**#2 Switching Java Versions (Aarav):**

**(Do not change Java version permanently)**

**#!/bin/bash**

**export JAVA\_HOME=/usr/local/openjdk-8**

**export PATH="$JAVA\_HOME/bin:$PATH"**

**MODULE\_PATH="$1"**

**if [ -z "$MODULE\_PATH" ]; then**

**echo "Usage: $0 <module\_path>"**

**exit 1**

**fi**

**cd /workspace/target-project || exit 1**

**echo "testing Step 1"**

**mvn clean install -pl "$MODULE\_PATH" -am -U \**

**-DskipTests -Ddependency-check.skip=true -Dgpg.skip=true -DfailIfNoTests=false \**

**-Dskip.installnodenpm -Dskip.npm -Dskip.yarn -Dlicense.skip -Dcheckstyle.skip -Drat.skip \**

**-Denforcer.skip -Danimal.sniffer.skip -Dmaven.javadoc.skip -Dfindbugs.skip -Dwarbucks.skip \**

**-Dmodernizer.skip -Dimpsort.skip -Dmdep.analyze.skip -Dpgpverify.skip -Dxml.skip \**

**-Dcobertura.skip=true -Dfindbugs.skip=true**

**dines@PC1:~/my\_docker\_project$ cat step\_two.sh**

**#!/bin/bash**

**MODULE\_PATH="$1"**

**# List of Java versions to try**

**JAVA\_VERSIONS=(**

**"/usr/local/openjdk-8"**

**"/usr/lib/jvm/java-11-openjdk-amd64"**

**"/usr/lib/jvm/java-17-openjdk-amd64"**

**)**

**for JAVA\_HOME\_PATH in "${JAVA\_VERSIONS[@]}"; do**

**if [ -x "$JAVA\_HOME\_PATH/bin/java" ]; then**

**export JAVA\_HOME="$JAVA\_HOME\_PATH"**

**export PATH="$JAVA\_HOME/bin:$PATH"**

**echo "Trying Java version: $("$JAVA\_HOME/bin/java" -version 2>&1 | head -n 1)"**

**/workspace/step\_one.sh "$MODULE\_PATH"**

**status=$?**

**if [ $status -eq 0 ]; then**

**echo "Success with Java at $JAVA\_HOME\_PATH"**

**exit 0**

**else**

**echo "Failed with Java at $JAVA\_HOME\_PATH"**

**fi**

**else**

**echo "Java not found at $JAVA\_HOME\_PATH"**

**fi**

**done**

**echo "All Java versions failed"**

**exit 1**

**#4 Changing http to https in pom.xml (Aarav):**

**(See which lines really need to be changed)**

**#!/bin/bash**

**POM\_FILE="pom.xml"**

**if [ ! -f "$POM\_FILE" ]; then**

**echo "No pom.xml file found in current directory."**

**exit 1**

**fi**

**# Backup original pom.xml**

**cp "$POM\_FILE" "${POM\_FILE}.bak"**

**# Replace all http:// with https:// in the context of dependencies and repositories**

**sed -i 's|<url>http://|<url>https://|g' "$POM\_FILE"**

**sed -i 's|<repositoryUrl>http://|<repositoryUrl>https://|g' "$POM\_FILE"**

**sed -i 's|<connection>http://|<connection>https://|g' "$POM\_FILE"**

**sed -i 's|<developerConnection>http://|<developerConnection>https://|g' "$POM\_FILE"**

**sed -i 's|<distributionManagement>.\*http://|<distributionManagement>https://|g' "$POM\_FILE"**

**sed -i 's|http://repo1.maven.org/maven2|https://repo1.maven.org/maven2|g' "$POM\_FILE"**

**sed -i 's|http://|https://|g' "$POM\_FILE" # fallback, more aggressive (can be commented if too broad)**

**echo "All applicable http:// URLs in $POM\_FILE have been replaced with https://"**

**exit 0**

Step One (Annikka)

#!/bin/bash

export JAVA\_HOME=/usr/lib/jvm/java-1.8.0-openjdk-arm64

export PATH="$JAVA\_HOME/bin:$PATH"

MODULE\_PATH="$1"

if [ -z "$MODULE\_PATH" ]; then

echo "Usage: $0 <module\_path>"

exit 1

fi

cd /workspace/target-project || exit 1

echo "testing Step 1"

mvn clean install -pl "$MODULE\_PATH" -am -U \

-DskipTests -Ddependency-check.skip=true -Dgpg.skip=true -DfailIfNoTests=false \

-Dskip.installnodenpm -Dskip.npm -Dskip.yarn -Dlicense.skip -Dcheckstyle.skip -Drat.skip \

-Denforcer.skip -Danimal.sniffer.skip -Dmaven.javadoc.skip -Dfindbugs.skip -Dwarbucks.skip \

-Dmodernizer.skip -Dimpsort.skip -Dmdep.analyze.skip -Dpgpverify.skip -Dxml.skip \

-Dcobertura.skip=true -Dfindbugs.skip=true

Step 3 (Annikka)

#!/bin/bash

versions=$(grep -rohP '<version>\K[0-9A-Za-z\.\-]+(?=-SNAPSHOT</version>)' --include=pom.xml . | sort -u)

for version in $versions; do

read -p "Replace ${version}-SNAPSHOT with what? " stable\_ver

find . -name "pom.xml" -exec sed -i.bak "s/${version}-SNAPSHOT/${stable\_ver}/g" {} \;

echo "${version}-SNAPSHOT has been replaced with ${stable\_ver}"

done

Super script

#!/bin/bash

REPO\_URL="$1"

COMMIT\_SHA="$2"

MODULE\_PATH="$3"

if [ -z "$REPO\_URL" ] || [ -z "$COMMIT\_SHA" ] || [ -z "$MODULE\_PATH" ]; then

echo "Usage: $0 <repo\_url> <commit\_sha> <module\_path>"

exit 1

fi

rm -rf target-project

git clone "$REPO\_URL" target-project

cd target-project || exit 1

git checkout "$COMMIT\_SHA"

SHA=$(git rev-parse HEAD)

LOG\_DIR="/root/logs"

mkdir -p "$LOG\_DIR"

cd "$MODULE\_PATH" || exit 1

check(){

if [ $1 -eq 0 ]; then

echo "success"

exit 0

fi

}

log\_output(){

steps\_combo="$1"

shift

echo "Running $steps\_combo"

log\_file="$LOG\_DIR/${SHA}\_\_${steps\_combo}.log"

{

echo "===== $steps\_combo ====="

"$@"

} &> "$log\_file"

status=$?

echo "Exit code for $steps\_combo: $status"

check $status

echo

}

log\_output "Step 1" /workspace/step\_one.sh "$MODULE\_PATH"

git reset --hard HEAD

git clean -xfd

log\_output "Step 1 + 2" /workspace/step\_two.sh "$MODULE\_PATH"

git reset --hard HEAD

git clean -xfd

log\_output "Step 1 + 3" bash -c "/workspace/step\_three.sh && /workspace/step\_one.sh '$MODULE\_PATH'"

git reset --hard HEAD

git clean -xfd

log\_output "Step 1 + 4" bash -c "/workspace/step\_four.sh && /workspace/step\_one.sh '$MODULE\_PATH'"

git reset --hard HEAD

git clean -xfd

log\_output "Step 1 + 2 + 3" bash -c "/workspace/step\_two.sh '$MODULE\_PATH' && /workspace/step\_three.sh"

git reset --hard HEAD

git clean -xfd

log\_output "Step 1 + 2 + 4" bash -c "/workspace/step\_two.sh '$MODULE\_PATH' && /workspace/step\_four.sh"

git reset --hard HEAD

git clean -xfd

log\_output "Step 1 + 3 + 4" bash -c "/workspace/step\_three.sh && /workspace/step\_four.sh && /workspace/step\_one.sh '$MODULE\_PATH'"

git reset --hard HEAD

git clean -xfd

log\_output "Step 1 + 2 + 3 + 4" bash -c "/workspace/step\_two.sh '$MODULE\_PATH' && /workspace/step\_three.sh && /workspace/step\_four.sh"

Dockerfile (Annikka)

FROM maven:3.8.6-openjdk-8

# Install Java 11 and 17

RUN apt-get update && apt-get install -y openjdk-11-jdk openjdk-17-jdk && \

apt-get clean && rm -rf /var/lib/apt/lists/\*

# Set up working directory

WORKDIR /workspace

COPY . .

# Make scripts executable

RUN chmod +x super\_script.sh step\_one.sh step\_two.sh step\_three.sh step\_four.sh

ENTRYPOINT ["./super\_script.sh"]

CMD ["https://github.com/Adyen/adyen-java-api-library", "6fb5cd049b57a22d2ec4465d204c15f1c90dd325", "."]

Docker Run Command (Aarav):

docker run --rm --name <containerName> <imageName \

<githubRepoURL> \

<SHACommit \

<PathToModule>

docker run -t --rm -v ${SCRIPT\_DIR}:/Scratch ${image} /bin/bash -x /Scratch/run\_experiment.sh ${slug} ${rounds} ${timeout} "${script}"

docker run -t --rm -v $super\_script.sh:/Scratch ${image} /bin/bash -x /Scratch/run\_experiment.sh ${slug} ${rounds} ${timeout} "${script}"