***.gitignore***

HELP.md

target/

!.mvn/wrapper/maven-wrapper.jar

!\*\*/src/main/\*\*/target/

!\*\*/src/test/\*\*/target/

### STS ###

.apt\_generated

.classpath

.factorypath

.project

.settings

.springBeans

.sts4-cache

### IntelliJ IDEA ###

.idea

\*.iws

\*.iml

\*.ipr

### NetBeans ###

/nbproject/private/

/nbbuild/

/dist/

/nbdist/

/.nb-gradle/

build/

!\*\*/src/main/\*\*/build/

!\*\*/src/test/\*\*/build/

### VS Code ###

.vscode/

***buildspec.yml***

version: 0.2

phases:

pre\_build:

commands:

- echo Logging in to Amazon ECR....

# Check AWS CLI version

- aws --version

# Log in to Amazon ECR

- aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin 577158399875.dkr.ecr.us-east-1.amazonaws.com

# Define repository URI and image tag

- REPOSITORY\_URI=577158399875.dkr.ecr.us-east-1.amazonaws.com/iitk-proj-awsdevops

- IMAGE\_TAG=build-$(echo $CODEBUILD\_BUILD\_ID | awk -F":" '{print $2}')

build:

commands:

- echo Build started on date

- echo Building the JAR file...

# Build the Java project using Maven

- mvn clean install

- echo Building the Docker image...

# Build and tag Docker image

- docker build -t $REPOSITORY\_URI:latest .

- docker tag $REPOSITORY\_URI:latest $REPOSITORY\_URI:$IMAGE\_TAG

post\_build:

commands:

- echo Build completed on date

- echo Pushing the Docker image to ECR...

# Push Docker images to ECR

- docker push $REPOSITORY\_URI:latest

- docker push $REPOSITORY\_URI:$IMAGE\_TAG

- echo Writing image definitions file...

# Define container name and create imagedefinitions.json for ECS

- DOCKER\_CONTAINER\_NAME=springboot-app

- printf '[{"name":"%s","imageUri":"%s"}]' $DOCKER\_CONTAINER\_NAME $REPOSITORY\_URI:$IMAGE\_TAG > imagedefinitions.json

- echo $DOCKER\_CONTAINER\_NAME

- echo Displaying imagedefinitions.json content

- cat imagedefinitions.json

artifacts:

files:

- imagedefinitions.json

- target/springboot-aws-deploy.jar

***DockerFile***

FROM eclipse-temurin:17-jdk-alpine

RUN apk add curl

VOLUME /tmp

EXPOSE 8080

ADD target/springboot-aws-deploy-service.jar springboot-aws-deploy-service.jar

ENTRYPOINT ["java","-jar","/springboot-aws-deploy-service.jar"]

***mvnw***

#!/bin/sh

# ----------------------------------------------------------------------------

# Licensed to the Apache Software Foundation (ASF) under one

# or more contributor license agreements. See the NOTICE file

# distributed with this work for additional information

# regarding copyright ownership. The ASF licenses this file

# to you under the Apache License, Version 2.0 (the

# "License"); you may not use this file except in compliance

# with the License. You may obtain a copy of the License at

#

# https://www.apache.org/licenses/LICENSE-2.0

#

# Unless required by applicable law or agreed to in writing,

# software distributed under the License is distributed on an

# "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY

# KIND, either express or implied. See the License for the

# specific language governing permissions and limitations

# under the License.

# ----------------------------------------------------------------------------

# ----------------------------------------------------------------------------

# Apache Maven Wrapper startup batch script, version 3.2.0

#

# Required ENV vars:

# ------------------

# JAVA\_HOME - location of a JDK home dir

#

# Optional ENV vars

# -----------------

# MAVEN\_OPTS - parameters passed to the Java VM when running Maven

# e.g. to debug Maven itself, use

# set MAVEN\_OPTS=-Xdebug -Xrunjdwp:transport=dt\_socket,server=y,suspend=y,address=8000

# MAVEN\_SKIP\_RC - flag to disable loading of mavenrc files

# ----------------------------------------------------------------------------

if [ -z "$MAVEN\_SKIP\_RC" ] ; then

if [ -f /usr/local/etc/mavenrc ] ; then

. /usr/local/etc/mavenrc

fi

if [ -f /etc/mavenrc ] ; then

. /etc/mavenrc

fi

if [ -f "$HOME/.mavenrc" ] ; then

. "$HOME/.mavenrc"

fi

fi

# OS specific support. $var \_must\_ be set to either true or false.

cygwin=false;

darwin=false;

mingw=false

case "$(uname)" in

CYGWIN\*) cygwin=true ;;

MINGW\*) mingw=true;;

Darwin\*) darwin=true

# Use /usr/libexec/java\_home if available, otherwise fall back to /Library/Java/Home

# See https://developer.apple.com/library/mac/qa/qa1170/\_index.html

if [ -z "$JAVA\_HOME" ]; then

if [ -x "/usr/libexec/java\_home" ]; then

JAVA\_HOME="$(/usr/libexec/java\_home)"; export JAVA\_HOME

else

JAVA\_HOME="/Library/Java/Home"; export JAVA\_HOME

fi

fi

;;

esac

if [ -z "$JAVA\_HOME" ] ; then

if [ -r /etc/gentoo-release ] ; then

JAVA\_HOME=$(java-config --jre-home)

fi

fi

# For Cygwin, ensure paths are in UNIX format before anything is touched

if $cygwin ; then

[ -n "$JAVA\_HOME" ] &&

JAVA\_HOME=$(cygpath --unix "$JAVA\_HOME")

[ -n "$CLASSPATH" ] &&

CLASSPATH=$(cygpath --path --unix "$CLASSPATH")

fi

# For Mingw, ensure paths are in UNIX format before anything is touched

if $mingw ; then

[ -n "$JAVA\_HOME" ] && [ -d "$JAVA\_HOME" ] &&

JAVA\_HOME="$(cd "$JAVA\_HOME" || (echo "cannot cd into $JAVA\_HOME."; exit 1); pwd)"

fi

if [ -z "$JAVA\_HOME" ]; then

javaExecutable="$(which javac)"

if [ -n "$javaExecutable" ] && ! [ "$(expr "\"$javaExecutable\"" : '\([^ ]\*\)')" = "no" ]; then

# readlink(1) is not available as standard on Solaris 10.

readLink=$(which readlink)

if [ ! "$(expr "$readLink" : '\([^ ]\*\)')" = "no" ]; then

if $darwin ; then

javaHome="$(dirname "\"$javaExecutable\"")"

javaExecutable="$(cd "\"$javaHome\"" && pwd -P)/javac"

else

javaExecutable="$(readlink -f "\"$javaExecutable\"")"

fi

javaHome="$(dirname "\"$javaExecutable\"")"

javaHome=$(expr "$javaHome" : '\(.\*\)/bin')

JAVA\_HOME="$javaHome"

export JAVA\_HOME

fi

fi

fi

if [ -z "$JAVACMD" ] ; then

if [ -n "$JAVA\_HOME" ] ; then

if [ -x "$JAVA\_HOME/jre/sh/java" ] ; then

# IBM's JDK on AIX uses strange locations for the executables

JAVACMD="$JAVA\_HOME/jre/sh/java"

else

JAVACMD="$JAVA\_HOME/bin/java"

fi

else

JAVACMD="$(\unset -f command 2>/dev/null; \command -v java)"

fi

fi

if [ ! -x "$JAVACMD" ] ; then

echo "Error: JAVA\_HOME is not defined correctly." >&2

echo " We cannot execute $JAVACMD" >&2

exit 1

fi

if [ -z "$JAVA\_HOME" ] ; then

echo "Warning: JAVA\_HOME environment variable is not set."

fi

# traverses directory structure from process work directory to filesystem root

# first directory with .mvn subdirectory is considered project base directory

find\_maven\_basedir() {

if [ -z "$1" ]

then

echo "Path not specified to find\_maven\_basedir"

return 1

fi

basedir="$1"

wdir="$1"

while [ "$wdir" != '/' ] ; do

if [ -d "$wdir"/.mvn ] ; then

basedir=$wdir

break

fi

# workaround for JBEAP-8937 (on Solaris 10/Sparc)

if [ -d "${wdir}" ]; then

wdir=$(cd "$wdir/.." || exit 1; pwd)

fi

# end of workaround

done

printf '%s' "$(cd "$basedir" || exit 1; pwd)"

}

# concatenates all lines of a file

concat\_lines() {

if [ -f "$1" ]; then

# Remove \r in case we run on Windows within Git Bash

# and check out the repository with auto CRLF management

# enabled. Otherwise, we may read lines that are delimited with

# \r\n and produce $'-Xarg\r' rather than -Xarg due to word

# splitting rules.

tr -s '\r\n' ' ' < "$1"

fi

}

log() {

if [ "$MVNW\_VERBOSE" = true ]; then

printf '%s\n' "$1"

fi

}

BASE\_DIR=$(find\_maven\_basedir "$(dirname "$0")")

if [ -z "$BASE\_DIR" ]; then

exit 1;

fi

MAVEN\_PROJECTBASEDIR=${MAVEN\_BASEDIR:-"$BASE\_DIR"}; export MAVEN\_PROJECTBASEDIR

log "$MAVEN\_PROJECTBASEDIR"

##########################################################################################

# Extension to allow automatically downloading the maven-wrapper.jar from Maven-central

# This allows using the maven wrapper in projects that prohibit checking in binary data.

##########################################################################################

wrapperJarPath="$MAVEN\_PROJECTBASEDIR/.mvn/wrapper/maven-wrapper.jar"

if [ -r "$wrapperJarPath" ]; then

log "Found $wrapperJarPath"

else

log "Couldn't find $wrapperJarPath, downloading it ..."

if [ -n "$MVNW\_REPOURL" ]; then

wrapperUrl="$MVNW\_REPOURL/org/apache/maven/wrapper/maven-wrapper/3.2.0/maven-wrapper-3.2.0.jar"

else

wrapperUrl="https://repo.maven.apache.org/maven2/org/apache/maven/wrapper/maven-wrapper/3.2.0/maven-wrapper-3.2.0.jar"

fi

while IFS="=" read -r key value; do

# Remove '\r' from value to allow usage on windows as IFS does not consider '\r' as a separator ( considers space, tab, new line ('\n'), and custom '=' )

safeValue=$(echo "$value" | tr -d '\r')

case "$key" in (wrapperUrl) wrapperUrl="$safeValue"; break ;;

esac

done < "$MAVEN\_PROJECTBASEDIR/.mvn/wrapper/maven-wrapper.properties"

log "Downloading from: $wrapperUrl"

if $cygwin; then

wrapperJarPath=$(cygpath --path --windows "$wrapperJarPath")

fi

if command -v wget > /dev/null; then

log "Found wget ... using wget"

[ "$MVNW\_VERBOSE" = true ] && QUIET="" || QUIET="--quiet"

if [ -z "$MVNW\_USERNAME" ] || [ -z "$MVNW\_PASSWORD" ]; then

wget $QUIET "$wrapperUrl" -O "$wrapperJarPath" || rm -f "$wrapperJarPath"

else

wget $QUIET --http-user="$MVNW\_USERNAME" --http-password="$MVNW\_PASSWORD" "$wrapperUrl" -O "$wrapperJarPath" || rm -f "$wrapperJarPath"

fi

elif command -v curl > /dev/null; then

log "Found curl ... using curl"

[ "$MVNW\_VERBOSE" = true ] && QUIET="" || QUIET="--silent"

if [ -z "$MVNW\_USERNAME" ] || [ -z "$MVNW\_PASSWORD" ]; then

curl $QUIET -o "$wrapperJarPath" "$wrapperUrl" -f -L || rm -f "$wrapperJarPath"

else

curl $QUIET --user "$MVNW\_USERNAME:$MVNW\_PASSWORD" -o "$wrapperJarPath" "$wrapperUrl" -f -L || rm -f "$wrapperJarPath"

fi

else

log "Falling back to using Java to download"

javaSource="$MAVEN\_PROJECTBASEDIR/.mvn/wrapper/MavenWrapperDownloader.java"

javaClass="$MAVEN\_PROJECTBASEDIR/.mvn/wrapper/MavenWrapperDownloader.class"

# For Cygwin, switch paths to Windows format before running javac

if $cygwin; then

javaSource=$(cygpath --path --windows "$javaSource")

javaClass=$(cygpath --path --windows "$javaClass")

fi

if [ -e "$javaSource" ]; then

if [ ! -e "$javaClass" ]; then

log " - Compiling MavenWrapperDownloader.java ..."

("$JAVA\_HOME/bin/javac" "$javaSource")

fi

if [ -e "$javaClass" ]; then

log " - Running MavenWrapperDownloader.java ..."

("$JAVA\_HOME/bin/java" -cp .mvn/wrapper MavenWrapperDownloader "$wrapperUrl" "$wrapperJarPath") || rm -f "$wrapperJarPath"

fi

fi

fi

fi

##########################################################################################

# End of extension

##########################################################################################

# If specified, validate the SHA-256 sum of the Maven wrapper jar file

wrapperSha256Sum=""

while IFS="=" read -r key value; do

case "$key" in (wrapperSha256Sum) wrapperSha256Sum=$value; break ;;

esac

done < "$MAVEN\_PROJECTBASEDIR/.mvn/wrapper/maven-wrapper.properties"

if [ -n "$wrapperSha256Sum" ]; then

wrapperSha256Result=false

if command -v sha256sum > /dev/null; then

if echo "$wrapperSha256Sum $wrapperJarPath" | sha256sum -c > /dev/null 2>&1; then

wrapperSha256Result=true

fi

elif command -v shasum > /dev/null; then

if echo "$wrapperSha256Sum $wrapperJarPath" | shasum -a 256 -c > /dev/null 2>&1; then

wrapperSha256Result=true

fi

else

echo "Checksum validation was requested but neither 'sha256sum' or 'shasum' are available."

echo "Please install either command, or disable validation by removing 'wrapperSha256Sum' from your maven-wrapper.properties."

exit 1

fi

if [ $wrapperSha256Result = false ]; then

echo "Error: Failed to validate Maven wrapper SHA-256, your Maven wrapper might be compromised." >&2

echo "Investigate or delete $wrapperJarPath to attempt a clean download." >&2

echo "If you updated your Maven version, you need to update the specified wrapperSha256Sum property." >&2

exit 1

fi

fi

MAVEN\_OPTS="$(concat\_lines "$MAVEN\_PROJECTBASEDIR/.mvn/jvm.config") $MAVEN\_OPTS"

# For Cygwin, switch paths to Windows format before running java

if $cygwin; then

[ -n "$JAVA\_HOME" ] &&

JAVA\_HOME=$(cygpath --path --windows "$JAVA\_HOME")

[ -n "$CLASSPATH" ] &&

CLASSPATH=$(cygpath --path --windows "$CLASSPATH")

[ -n "$MAVEN\_PROJECTBASEDIR" ] &&

MAVEN\_PROJECTBASEDIR=$(cygpath --path --windows "$MAVEN\_PROJECTBASEDIR")

fi

# Provide a "standardized" way to retrieve the CLI args that will

# work with both Windows and non-Windows executions.

MAVEN\_CMD\_LINE\_ARGS="$MAVEN\_CONFIG $\*"

export MAVEN\_CMD\_LINE\_ARGS

WRAPPER\_LAUNCHER=org.apache.maven.wrapper.MavenWrapperMain

# shellcheck disable=SC2086 # safe args

exec "$JAVACMD" \

$MAVEN\_OPTS \

$MAVEN\_DEBUG\_OPTS \

-classpath "$MAVEN\_PROJECTBASEDIR/.mvn/wrapper/maven-wrapper.jar" \

"-Dmaven.multiModuleProjectDirectory=${MAVEN\_PROJECTBASEDIR}" \

${WRAPPER\_LAUNCHER} $MAVEN\_CONFIG "$@"

***mvnw.cmd***

@REM ----------------------------------------------------------------------------

@REM Licensed to the Apache Software Foundation (ASF) under one

@REM or more contributor license agreements. See the NOTICE file

@REM distributed with this work for additional information

@REM regarding copyright ownership. The ASF licenses this file

@REM to you under the Apache License, Version 2.0 (the

@REM "License"); you may not use this file except in compliance

@REM with the License. You may obtain a copy of the License at

@REM

@REM https://www.apache.org/licenses/LICENSE-2.0

@REM

@REM Unless required by applicable law or agreed to in writing,

@REM software distributed under the License is distributed on an

@REM "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY

@REM KIND, either express or implied. See the License for the

@REM specific language governing permissions and limitations

@REM under the License.

@REM ----------------------------------------------------------------------------

@REM ----------------------------------------------------------------------------

@REM Apache Maven Wrapper startup batch script, version 3.2.0

@REM

@REM Required ENV vars:

@REM JAVA\_HOME - location of a JDK home dir

@REM

@REM Optional ENV vars

@REM MAVEN\_BATCH\_ECHO - set to 'on' to enable the echoing of the batch commands

@REM MAVEN\_BATCH\_PAUSE - set to 'on' to wait for a keystroke before ending

@REM MAVEN\_OPTS - parameters passed to the Java VM when running Maven

@REM e.g. to debug Maven itself, use

@REM set MAVEN\_OPTS=-Xdebug -Xrunjdwp:transport=dt\_socket,server=y,suspend=y,address=8000

@REM MAVEN\_SKIP\_RC - flag to disable loading of mavenrc files

@REM ----------------------------------------------------------------------------

@REM Begin all REM lines with '@' in case MAVEN\_BATCH\_ECHO is 'on'

@echo off

@REM set title of command window

title %0

@REM enable echoing by setting MAVEN\_BATCH\_ECHO to 'on'

@if "%MAVEN\_BATCH\_ECHO%" == "on" echo %MAVEN\_BATCH\_ECHO%

@REM set %HOME% to equivalent of $HOME

if "%HOME%" == "" (set "HOME=%HOMEDRIVE%%HOMEPATH%")

@REM Execute a user defined script before this one

if not "%MAVEN\_SKIP\_RC%" == "" goto skipRcPre

@REM check for pre script, once with legacy .bat ending and once with .cmd ending

if exist "%USERPROFILE%\mavenrc\_pre.bat" call "%USERPROFILE%\mavenrc\_pre.bat" %\*

if exist "%USERPROFILE%\mavenrc\_pre.cmd" call "%USERPROFILE%\mavenrc\_pre.cmd" %\*

:skipRcPre

@setlocal

set ERROR\_CODE=0

@REM To isolate internal variables from possible post scripts, we use another setlocal

@setlocal

@REM ==== START VALIDATION ====

if not "%JAVA\_HOME%" == "" goto OkJHome

echo.

echo Error: JAVA\_HOME not found in your environment. >&2

echo Please set the JAVA\_HOME variable in your environment to match the >&2

echo location of your Java installation. >&2

echo.

goto error

:OkJHome

if exist "%JAVA\_HOME%\bin\java.exe" goto init

echo.

echo Error: JAVA\_HOME is set to an invalid directory. >&2

echo JAVA\_HOME = "%JAVA\_HOME%" >&2

echo Please set the JAVA\_HOME variable in your environment to match the >&2

echo location of your Java installation. >&2

echo.

goto error

@REM ==== END VALIDATION ====

:init

@REM Find the project base dir, i.e. the directory that contains the folder ".mvn".

@REM Fallback to current working directory if not found.

set MAVEN\_PROJECTBASEDIR=%MAVEN\_BASEDIR%

IF NOT "%MAVEN\_PROJECTBASEDIR%"=="" goto endDetectBaseDir

set EXEC\_DIR=%CD%

set WDIR=%EXEC\_DIR%

:findBaseDir

IF EXIST "%WDIR%"\.mvn goto baseDirFound

cd ..

IF "%WDIR%"=="%CD%" goto baseDirNotFound

set WDIR=%CD%

goto findBaseDir

:baseDirFound

set MAVEN\_PROJECTBASEDIR=%WDIR%

cd "%EXEC\_DIR%"

goto endDetectBaseDir

:baseDirNotFound

set MAVEN\_PROJECTBASEDIR=%EXEC\_DIR%

cd "%EXEC\_DIR%"

:endDetectBaseDir

IF NOT EXIST "%MAVEN\_PROJECTBASEDIR%\.mvn\jvm.config" goto endReadAdditionalConfig

@setlocal EnableExtensions EnableDelayedExpansion

for /F "usebackq delims=" %%a in ("%MAVEN\_PROJECTBASEDIR%\.mvn\jvm.config") do set JVM\_CONFIG\_MAVEN\_PROPS=!JVM\_CONFIG\_MAVEN\_PROPS! %%a

@endlocal & set JVM\_CONFIG\_MAVEN\_PROPS=%JVM\_CONFIG\_MAVEN\_PROPS%

:endReadAdditionalConfig

SET MAVEN\_JAVA\_EXE="%JAVA\_HOME%\bin\java.exe"

set WRAPPER\_JAR="%MAVEN\_PROJECTBASEDIR%\.mvn\wrapper\maven-wrapper.jar"

set WRAPPER\_LAUNCHER=org.apache.maven.wrapper.MavenWrapperMain

set WRAPPER\_URL="https://repo.maven.apache.org/maven2/org/apache/maven/wrapper/maven-wrapper/3.2.0/maven-wrapper-3.2.0.jar"

FOR /F "usebackq tokens=1,2 delims==" %%A IN ("%MAVEN\_PROJECTBASEDIR%\.mvn\wrapper\maven-wrapper.properties") DO (

IF "%%A"=="wrapperUrl" SET WRAPPER\_URL=%%B

)

@REM Extension to allow automatically downloading the maven-wrapper.jar from Maven-central

@REM This allows using the maven wrapper in projects that prohibit checking in binary data.

if exist %WRAPPER\_JAR% (

if "%MVNW\_VERBOSE%" == "true" (

echo Found %WRAPPER\_JAR%

)

) else (

if not "%MVNW\_REPOURL%" == "" (

SET WRAPPER\_URL="%MVNW\_REPOURL%/org/apache/maven/wrapper/maven-wrapper/3.2.0/maven-wrapper-3.2.0.jar"

)

if "%MVNW\_VERBOSE%" == "true" (

echo Couldn't find %WRAPPER\_JAR%, downloading it ...

echo Downloading from: %WRAPPER\_URL%

)

powershell -Command "&{"^

"$webclient = new-object System.Net.WebClient;"^

"if (-not ([string]::IsNullOrEmpty('%MVNW\_USERNAME%') -and [string]::IsNullOrEmpty('%MVNW\_PASSWORD%'))) {"^

"$webclient.Credentials = new-object System.Net.NetworkCredential('%MVNW\_USERNAME%', '%MVNW\_PASSWORD%');"^

"}"^

"[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12; $webclient.DownloadFile('%WRAPPER\_URL%', '%WRAPPER\_JAR%')"^

"}"

if "%MVNW\_VERBOSE%" == "true" (

echo Finished downloading %WRAPPER\_JAR%

)

)

@REM End of extension

@REM If specified, validate the SHA-256 sum of the Maven wrapper jar file

SET WRAPPER\_SHA\_256\_SUM=""

FOR /F "usebackq tokens=1,2 delims==" %%A IN ("%MAVEN\_PROJECTBASEDIR%\.mvn\wrapper\maven-wrapper.properties") DO (

IF "%%A"=="wrapperSha256Sum" SET WRAPPER\_SHA\_256\_SUM=%%B

)

IF NOT %WRAPPER\_SHA\_256\_SUM%=="" (

powershell -Command "&{"^

"$hash = (Get-FileHash \"%WRAPPER\_JAR%\" -Algorithm SHA256).Hash.ToLower();"^

"If('%WRAPPER\_SHA\_256\_SUM%' -ne $hash){"^

" Write-Output 'Error: Failed to validate Maven wrapper SHA-256, your Maven wrapper might be compromised.';"^

" Write-Output 'Investigate or delete %WRAPPER\_JAR% to attempt a clean download.';"^

" Write-Output 'If you updated your Maven version, you need to update the specified wrapperSha256Sum property.';"^

" exit 1;"^

"}"^

"}"

if ERRORLEVEL 1 goto error

)

@REM Provide a "standardized" way to retrieve the CLI args that will

@REM work with both Windows and non-Windows executions.

set MAVEN\_CMD\_LINE\_ARGS=%\*

%MAVEN\_JAVA\_EXE% ^

%JVM\_CONFIG\_MAVEN\_PROPS% ^

%MAVEN\_OPTS% ^

%MAVEN\_DEBUG\_OPTS% ^

-classpath %WRAPPER\_JAR% ^

"-Dmaven.multiModuleProjectDirectory=%MAVEN\_PROJECTBASEDIR%" ^

%WRAPPER\_LAUNCHER% %MAVEN\_CONFIG% %\*

if ERRORLEVEL 1 goto error

goto end

:error

set ERROR\_CODE=1

:end

@endlocal & set ERROR\_CODE=%ERROR\_CODE%

if not "%MAVEN\_SKIP\_RC%"=="" goto skipRcPost

@REM check for post script, once with legacy .bat ending and once with .cmd ending

if exist "%USERPROFILE%\mavenrc\_post.bat" call "%USERPROFILE%\mavenrc\_post.bat"

if exist "%USERPROFILE%\mavenrc\_post.cmd" call "%USERPROFILE%\mavenrc\_post.cmd"

:skipRcPost

@REM pause the script if MAVEN\_BATCH\_PAUSE is set to 'on'

if "%MAVEN\_BATCH\_PAUSE%"=="on" pause

if "%MAVEN\_TERMINATE\_CMD%"=="on" exit %ERROR\_CODE%

cmd /C exit /B %ERROR\_CODE%

***pom.xml***

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>3.1.3</version>

<relativePath/> <!-- lookup parent from repository -->

</parent>

<groupId>com.example</groupId>

<artifactId>springboot-aws-deploy</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>springboot-aws-deploy</name>

<description>Demo project for Spring Boot</description>

<properties>

<java.version>17</java.version>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-actuator</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

<finalName>springboot-aws-deploy-service</finalName>

</build>

</project>