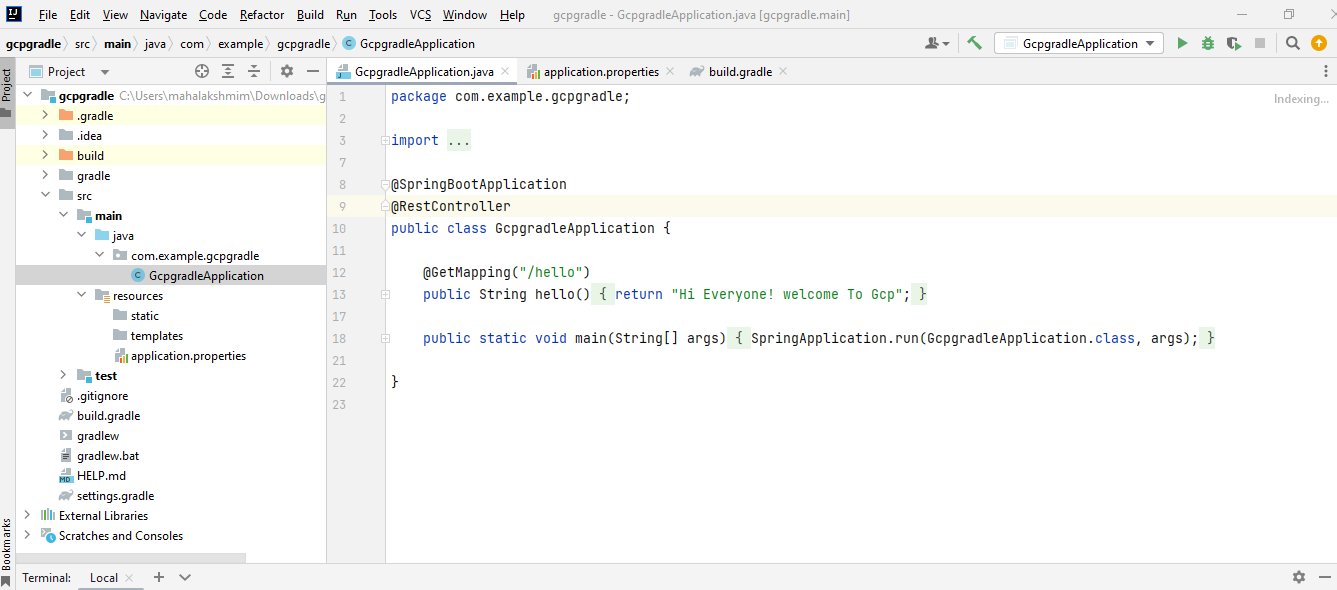
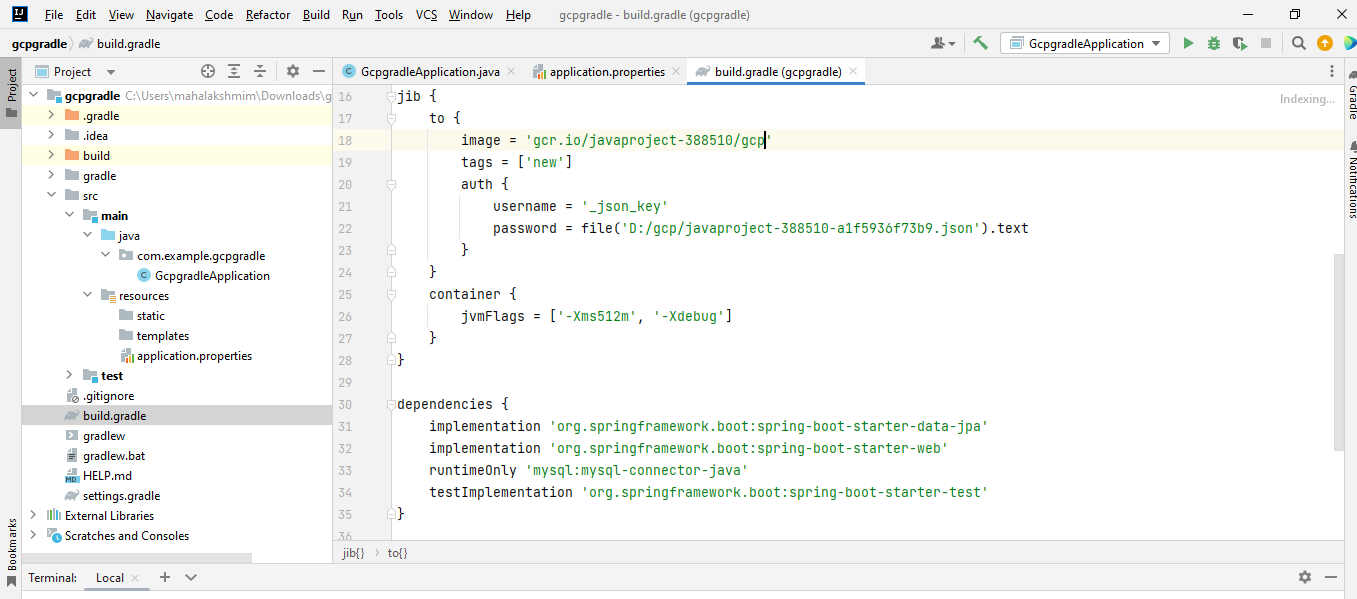
**How to work with Gcp with step by step in detail nad how to push the spring boot application a s a jib into container registry in gcp**

**Step1: Create one sample Application using in springboot in Gradle**



**Step2:In that Application need to add the JIB Configuration in build.gradle file**



**Configurations:**

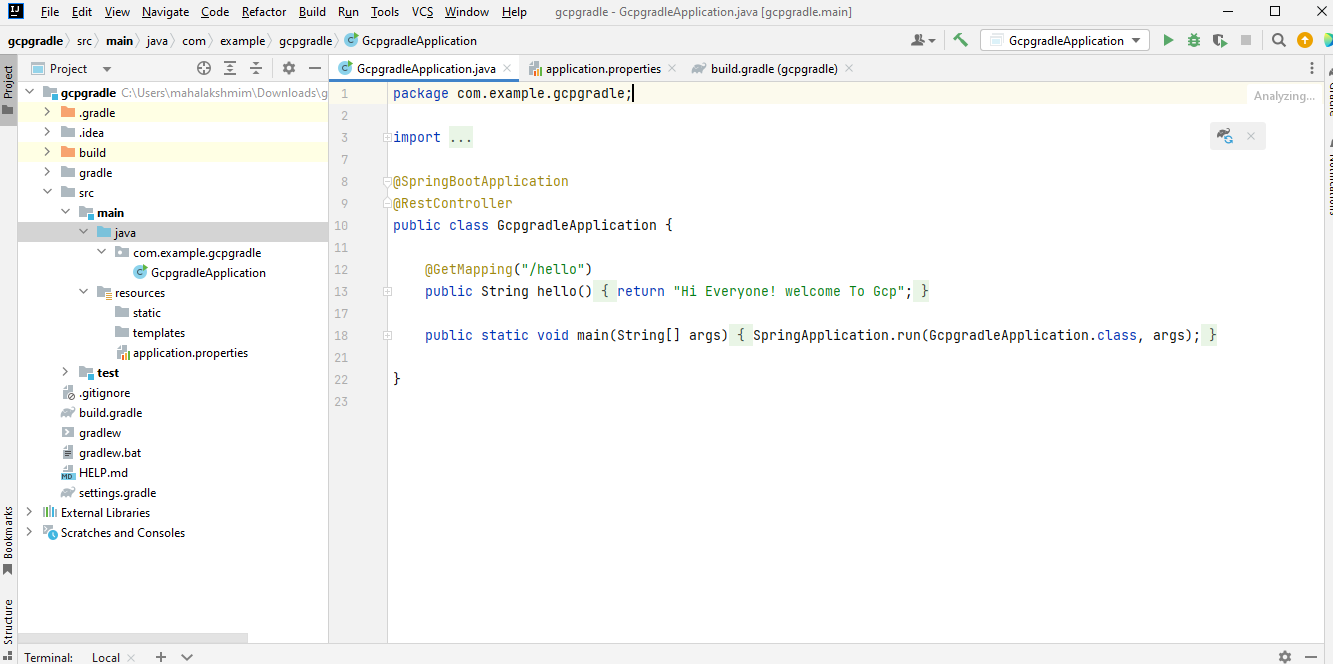
plugins **{** id 'java'  
 id 'org.springframework.boot' version '2.5.0'  
 id 'io.spring.dependency-management' version '1.1.0'  
 id 'com.google.cloud.tools.jib' version '3.1.0'  
**}**group = 'com.example'  
version = '0.0.1-SNAPSHOT'  
sourceCompatibility = '1.8'  
  
repositories **{** mavenCentral()  
**}**jib **{** to **{** image = 'gcr.io/javaproject-388510/gcp'  
 tags = ['new']  
 auth **{** username = '\_json\_key'  
 password = file('D:/gcp/javaproject-388510-a1f5936f73b9.json').text  
 **}  
 }** container **{** jvmFlags = ['-Xms512m', '-Xdebug']  
 **}  
}**dependencies **{** implementation 'org.springframework.boot:spring-boot-starter-data-jpa'  
 implementation 'org.springframework.boot:spring-boot-starter-web'  
 runtimeOnly 'mysql:mysql-connector-java'  
 testImplementation 'org.springframework.boot:spring-boot-starter-test'  
**}**tasks.named('test') **{** useJUnitPlatform()  
**}**

**Step3:After adding this jib in build .gradle need to use this command in terminal**

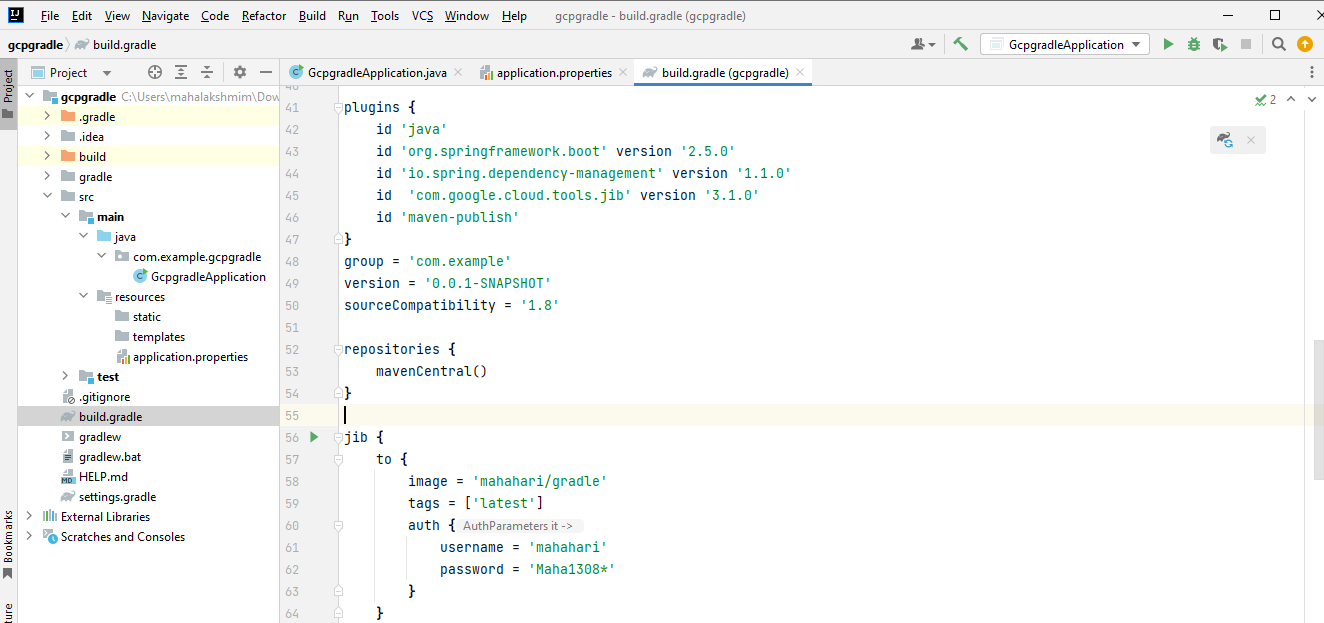
**Gradle jib**

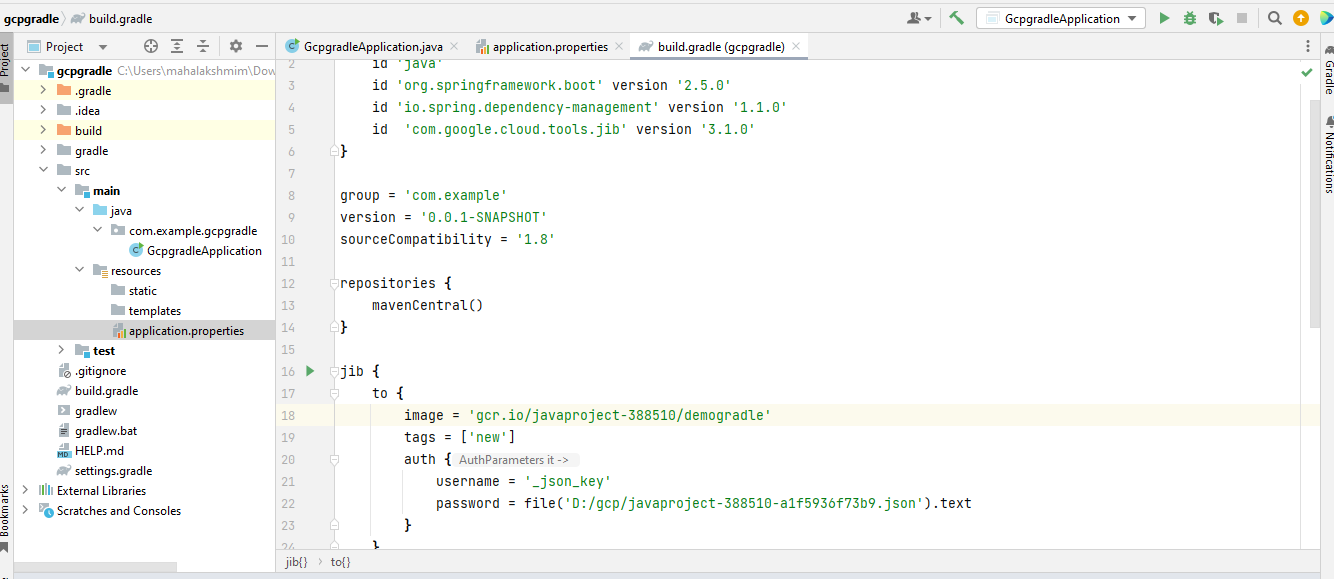
How to create a spring boot application and how to push the jib into docker with step by step screenshot in detail

Step1:create a sample spring boot application



Step2:After that in build.gradle need to add some configuration setup for pushing a jib in docker as a image

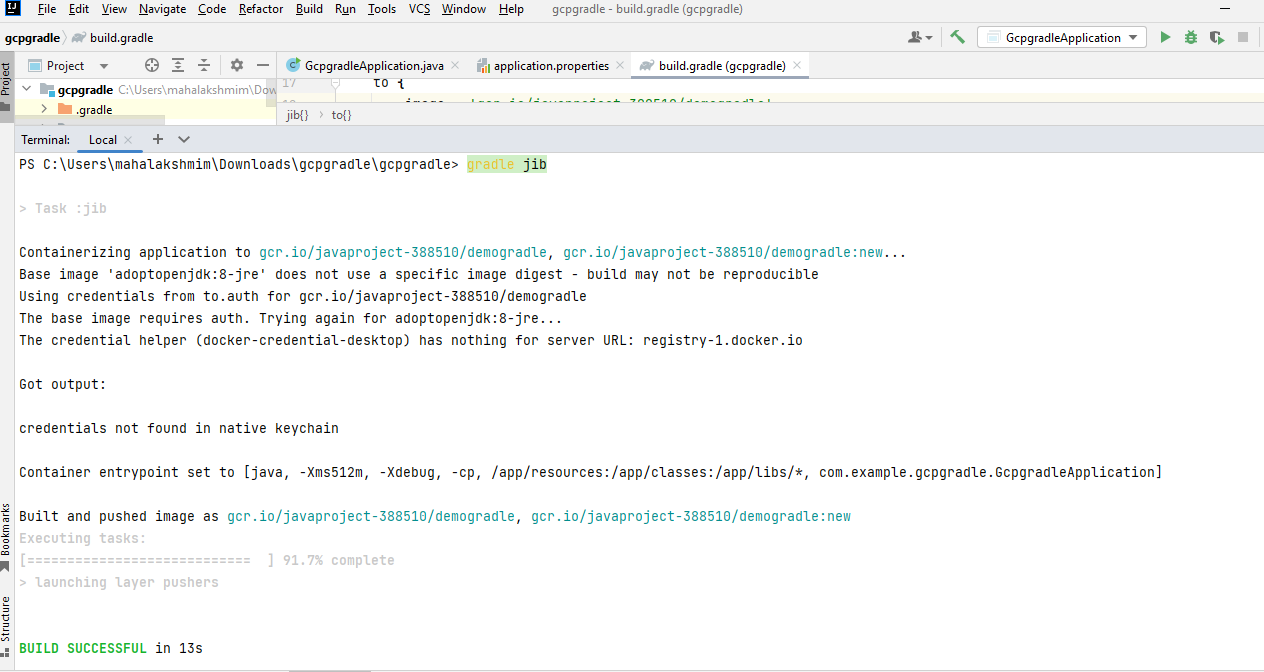


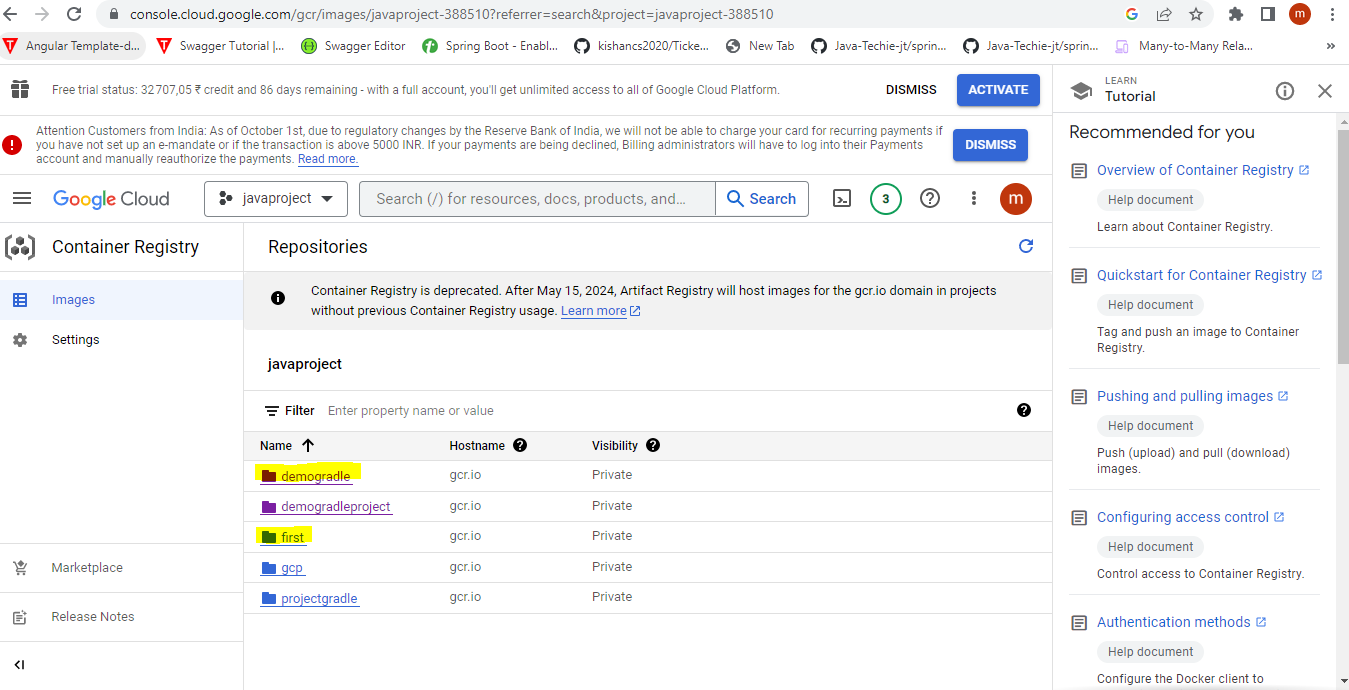


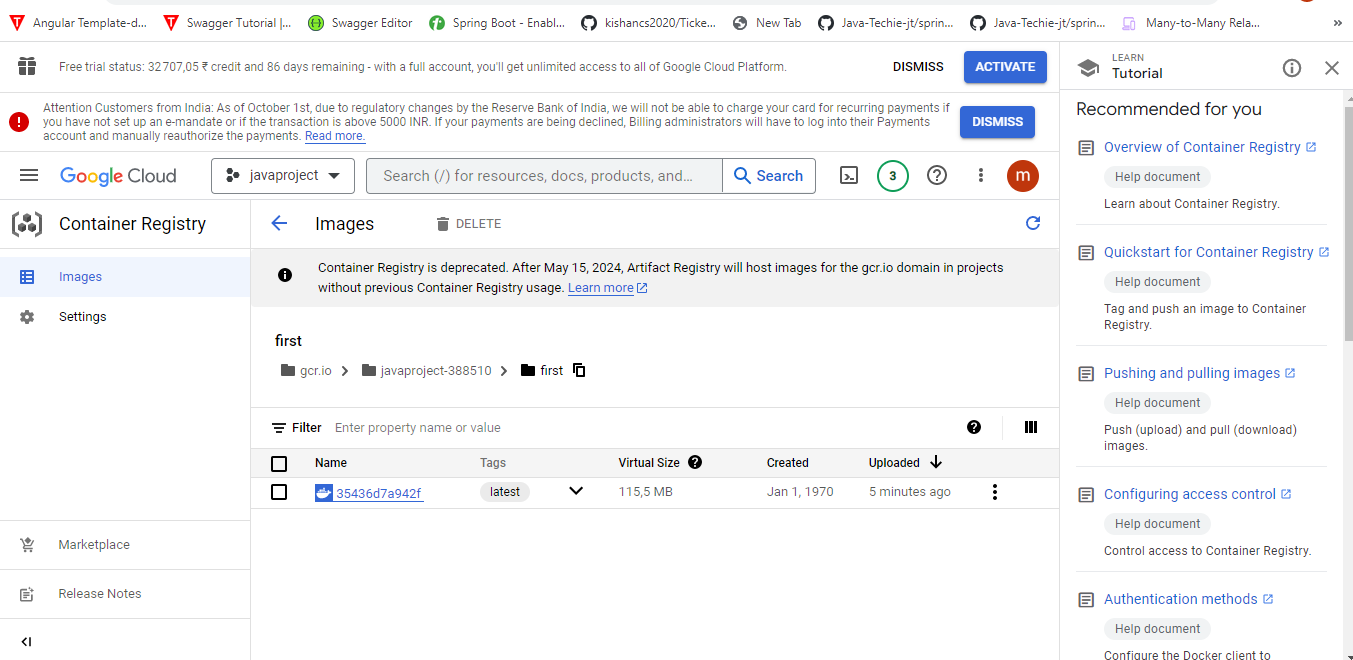
plugins **{** id 'java'  
 id 'org.springframework.boot' version '2.5.0'  
 id 'io.spring.dependency-management' version '1.1.0'  
 id 'com.google.cloud.tools.jib' version '3.1.0'  
 id 'maven-publish'  
**}**group = 'com.example'  
version = '0.0.1-SNAPSHOT'  
sourceCompatibility = '1.8'  
  
repositories **{** mavenCentral()  
**}**jib **{** to **{** image = 'mahahari/gradle'  
 tags = ['latest']  
 auth **{** username = 'mahahari'  
 password = 'Maha1308\*'  
 **}  
 }** container **{** jvmFlags = ['-Xms512m', '-Xmx1024m']  
 mainClass = 'com.example.gcp'  
 ports = ['8080', '9090']  
  
 **}  
  
}**dependencies **{** implementation 'org.springframework.boot:spring-boot-starter-data-jpa'  
 implementation 'org.springframework.boot:spring-boot-starter-web'  
 implementation ("mysql:mysql-connector-java")  
 testImplementation 'org.springframework.boot:spring-boot-starter-test'  
**}**tasks.named('test') **{** useJUnitPlatform()  
**}**

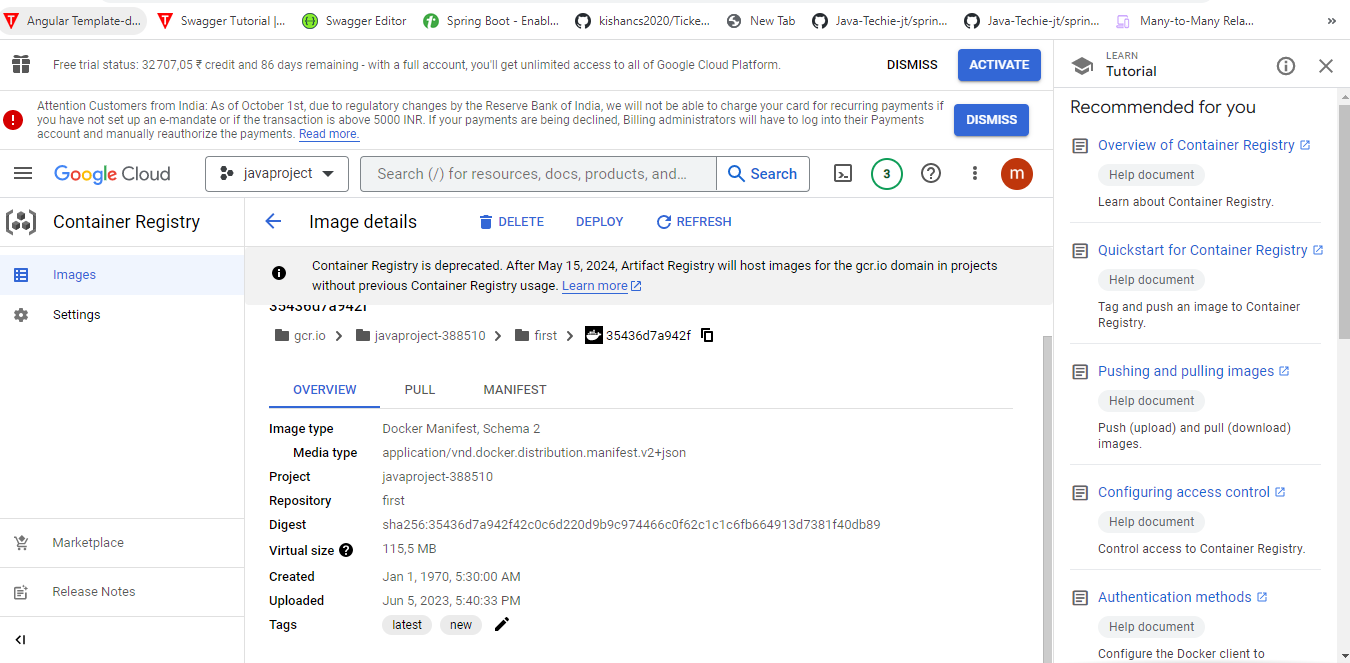


By using gradle jib command before running this command sign out from docker in local and global because sometimes it will shows some error

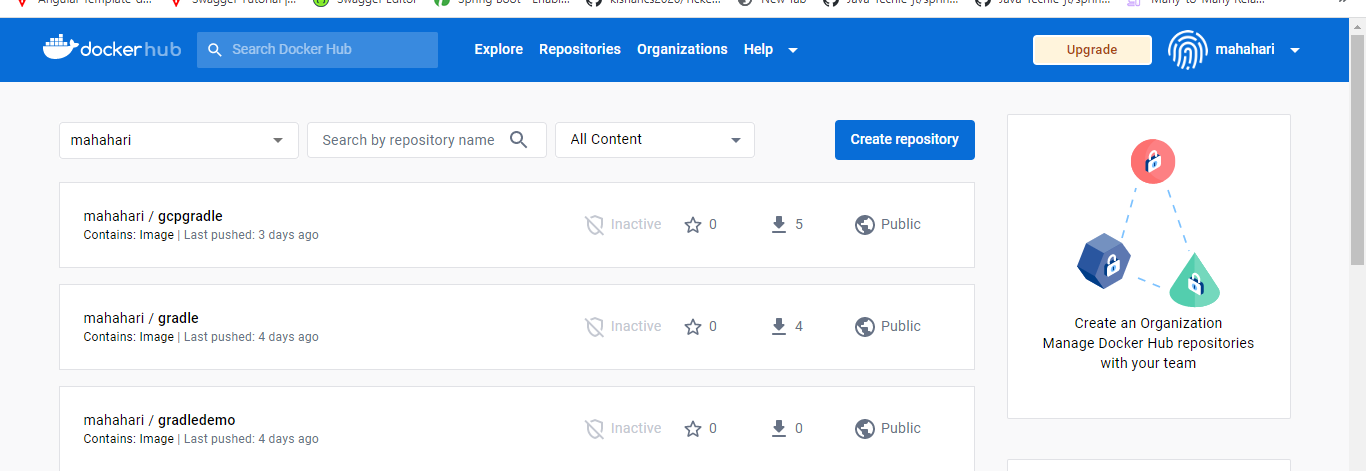








**Step3:**Based on the configuration then use **gradle jib** command it will create a repository in docker hub repository



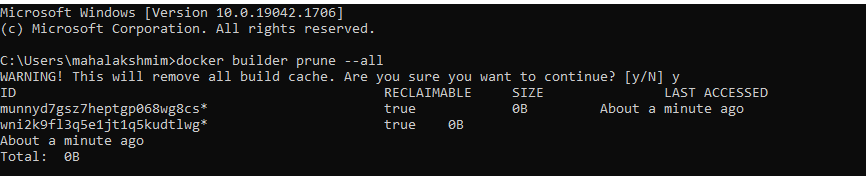
The Docker image has been created as a jib in docker hub Repository.the above example will shown the example in a clear manner.

Step4:Then how to push the image from Docker hub to GCP Artifact Registry

With commands:

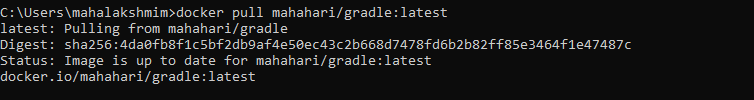
1.before using this command you need to run the Docker Desktop locally then only the Docker command will run and also you need to configure the google cli SDK command line In Local system.

**1.Docker builder prune –all**



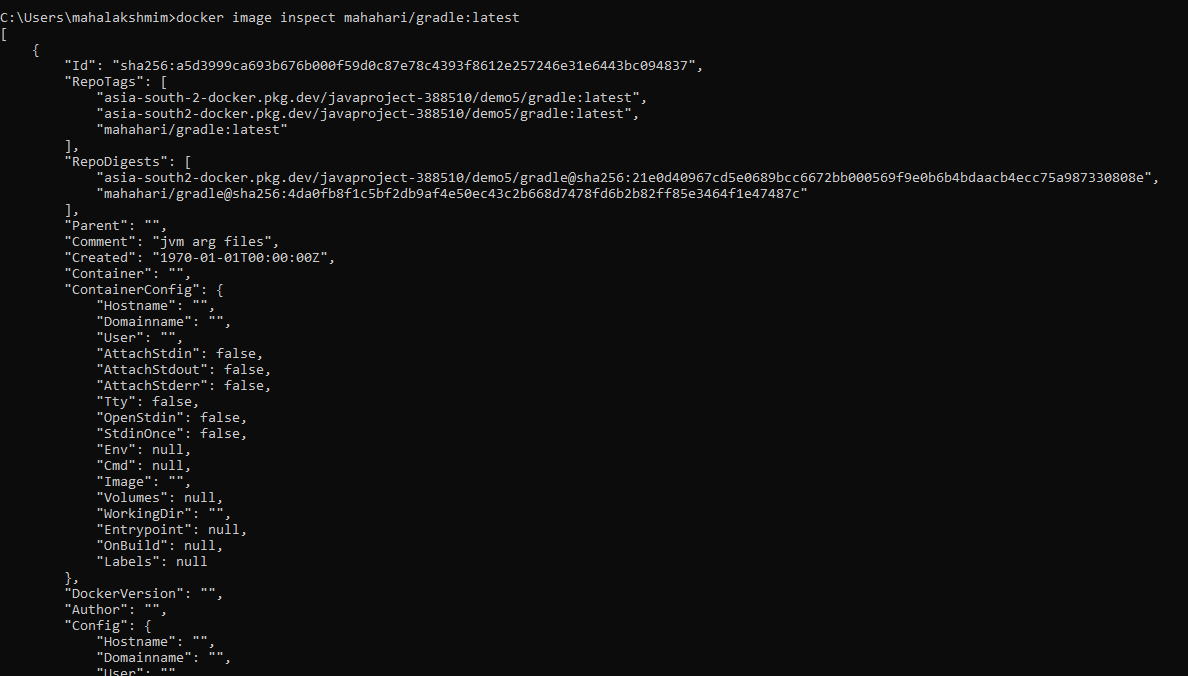
**2.Docker pull DockerHub\_username/Image Name:tag**

**Eg:docker pull mahahari/gradle :latest**



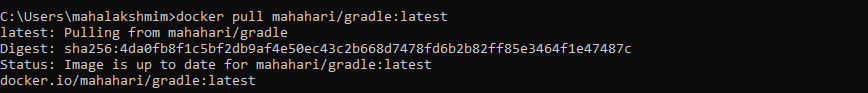
**3.Docker image inspect DockerHub\_userName/Image\_name:tag**

**Eg:docker image inspect mahahari/gradle:latest**



4.docker pull Dockerhub\_userName/Image\_name:tag

Eg:docker pull mahahari/gradle:latest

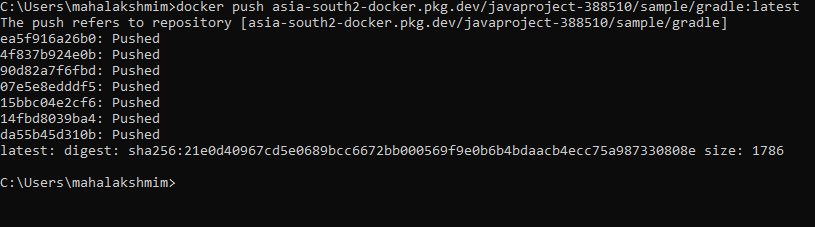


5.

docker tag mahahari/gradle:latest asia-south2-docker.pkg.dev/javaproject-388510/sample/gradle:latest

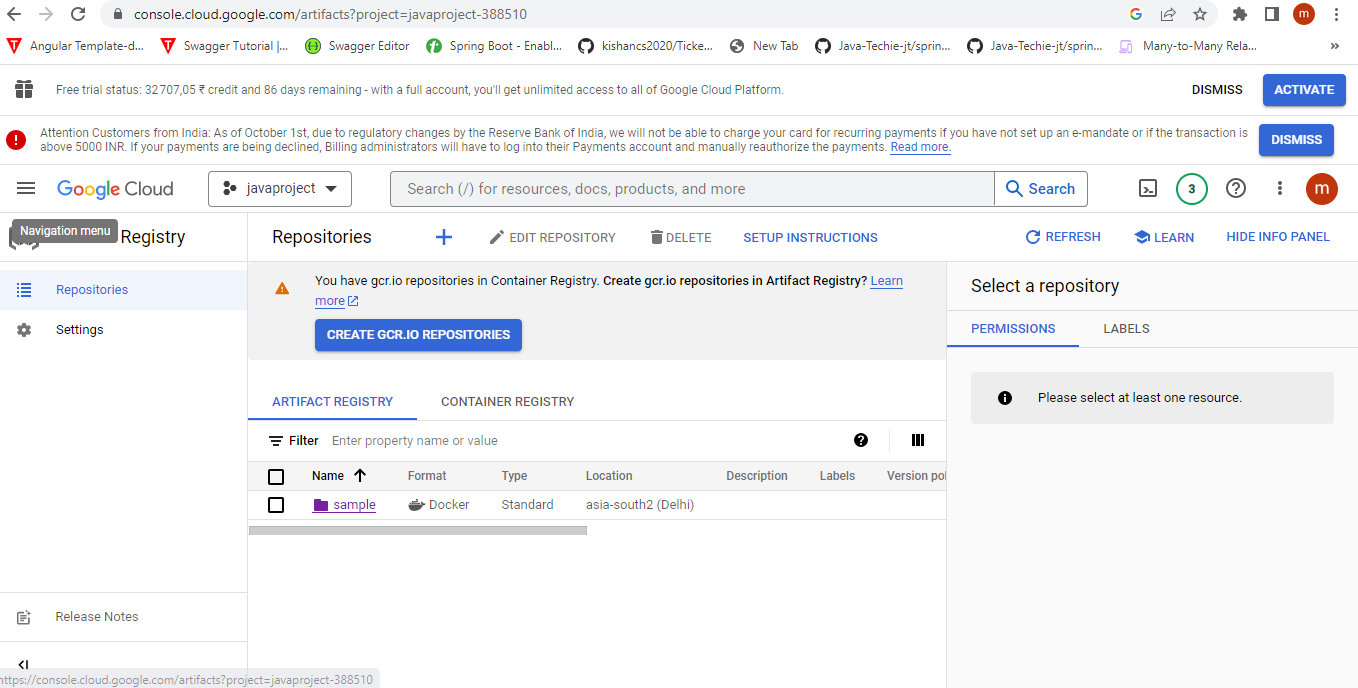


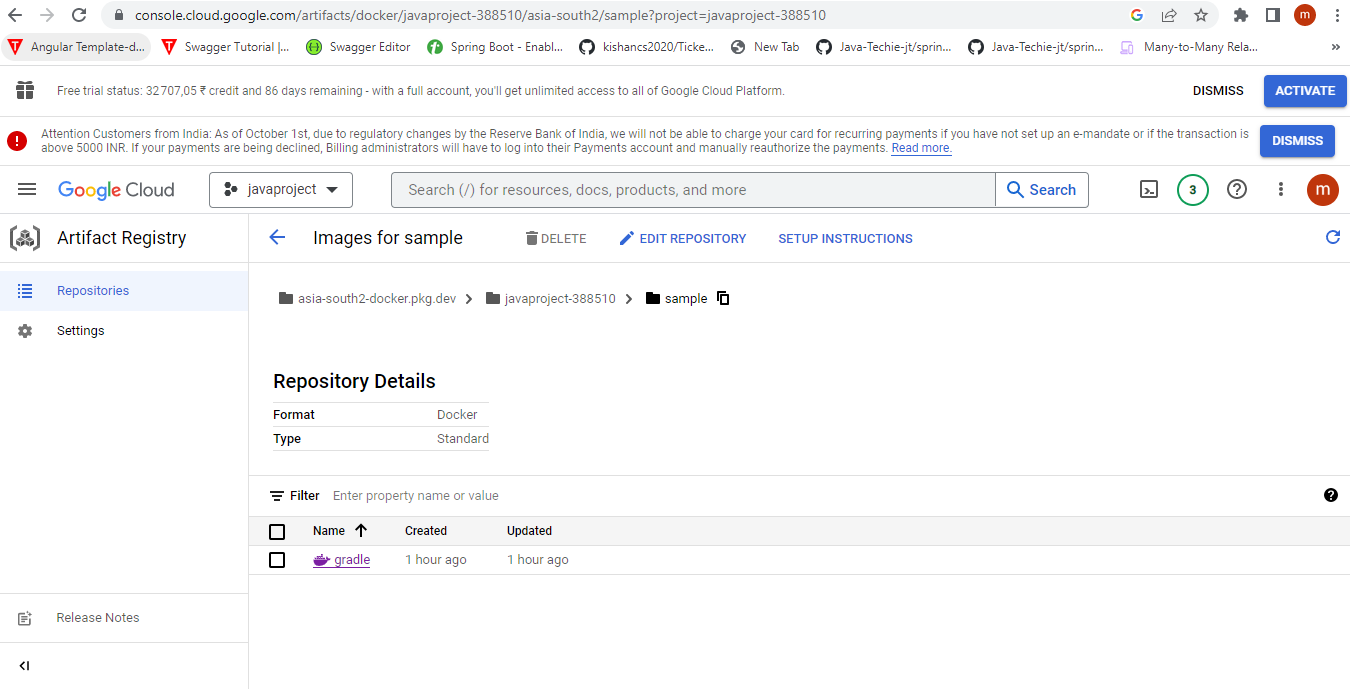
6. docker push location-docker.pkg.dev/projectname/repository\_name/image:latest

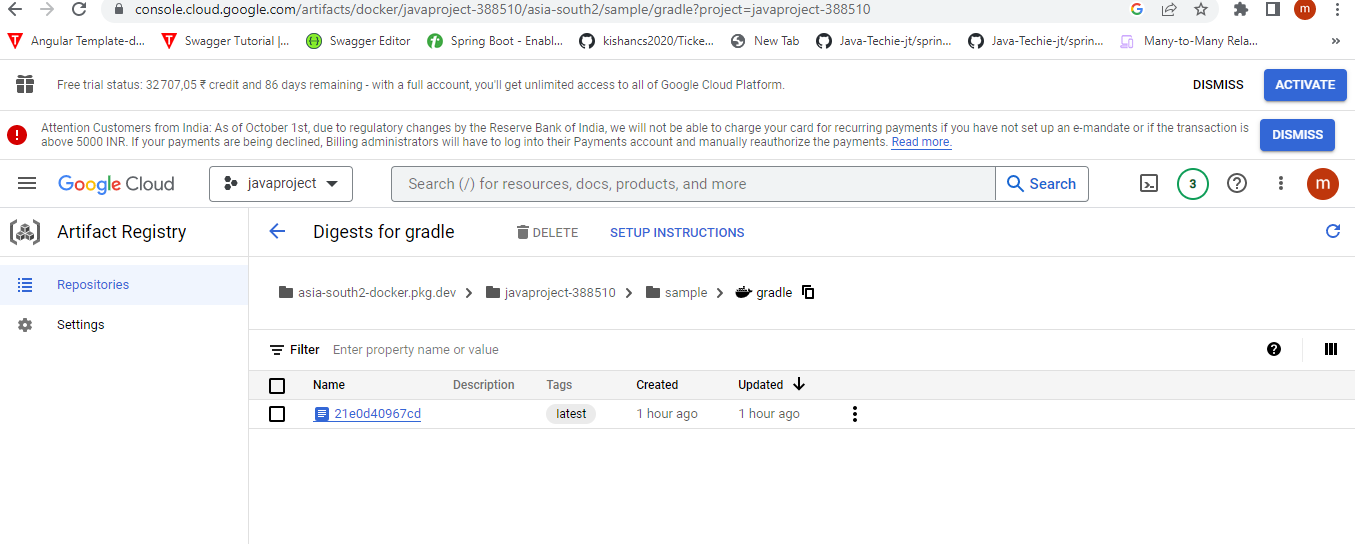
Eg:docker push asia-south2-docker.pkg.dev/javaproject-388510/sample/gradle:latest

7.then go and chck in gcp it will seen there

Go to Artifact Registry in GCp then check the Repository an image tag has been in Gcp go and chcek it means an image has been generated







Atlast an image tag has been generated based on the jib in spring boot application

