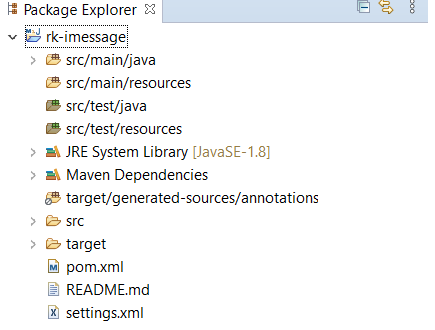
**Process – 1: Manually publishing jar file to JFrog artifactory using Maven commands**

* Code Base: Required files – Project Structure

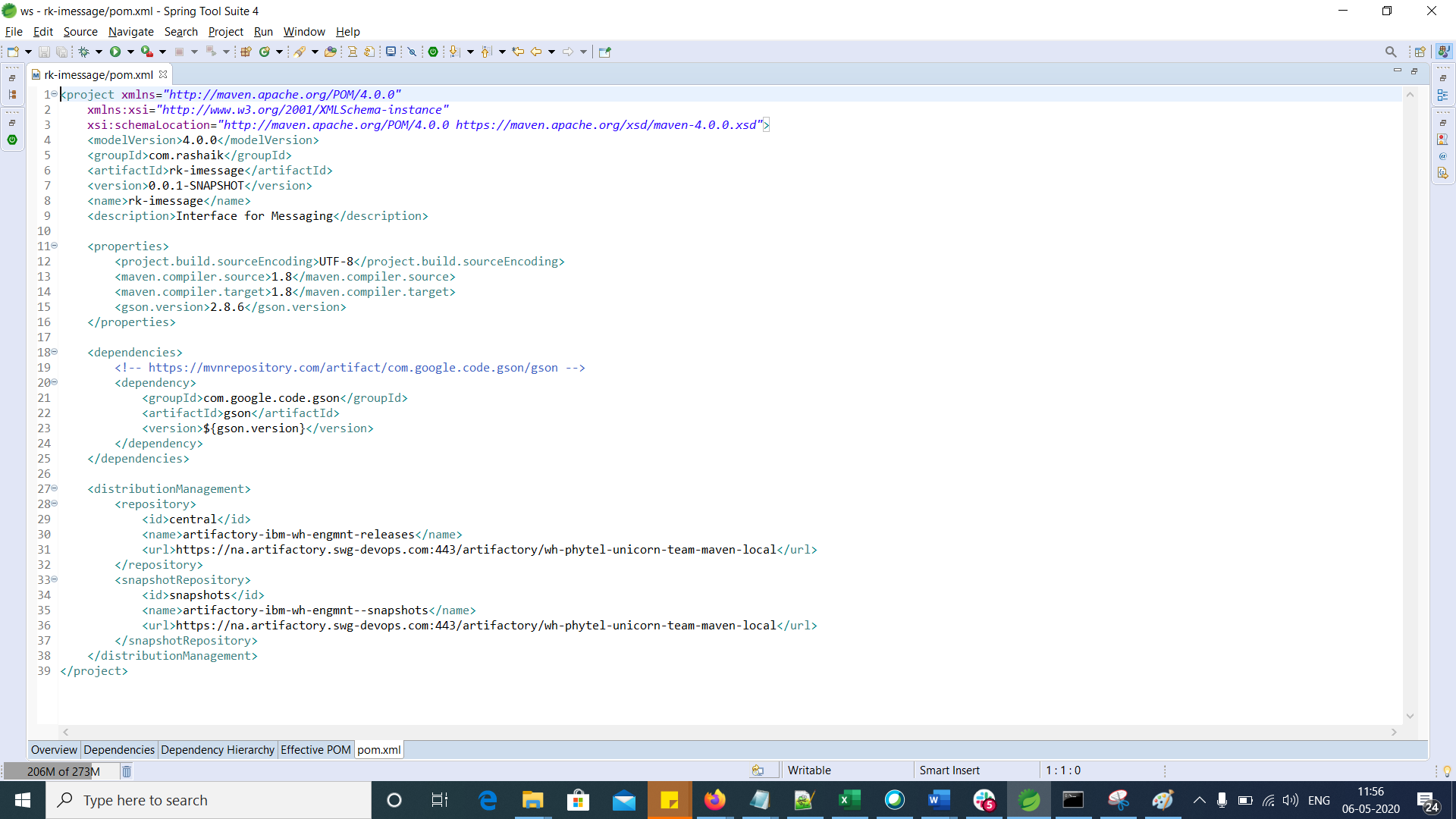


* Execute commands – to deploy to JFrog

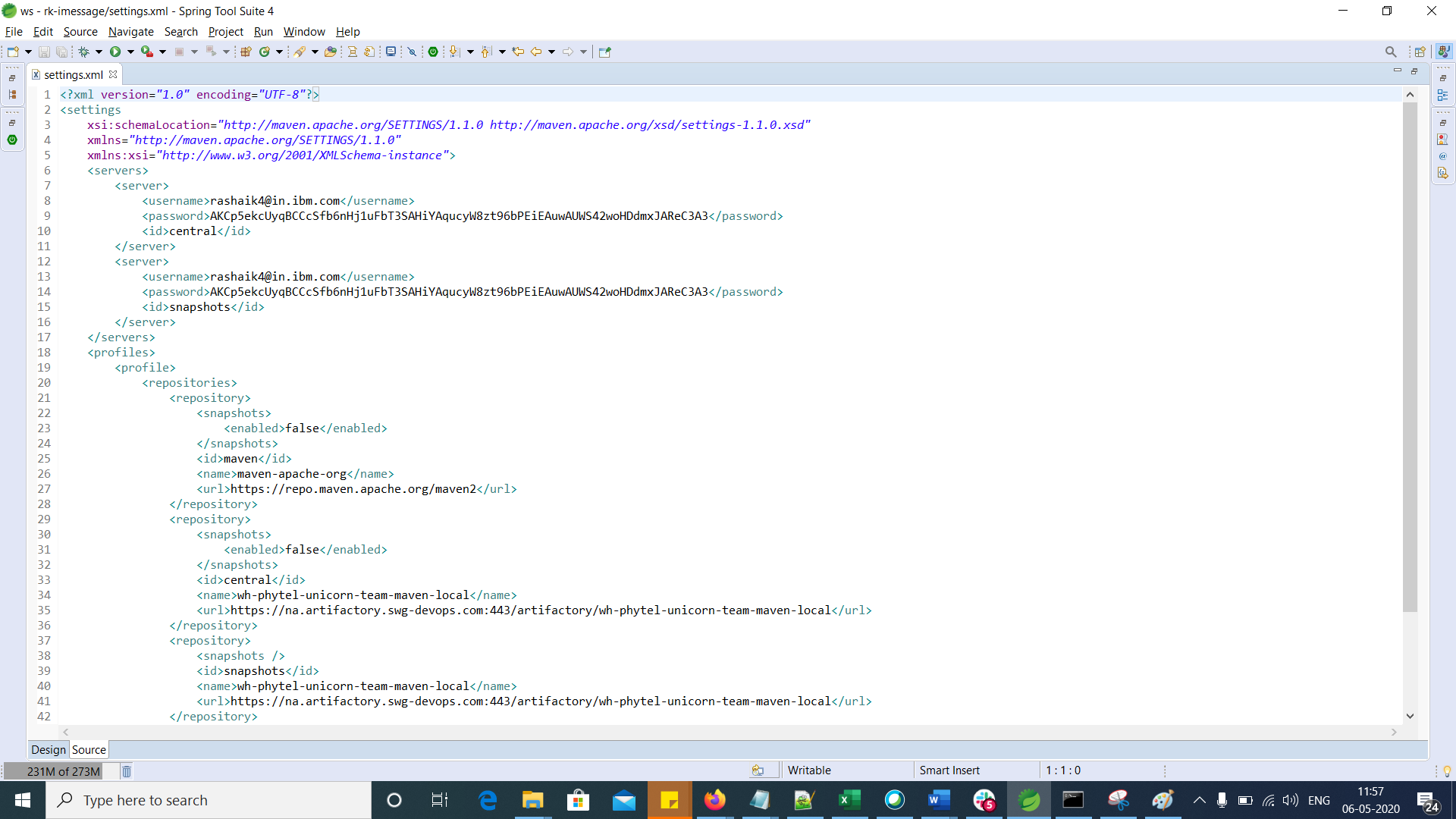
mvn clean deploy --settings settings.xml [OR]

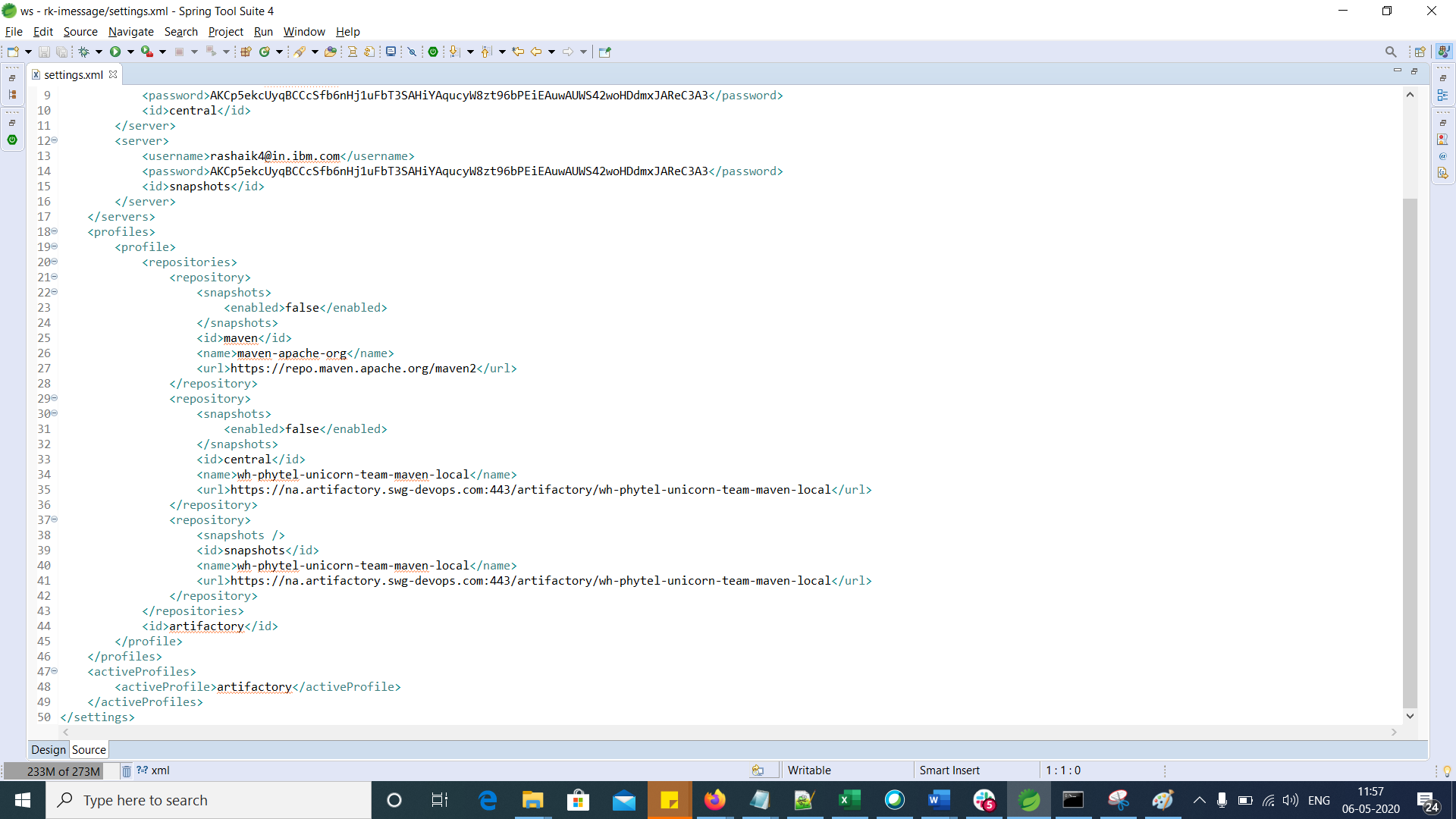
mvn deploy -Dusername=rashaik4@in.ibm.com -Dpassword=AKCp5ekcUyqBCCcSfb6nHj1uFbT3SAHiYAqucyW8zt96bPEiEAuwAUWS42woHDdmxJAReC3A3

* Pom.xml code – distributionmanagement is important

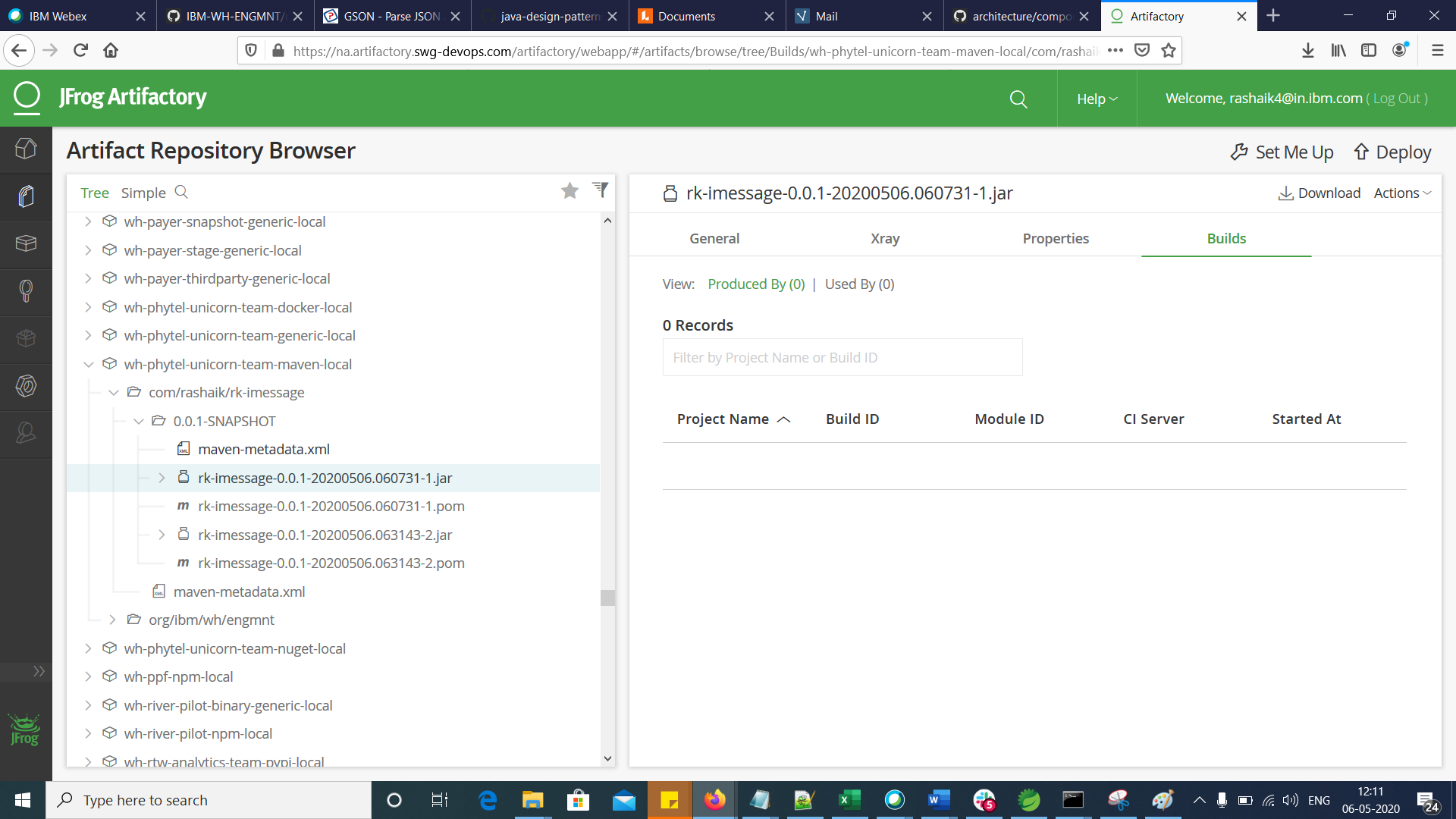


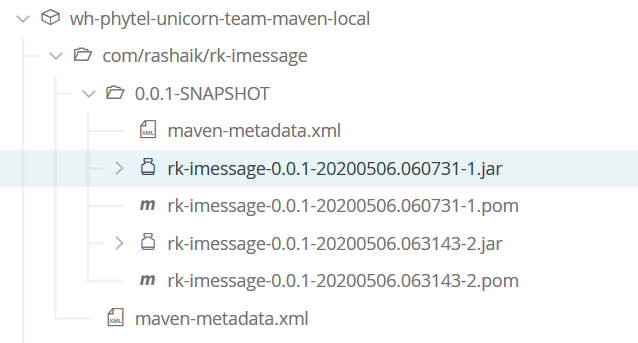
* Settings.xml code – Jfrog credentials



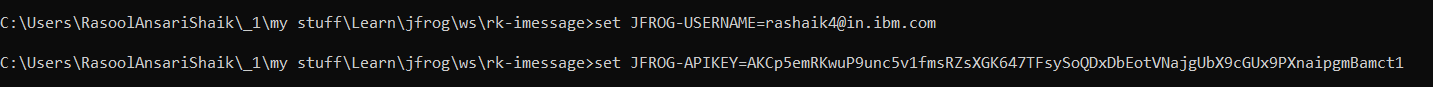


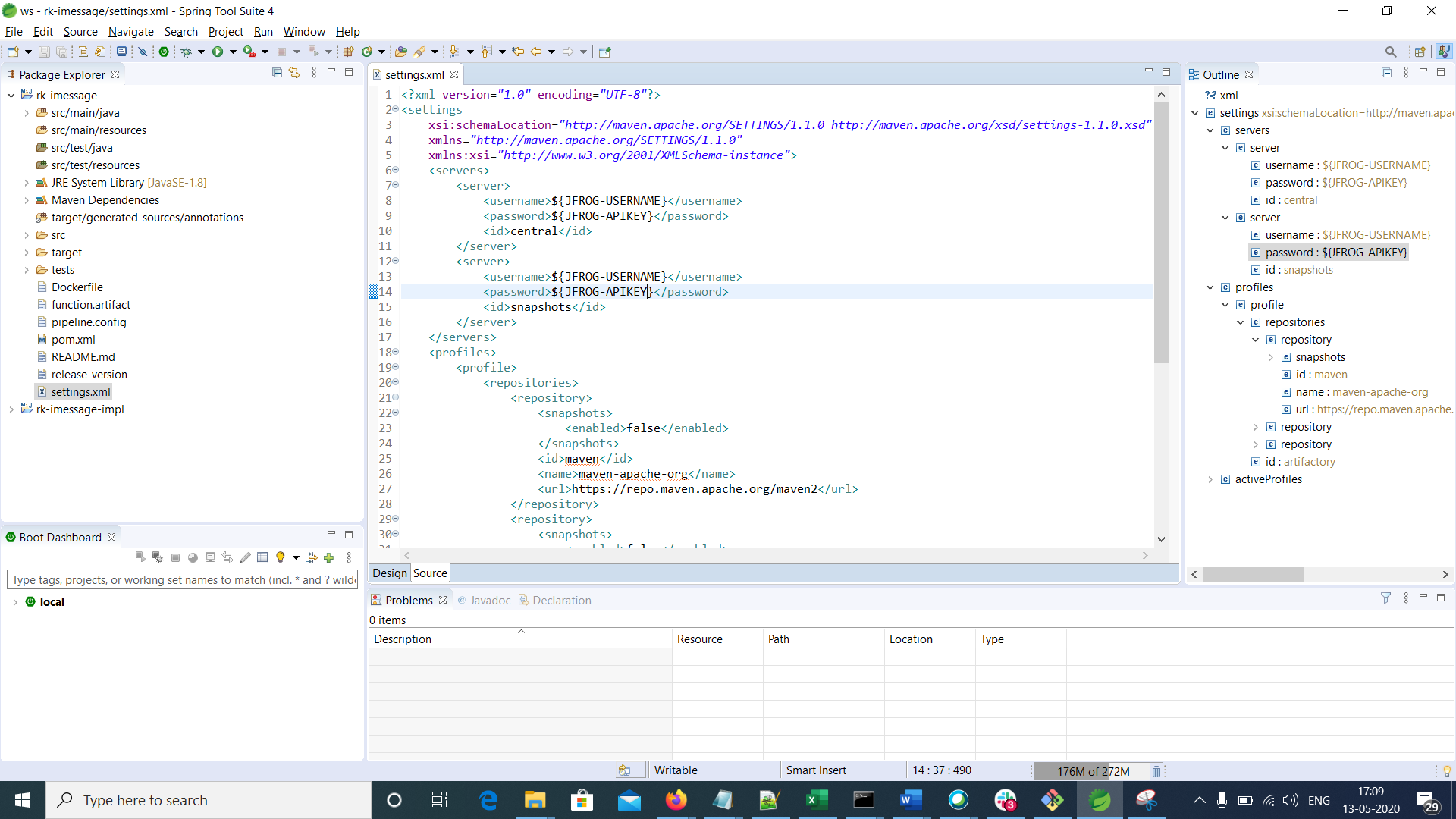
* JFrog url - <https://na.artifactory.swg-devops.com/artifactory>
* Artefactory - wh-phytel-unicorn-team-maven-local





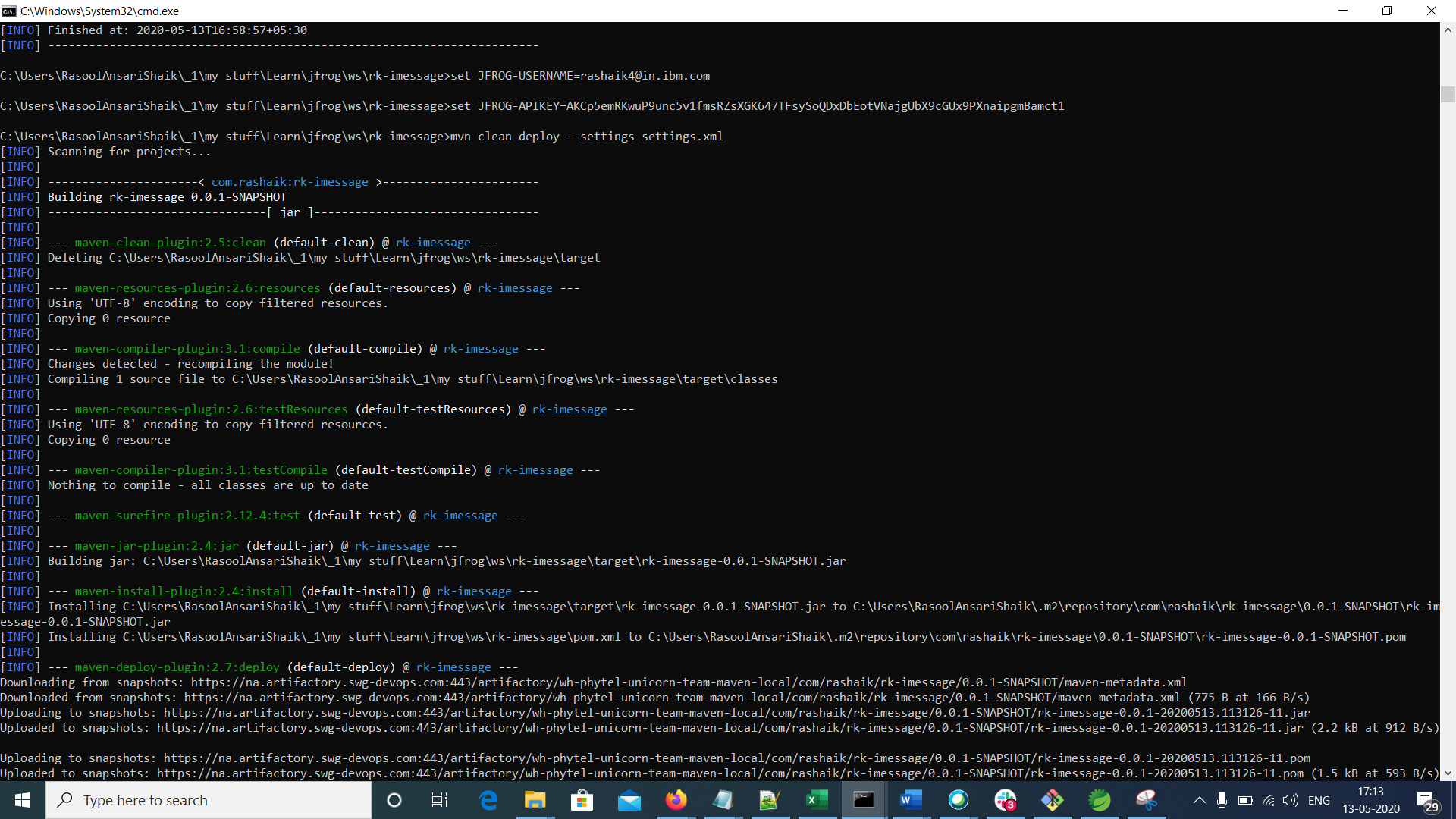
* There is another way to use settings.xml instead of username and password. Set the username and password as environment variables and refer these environment variables in the settings.xml.

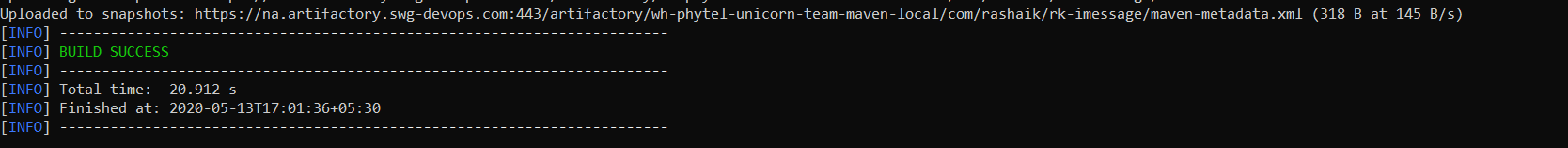




Now you can run the same above command to publish the artefact to jFrog

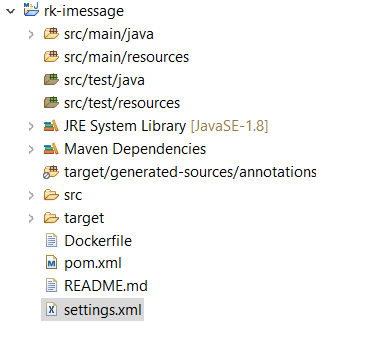
mvn clean deploy --settings settings.xml





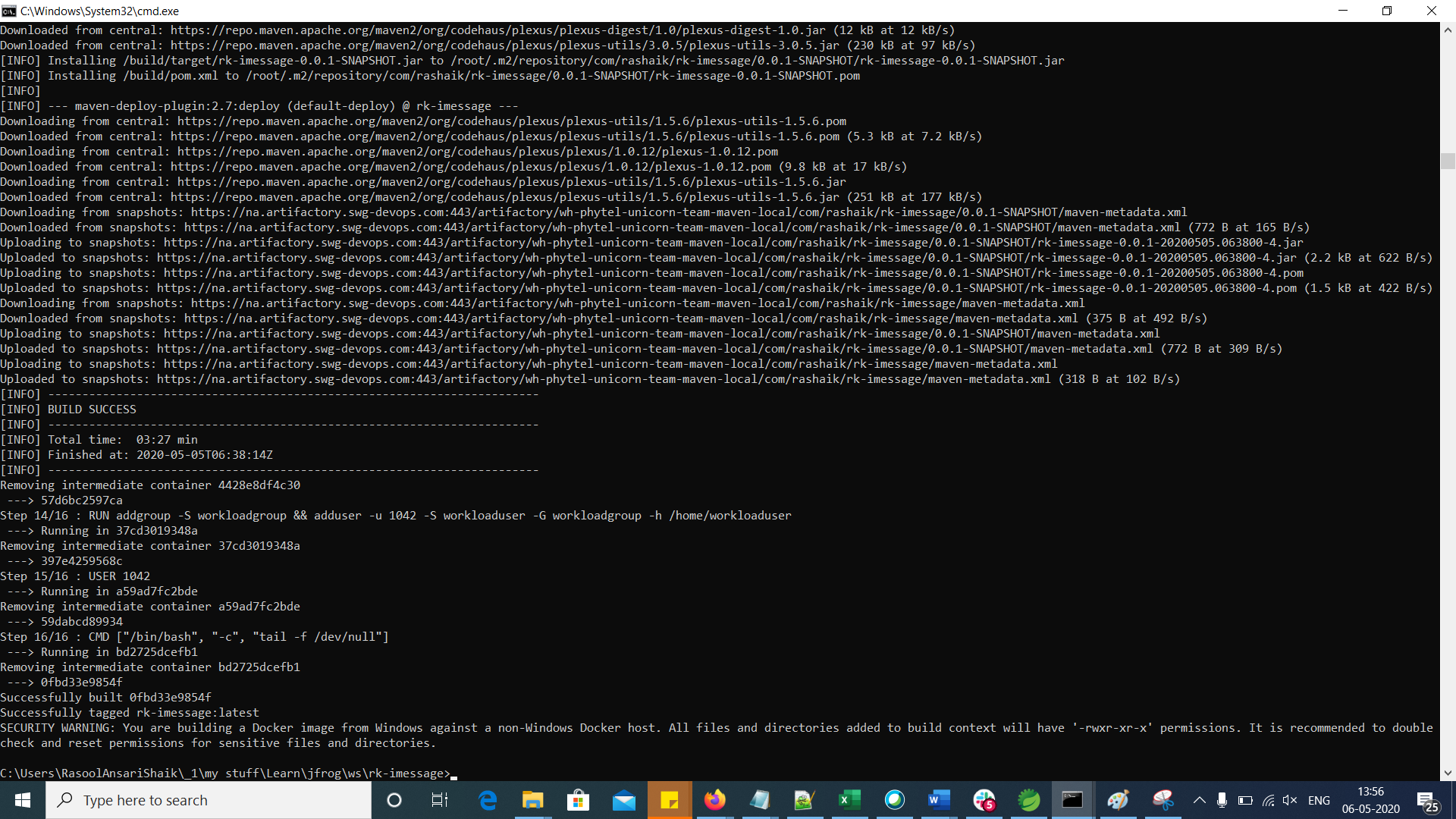
**Process – 2: Manually publishing jar file to JFrog antifactory using Docker image**

* Code Base: Required files – Project Structure

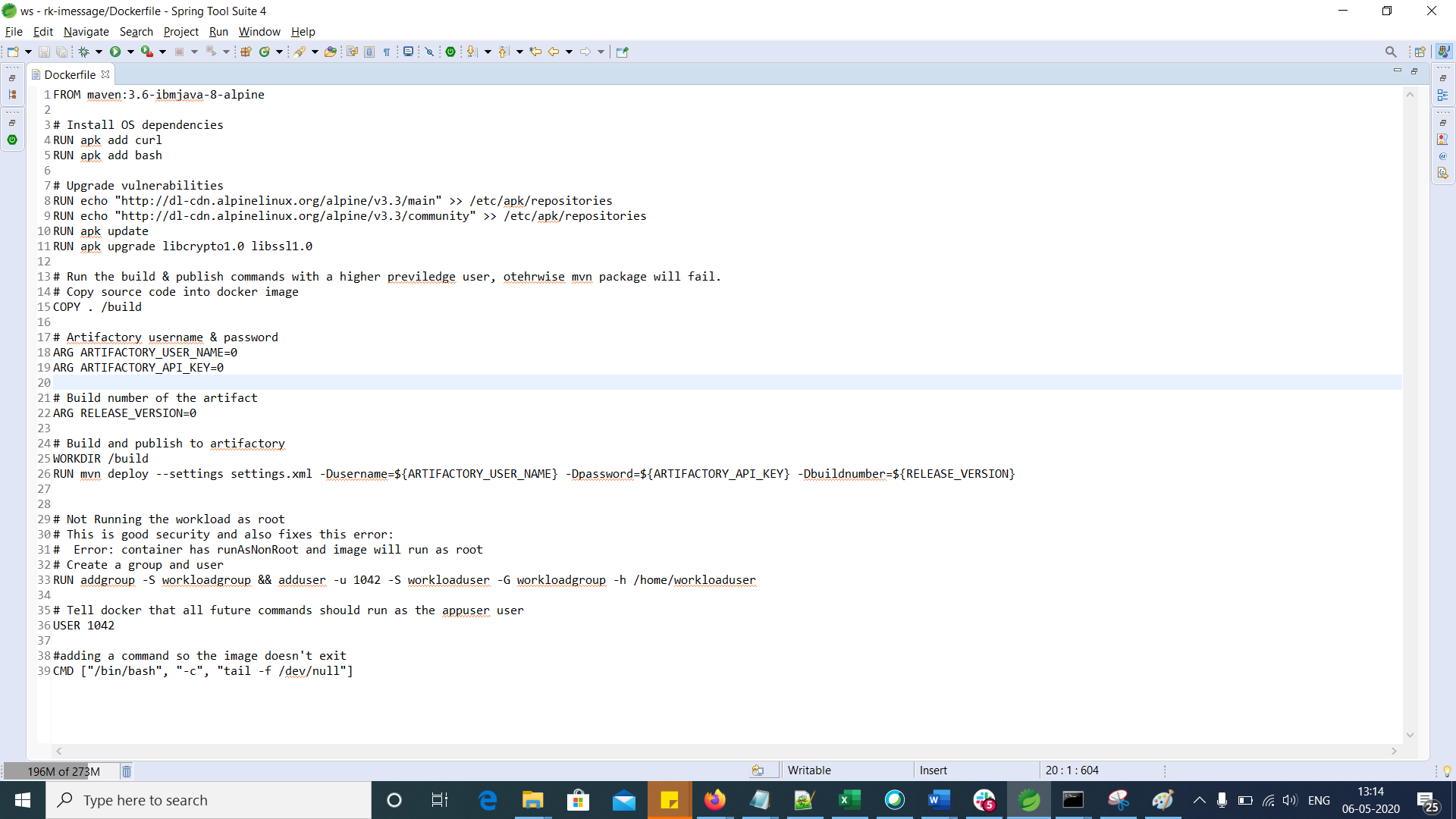
****

* Execute commands – to deploy to JFrog

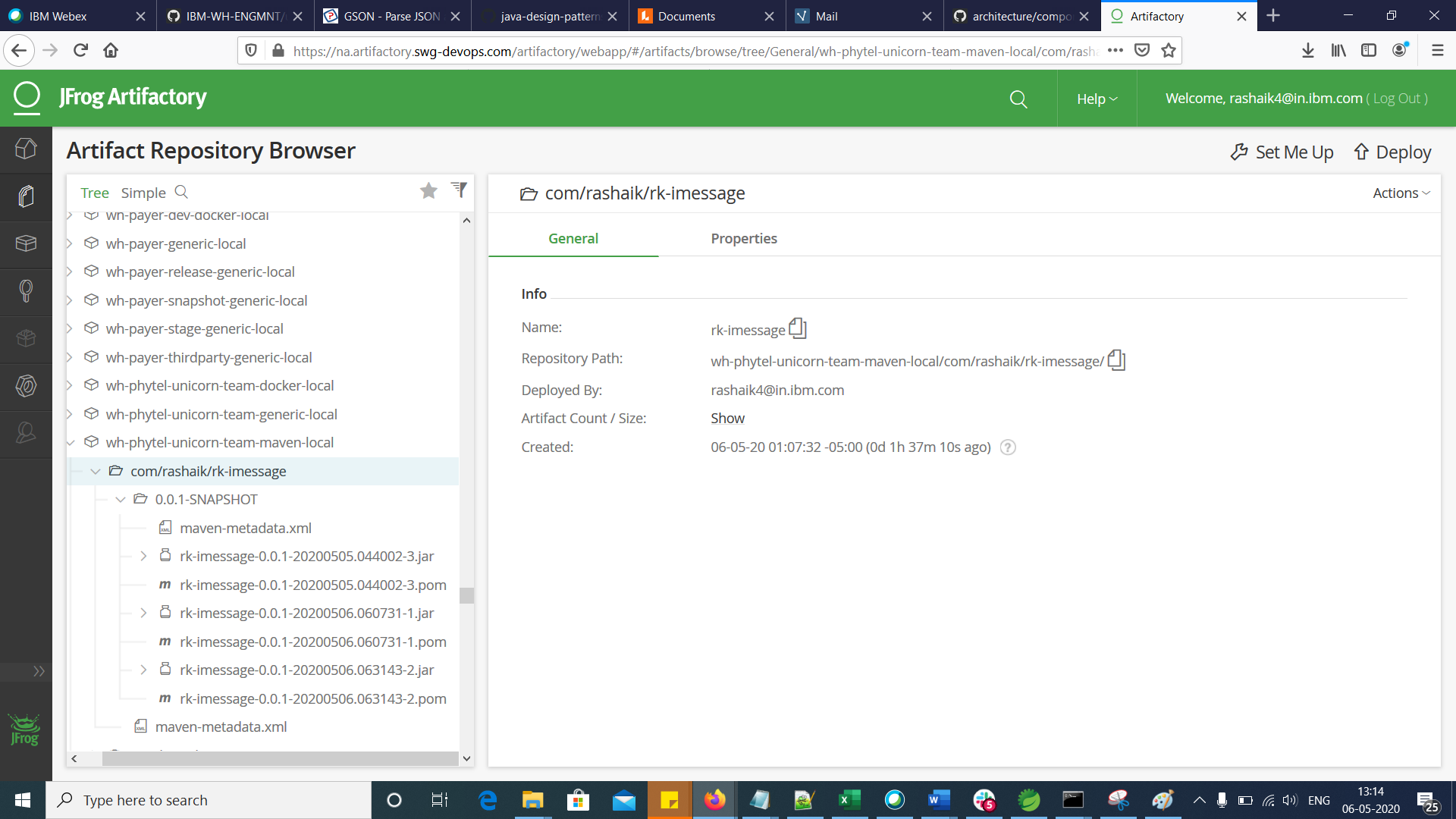
docker build -t rk-imessage:latest .

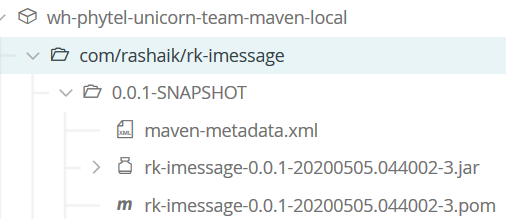


* Dockerfile



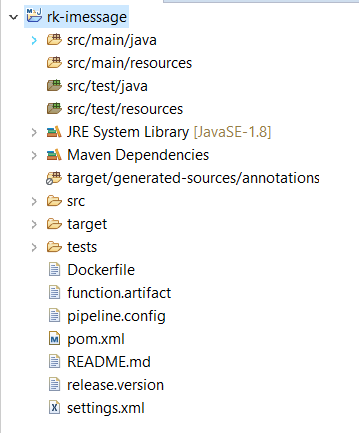
* JFrog url - <https://na.artifactory.swg-devops.com/artifactory>
* Artefactory - wh-phytel-unicorn-team-maven-local



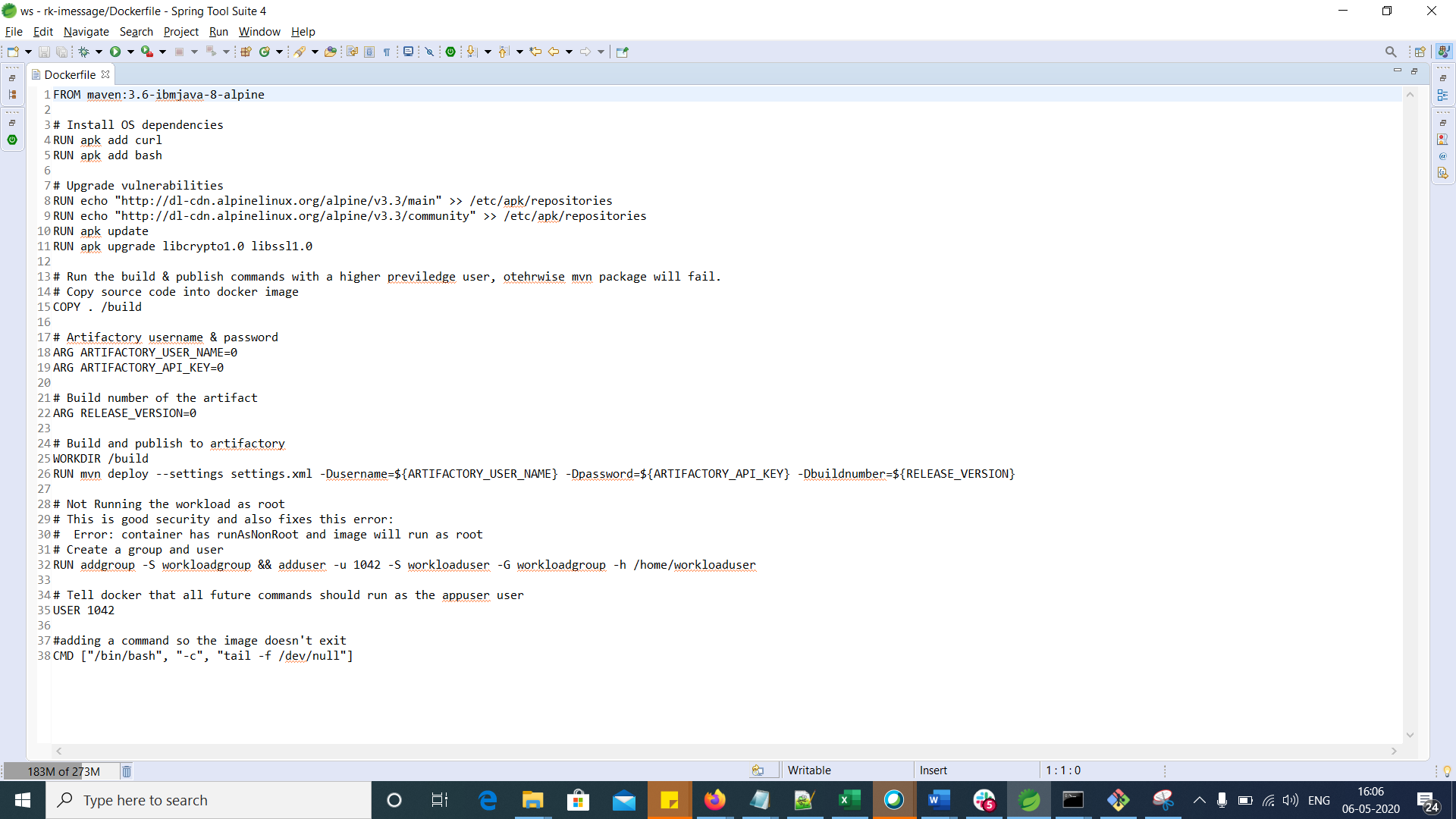


**Process – 3: Automatically publishing jar file to JFrog antifactory using CICD pipeline**

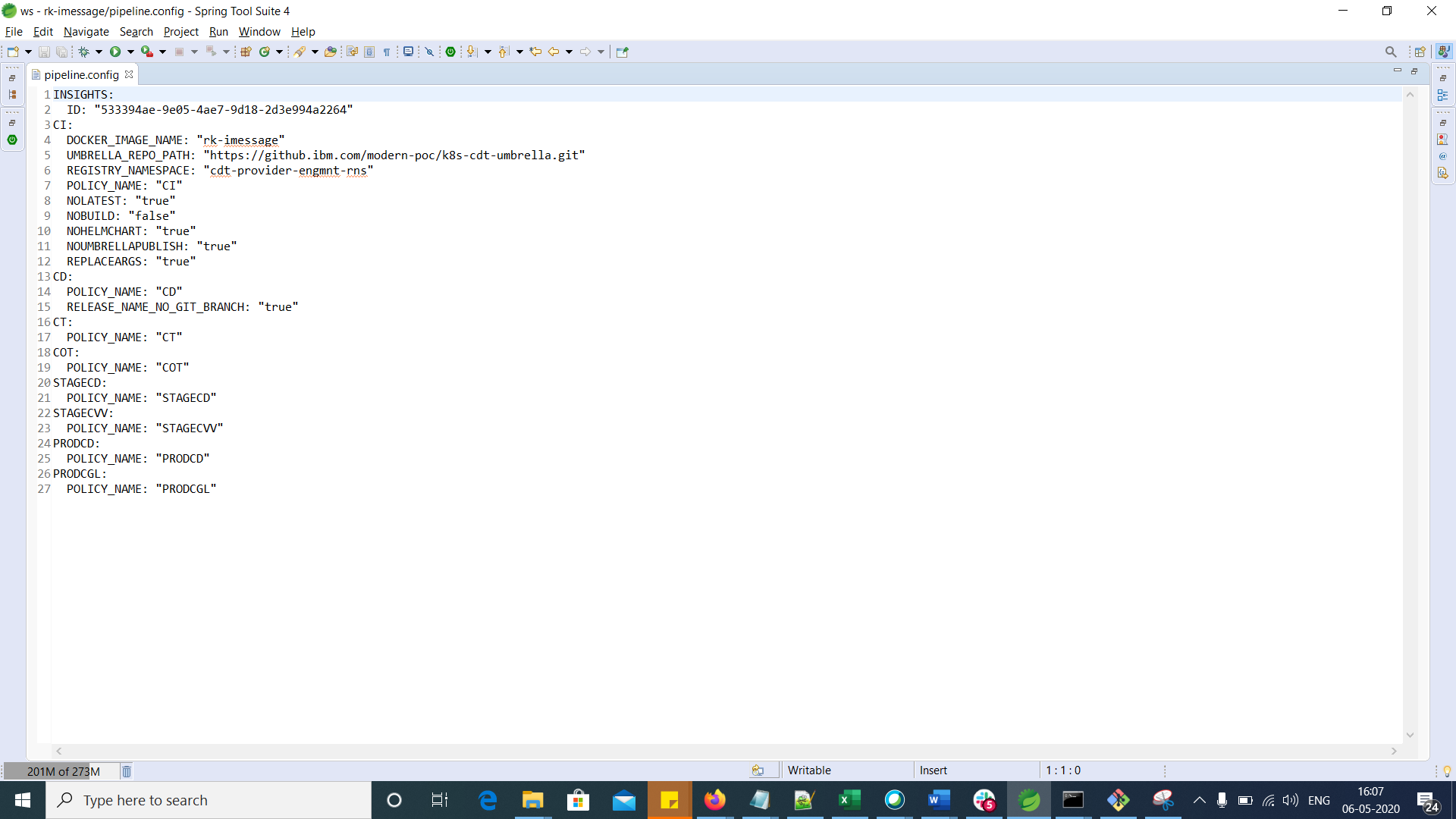
* Code Base: Required files – Project Structure



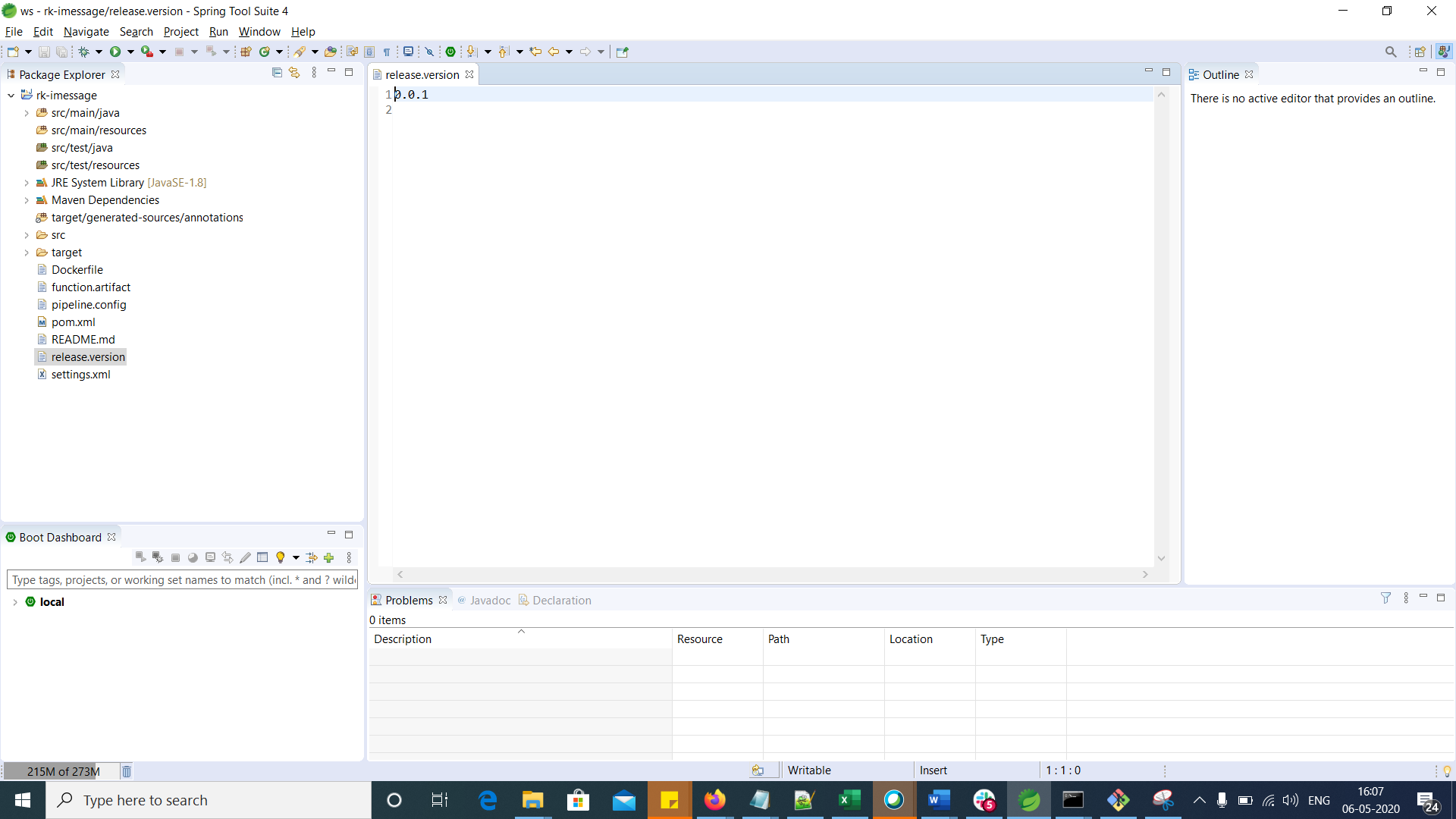
* File – Dockerfile



* File – pipeline.config

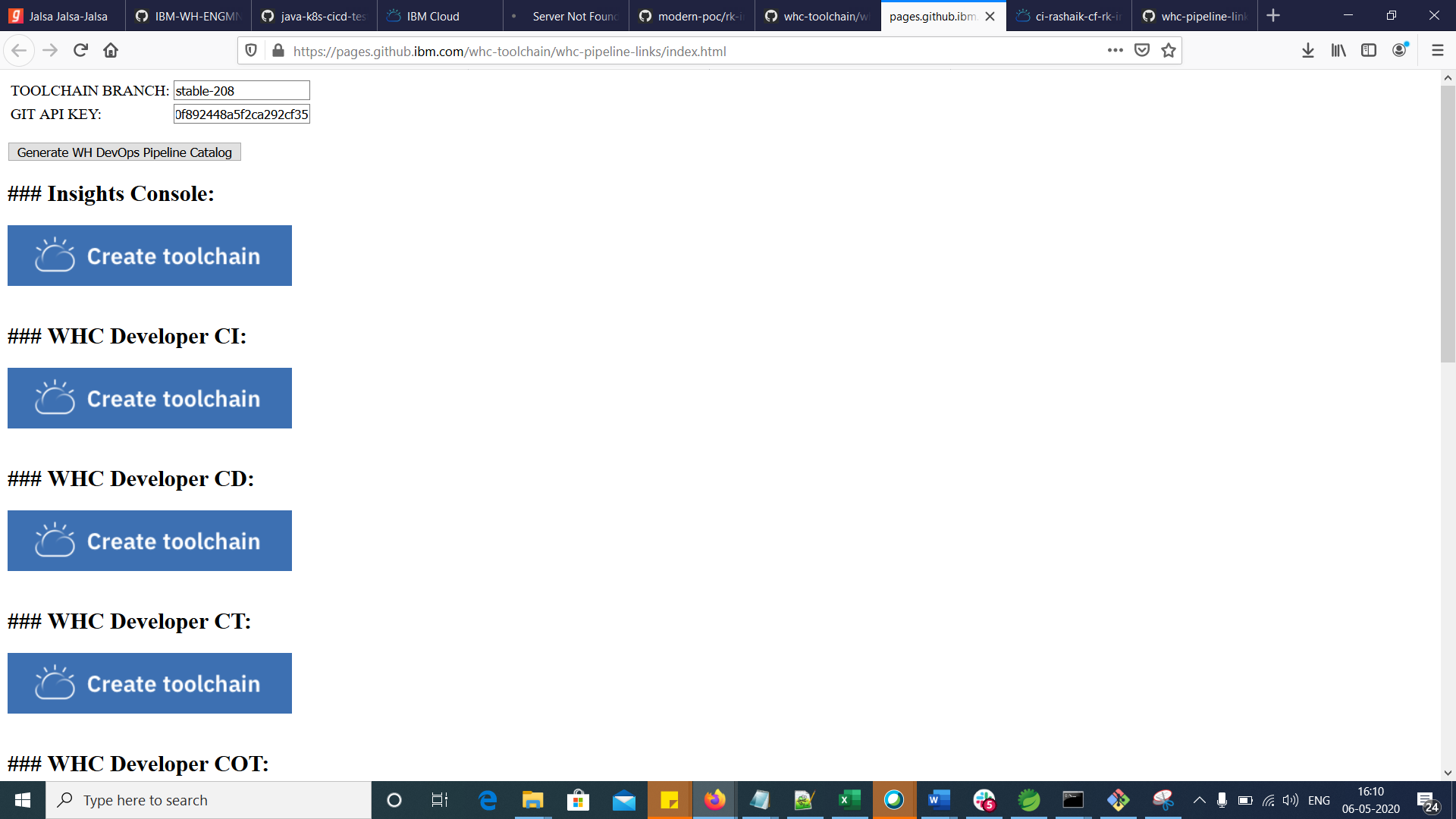


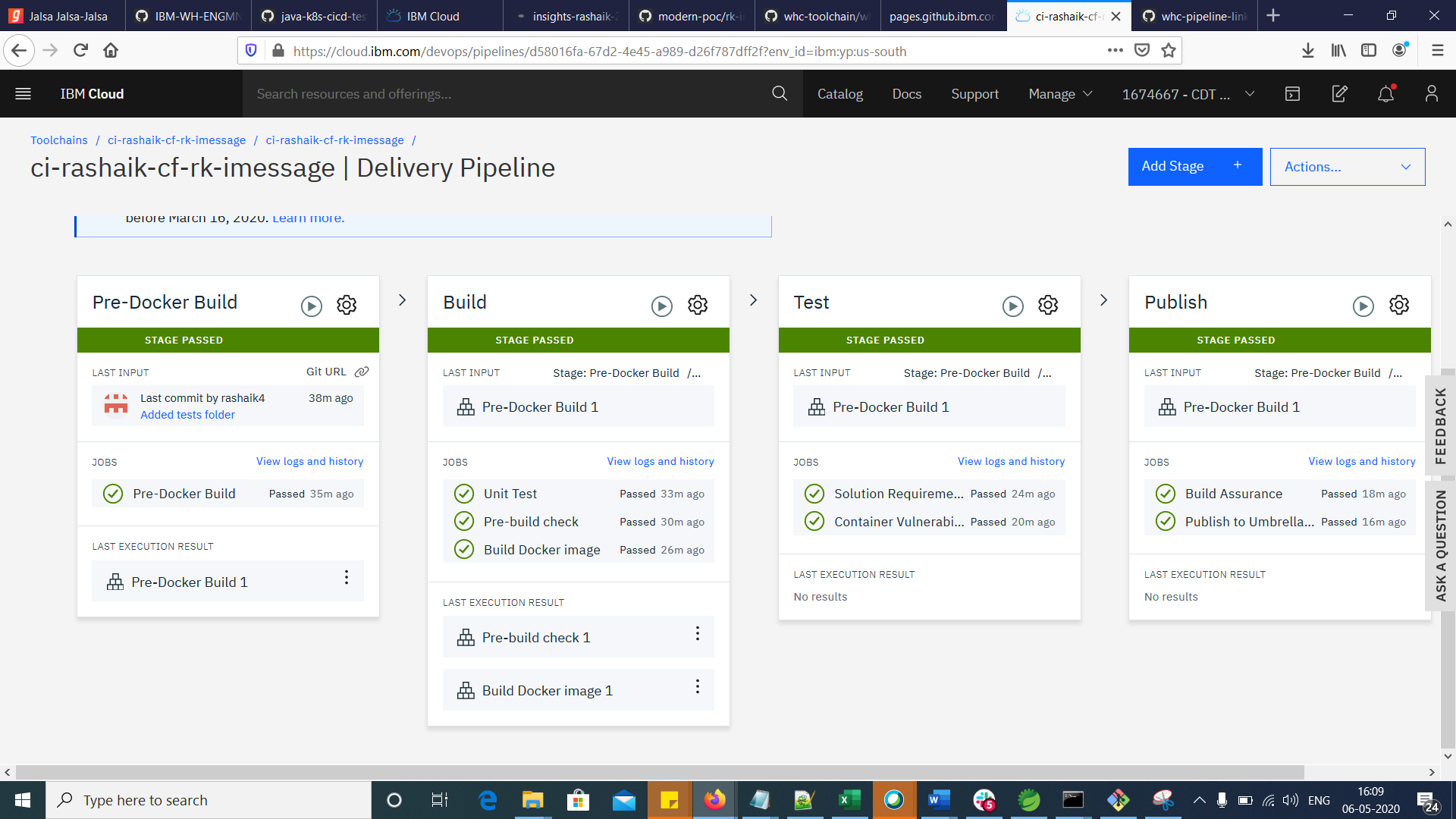
* File – release-version



* Create pipeline for CI toolchain

Reference: <https://github.ibm.com/whc-toolchain/whc-pipeline-links>





* JFrog url - <https://na.artifactory.swg-devops.com/artifactory>
* Artefactory - wh-phytel-unicorn-team-maven-local

