1)
Docker container contains code and environment. So libraries,system tools, runtime.
Docker is lightweight and portable, so that the application runs consistently on each operating system. The container achieves this by using the underlying OS resources directly.

Virtual machine allows to run virtual computers on one machine. The vm contains its own OS and so on.
2)
frequently integrate code into a shared repository and run automated tests.

3)
used commands:
mvn clean package to get jar

dockerfile:

```
                FROM ubuntu:latest
# Creates a layer that updates the package list
RUN apt-get update
# Creates a layer that installs Python3
RUN apt-get install -y openjdk-21-jre-headless
# Creates a layer that copies files from the local directory to
COPY . /app
# Sets the default command to run the application
ENTRYPOINT ["java", "-jar", "/app/checker.jar" ]
CMD []
```

```
sudo docker build -t firstdocker .
```

```
sudo docker run firstdocker https://www.frankfurt-university.de/
```

4)
Jenkins
5)
only run tests where the underlying code was changed since the last tests.
Start the test that failed last first.
Use faster hardware by temporarily using cloud resources to execute your tests.
Avoid re downloading or rebuilding.
Split tests into multiple threads and run them in parallel.
Dont wait for real API or database responses in integration tests.