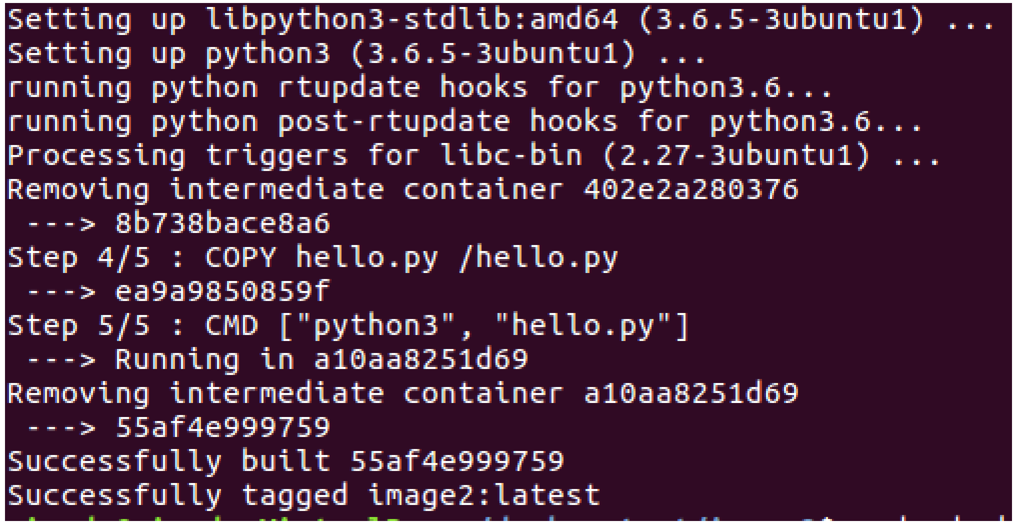
```
vivado@vivado-VirtualBox:~/docker_test/image2$ sudo docker build -t image2 .
Sending build context to Docker daemon  3.072kB
Step 1/5 : FROM ubuntu
 ---> cd6d8154f1e1
Step 2/5 : RUN apt-get update
 ---> Running in 3a82fb21b459
Get:1 http://security.ubuntu.com/ubuntu bionic-security InRelease [83.2 kB]
Get:2 http://archive.ubuntu.com/ubuntu bionic InRelease [242 kB]
Get:3 http://security.ubuntu.com/ubuntu bionic-security/universe Sources [23.0 kB]
Get:4 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [232 kB]
Get:5 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:6 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packages [1364 B]
Get:7 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:8 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [109 kB]
Get:9 http://archive.ubuntu.com/ubuntu bionic/universe Sources [11.5 MB]
Get:10 http://archive.ubuntu.com/ubuntu bionic/universe amd64 Packages [11.3 MB]
Get:11 http://archive.ubuntu.com/ubuntu bionic/restricted amd64 Packages [13.5 kB]
Get:12 http://archive.ubuntu.com/ubuntu bionic/main amd64 Packages [1344 kB]
Get:13 http://archive.ubuntu.com/ubuntu bionic/multiverse amd64 Packages [186 kB]
Get:14 http://archive.ubuntu.com/ubuntu bionic-updates/universe Sources [118 kB]
Get:15 http://archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [716 kB]
Get:16 http://archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 Packages [6161 B]
Get:17 http://archive.ubuntu.com/ubuntu bionic-updates/restricted amd64 Packages [10.8 kB]
Get:18 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [522 kB]
Get:19 http://archive.ubuntu.com/ubuntu bionic-backports/universe amd64 Packages [2975 B]
Fetched 26.6 MB in 58s (461 kB/s)
Reading package lists...
Removing intermediate container 3a82fb21b459
 ---> c38377346096
Step 3/5 : RUN apt-get install -y python3
 ---> Running in 402e2a280376
```

Cont…building…

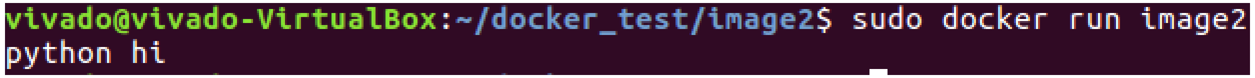


```
Step 3/5 : RUN apt-get install -y python3
 ---> Running in 402e2a280376
                 ists...
Building dependency tree...
Reading state information...
The following additional packages will be installed:
  file libexpat1 libmagic-mgc libmagic1 libmpdec2 libpython3-stdlib
  libpython3.6-minimal libpython3.6-stdlib libreadline7 libsqlite3-0 libssl1.1
  mime-support python3-minimal python3.6 python3.6-minimal readline-common
  xz-utils
Suggested packages:
  python3-doc python3-tk python3-venv python3.6-venv python3.6-doc binutils
  binfmt-support readline-doc
The following NEW packages will be installed:
  file libexpat1 libmagic-mgc libmagic1 libmpdec2 libpython3-stdlib
  libpython3.6-minimal libpython3.6-stdlib libreadline7 libsqlite3-0 libssl1.1
  mime-support python3 python3-minimal python3.6 python3.6-minimal
  readline-common xz-utils
0 upgraded, 18 newly installed, 0 to remove and 4 not upgraded.
Need to get 6447 kB of archives.
After this operation, 33.5 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libssl1.1 amd64 1.1.0g-2ubuntu4.1 [1128 kB]
Get:2 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libpython3.6-minimal amd64 3.6.6-1~18.04 [532 kB
Get:3 http://archive.ubuntu.com/ubuntu bionic/main amd64 libexpat1 amd64 2.2.5-3 [80.2 kB]
Get:4 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 python3.6-minimal amd64 3.6.6-1~18.04 [1617 kB]
Get:5 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 python3-minimal amd64 3.6.5-3ubuntu1 [23.7 kB]
Get:6 http://archive.ubuntu.com/ubuntu bionic/main amd64 mime-support all 3.60ubuntu1 [30.1 kB]
Get:7 http://archive.ubuntu.com/ubuntu bionic/main amd64 libmpdec2 amd64 2.4.2-1ubuntu1 [84.1 kB]
Get:8 http://archive.ubuntu.com/ubuntu bionic/main amd64 readline-common all 7.0-3 [52.9 kB]
Get:9 http://archive.ubuntu.com/ubuntu bionic/main amd64 libreadline7 amd64 7.0-3 [124 kB]
```

Finish building/run

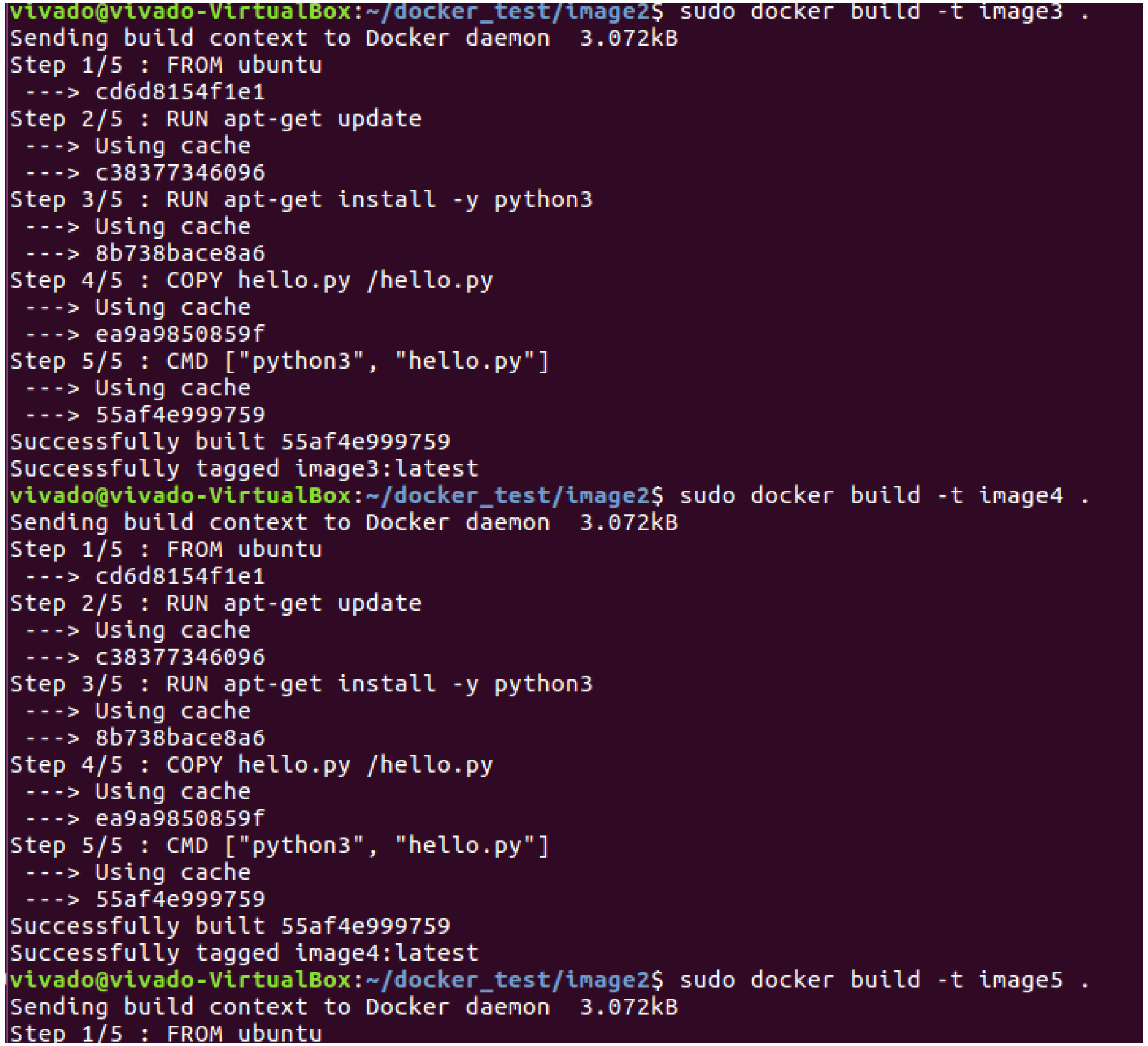


```
Setting up libpython3-stdlib:amd64 (3.6.5-3ubuntu1) ...
Setting up python3 (3.6.5-3ubuntu1) ...
running python rtupdate hooks for python3.6...
running python post-rtupdate hooks for python3.6...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
Removing intermediate container 402e2a280376
 ---> 8b738bace8a6
Step 4/5 : COPY hello.py /hello.py
 ---> ea9a98508 59f
Step 5/5 : CMD ["python3", "hello.py"]
 ---> Running in a10aa8251d69
Removing intermediate container a10aa8251d69
 ---> 55af4e999759
Successfully built 55af4e999759
Successfully tagged image2:latest
```

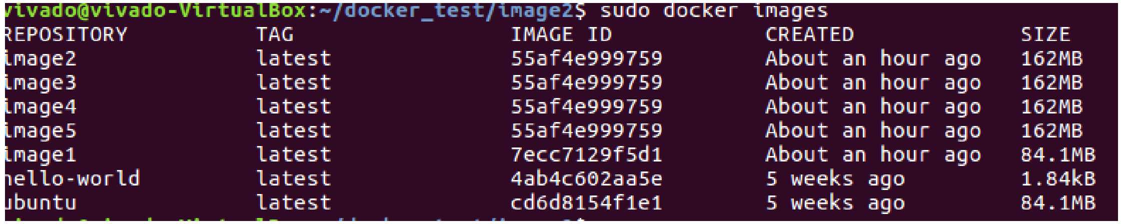


```
vivado@vivado-VirtualBox:~/docker_test/image2$ sudo docker run image2
python hi
```

Multiple containers



```
vivado@vivado-VirtualBox:~/docker_test/image2$ sudo docker build -t image3 .
Sending build context to Docker daemon  3.072kB
Step 1/5 : FROM ubuntu
 ---> cd6d8154f1e1
Step 2/5 : RUN apt-get update
 ---> Using cache
 ---> c38377346096
Step 3/5 : RUN apt-get install -y python3
 ---> Using cache
 ---> 8b738bace8a6
Step 4/5 : COPY hello.py /hello.py
 ---> Using cache
 ---> ea9a9850859f
Step 5/5 : CMD ["python3", "hello.py"]
 ---> Using cache
 ---> 55af4e999759
Successfully built 55af4e999759
Successfully tagged image3:latest
vivado@vivado-VirtualBox:~/docker_test/image2$ sudo docker build -t image4 .
Sending build context to Docker daemon  3.072kB
Step 1/5 : FROM ubuntu
 ---> cd6d8154f1e1
Step 2/5 : RUN apt-get update
 ---> Using cache
 ---> c38377346096
Step 3/5 : RUN apt-get install -y python3
 ---> Using cache
 ---> 8b738bace8a6
Step 4/5 : COPY hello.py /hello.py
 ---> Using cache
 ---> ea9a9850859f
Step 5/5 : CMD ["python3", "hello.py"]
 ---> Using cache
 ---> 55af4e999759
Successfully built 55af4e999759
Successfully tagged image4:latest
vivado@vivado-VirtualBox:~/docker_test/image2$ sudo docker build -t image5 .
Sending build context to Docker daemon  3.072kB
Step 1/5 : FROM ubuntu
```



```
vivado@vivado-VirtualBox:~/docker_test/image2$ sudo docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
image2              latest              55af4e999759        About an hour ago   162MB
image3              latest              55af4e999759        About an hour ago   162MB
image4              latest              55af4e999759        About an hour ago   162MB
image5              latest              55af4e999759        About an hour ago   162MB
image1              latest              7ecc7129f5d1        About an hour ago   84.1MB
hello-world         latest              4ab4c602aa5e        5 weeks ago         1.84kB
ubuntu              latest              cd6d8154f1e1        5 weeks ago         84.1MB
```