Nexus 3 Repository manager – using Gradle

Step1 – Running the nexus repository image in docker container

docker login

docker pull sonatype/nexus3

docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

sonatype/nexus3 latest b786c2be8466 7 days ago 488 MB

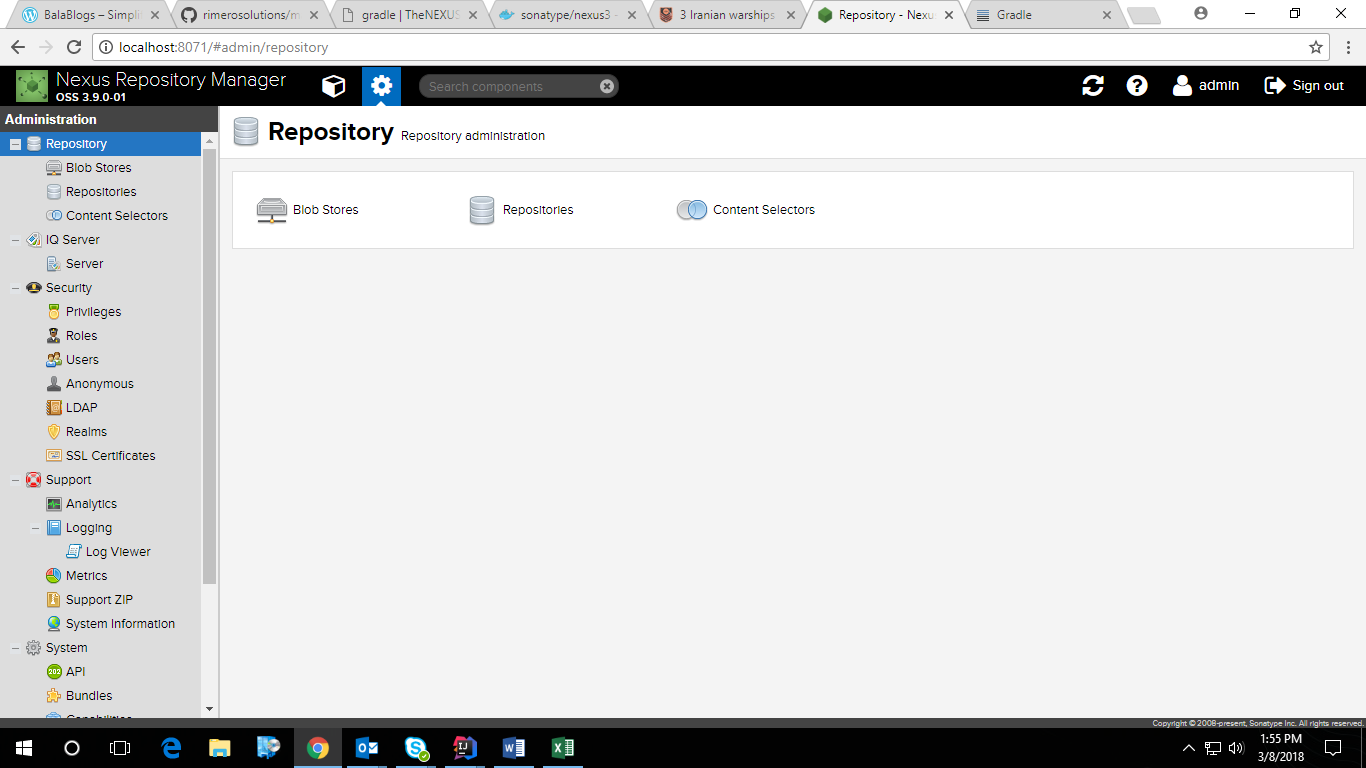
docker run -d -p 8071:8081 --name nexus sonatype/nexus3

docker container ps –a

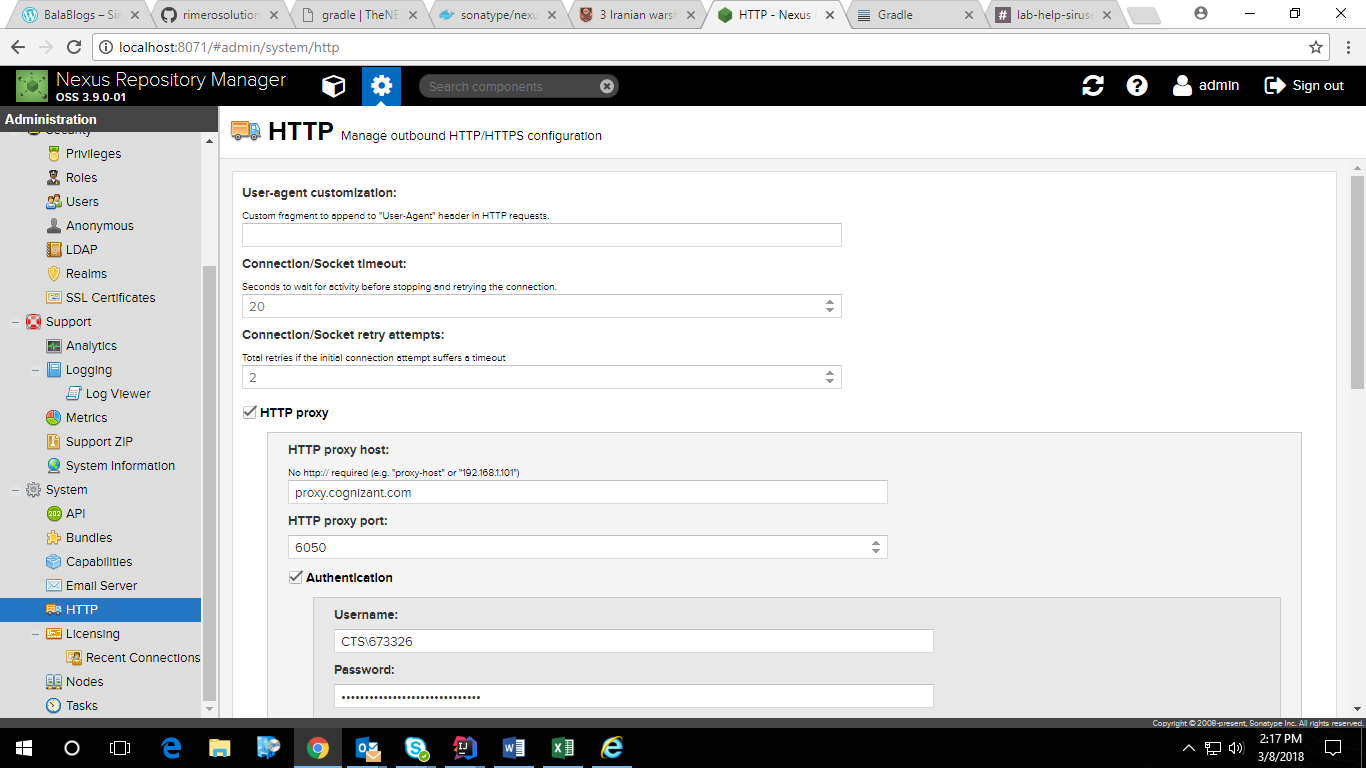
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

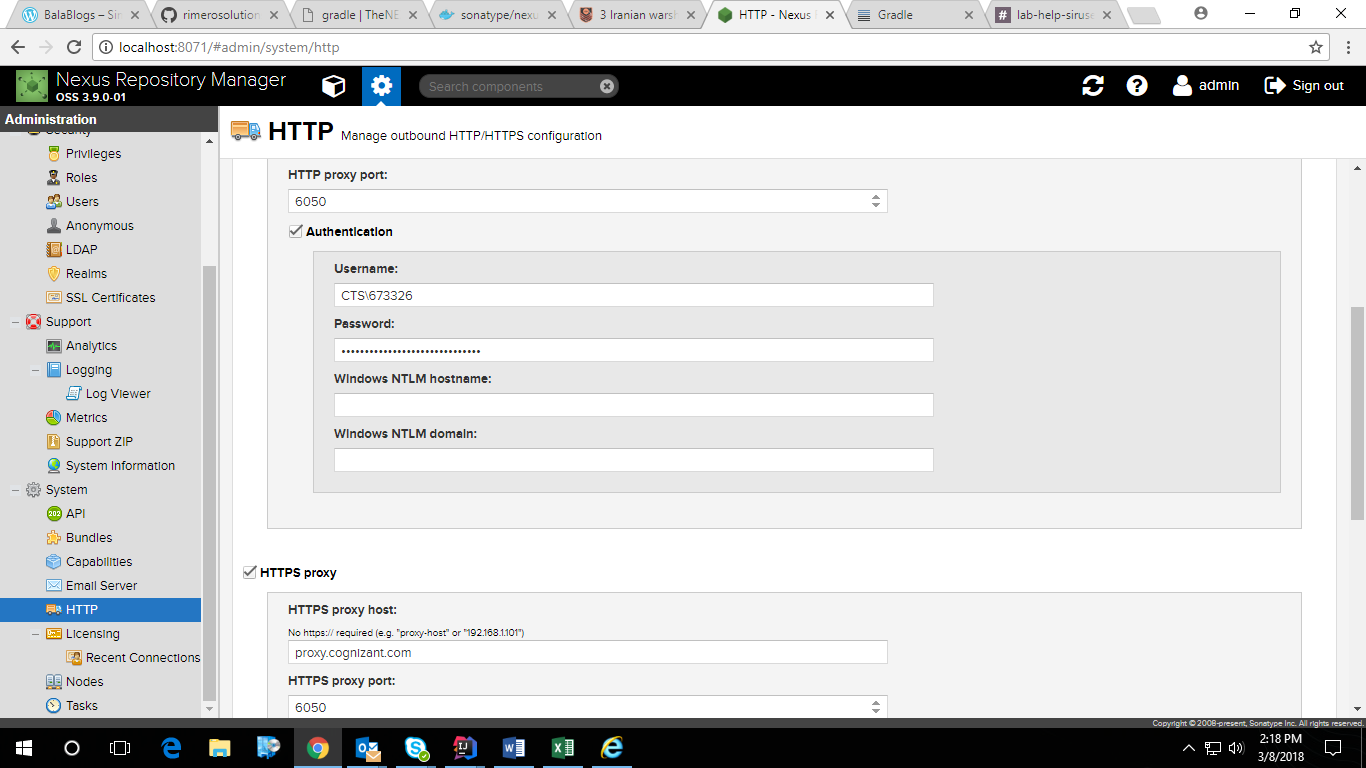
2af8de76dd3e sonatype/nexus3 "sh -c ${SONATYPE\_..." 2 days ago Up About an hour 0.0.0.0:8071->8081/tcp nexus

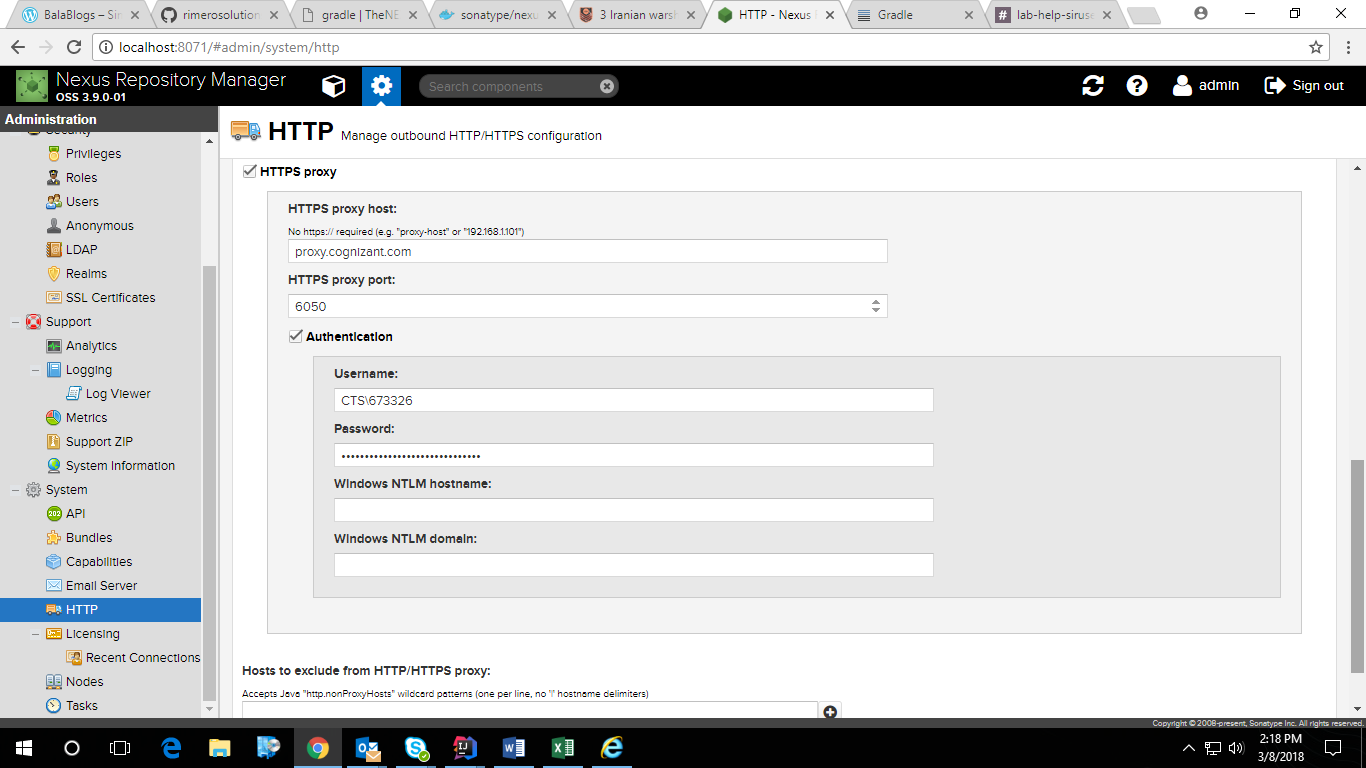
Open Browser and give url <http://localhost:8071> and login as user: admin Password: admin123

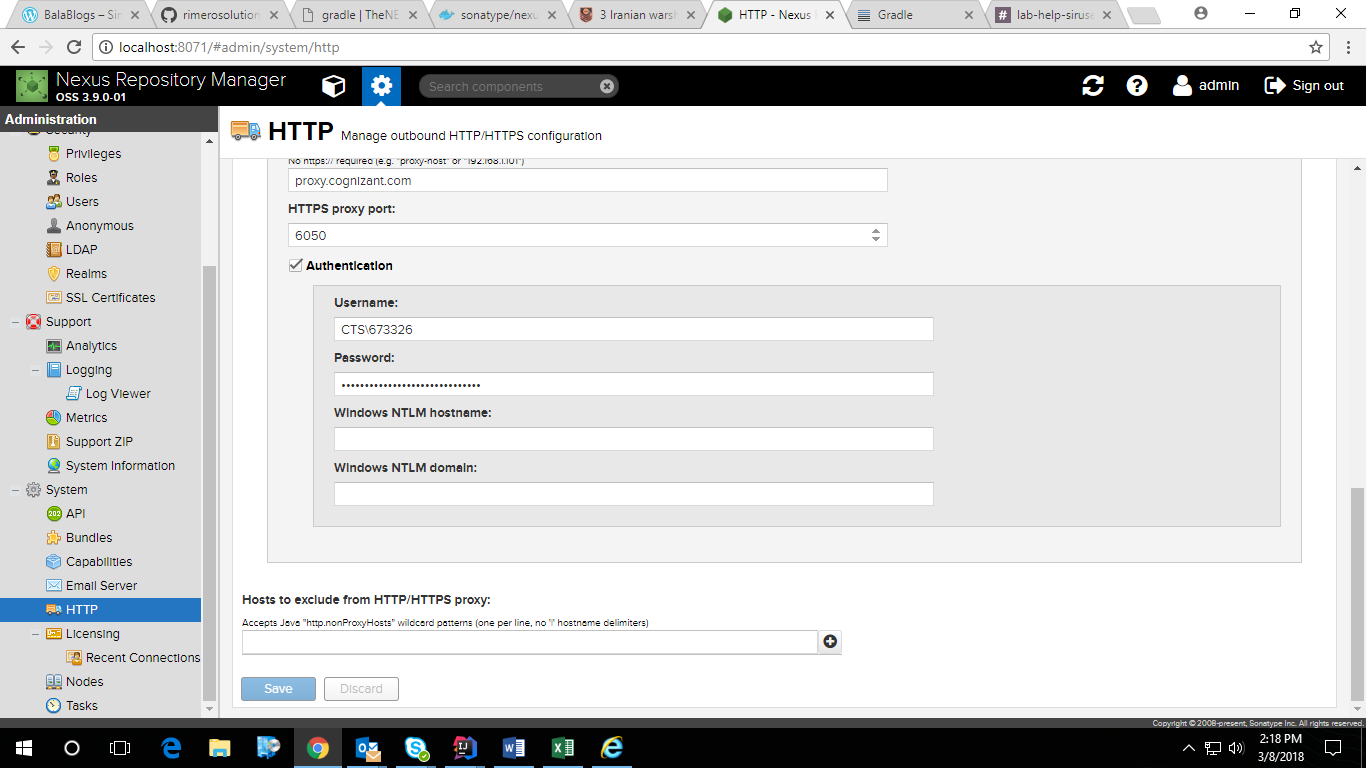


Step 2 - Configuring proxy server setting to Nexus as a Maven repo



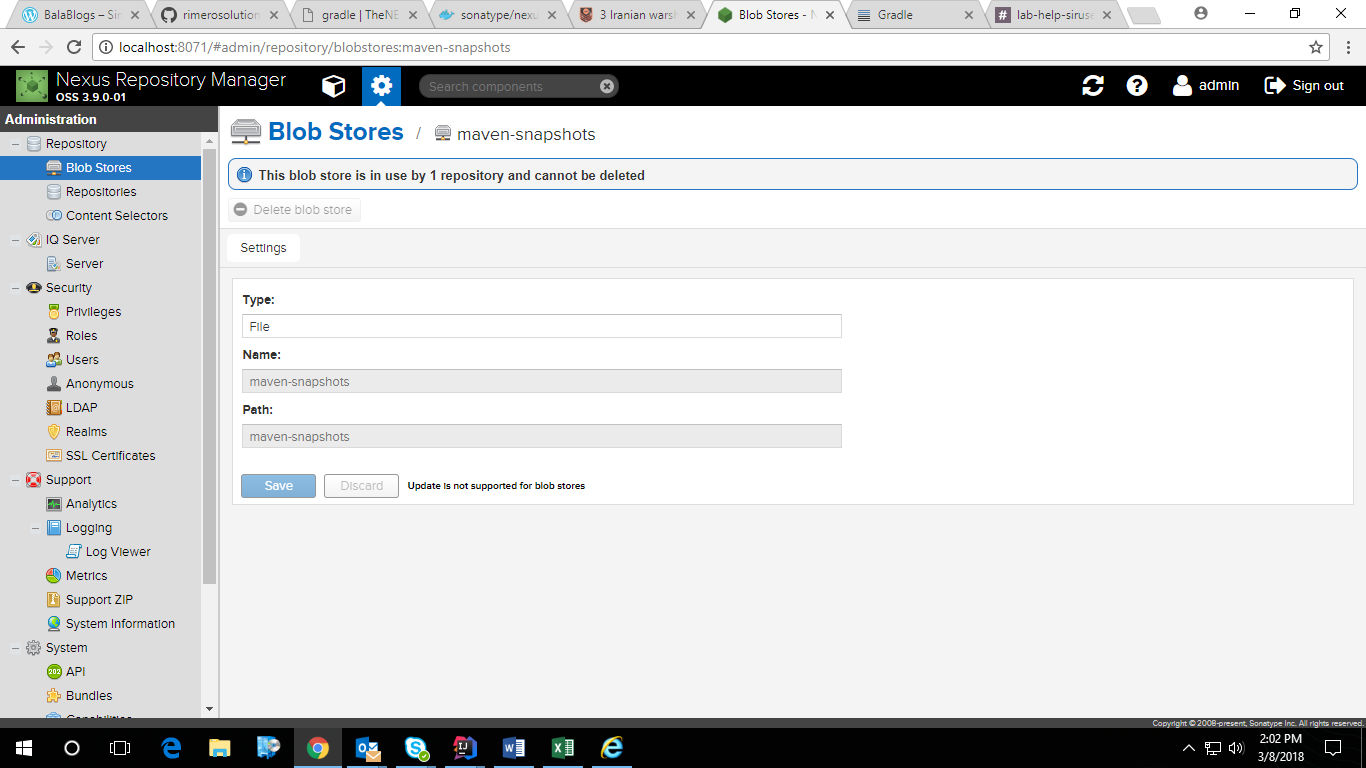




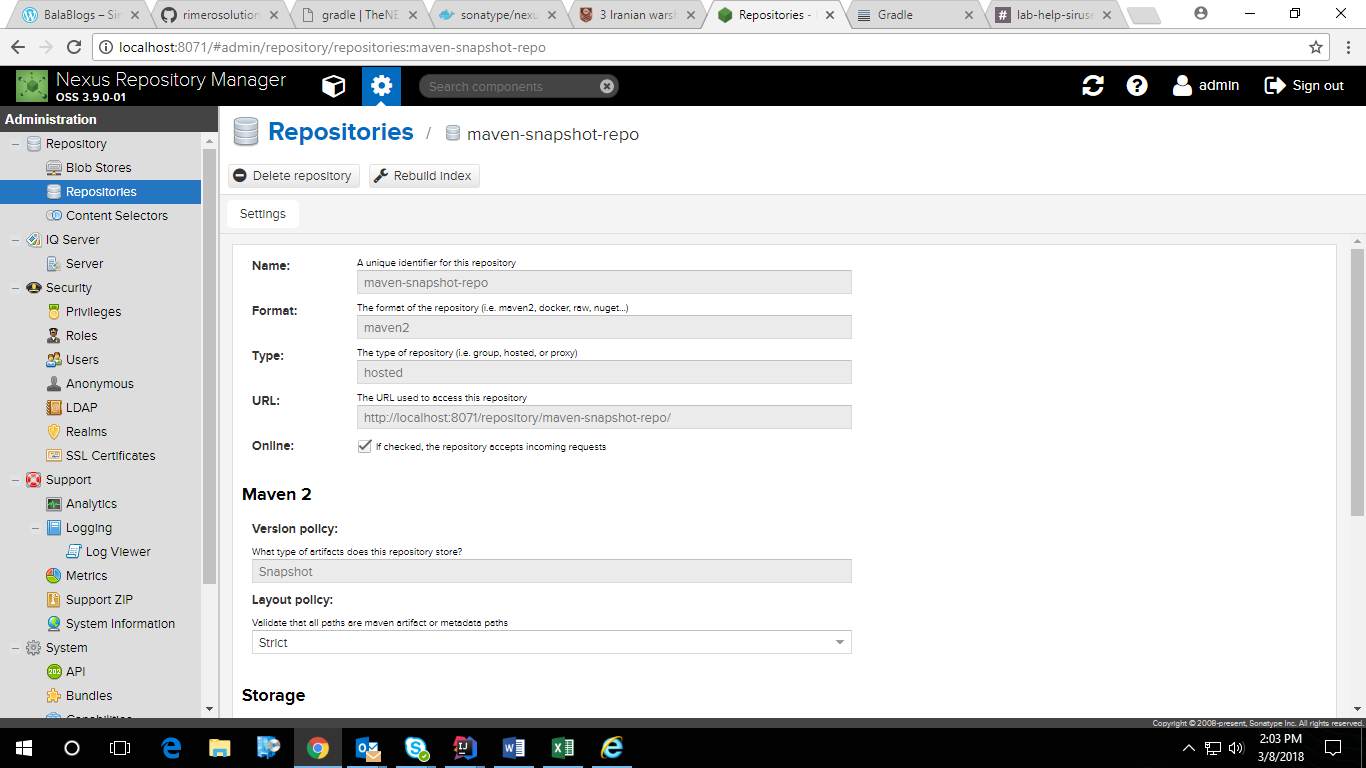


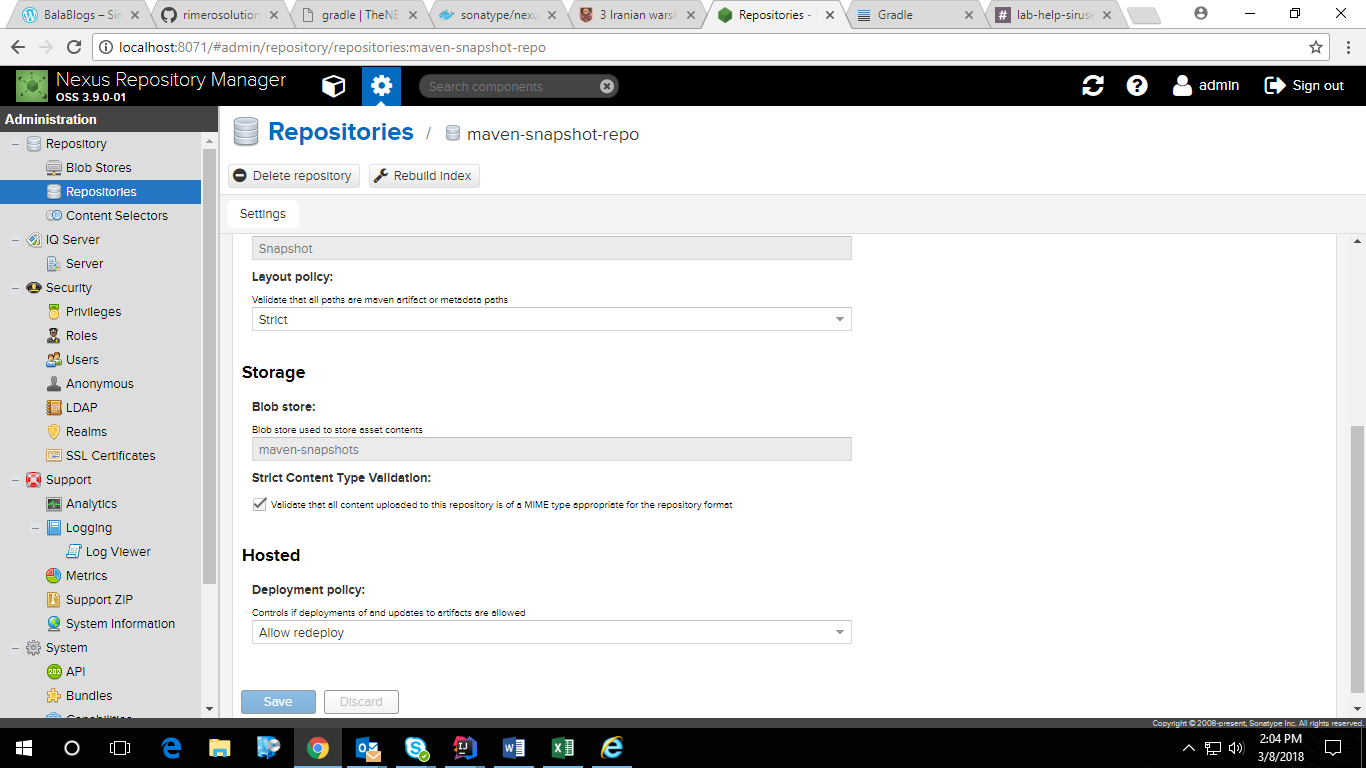
Step3 – Configuring Nexus as a Maven repo

1. create a private (hosted) repository for our snapshots
2. Create a Blob Store named maven-snapshots

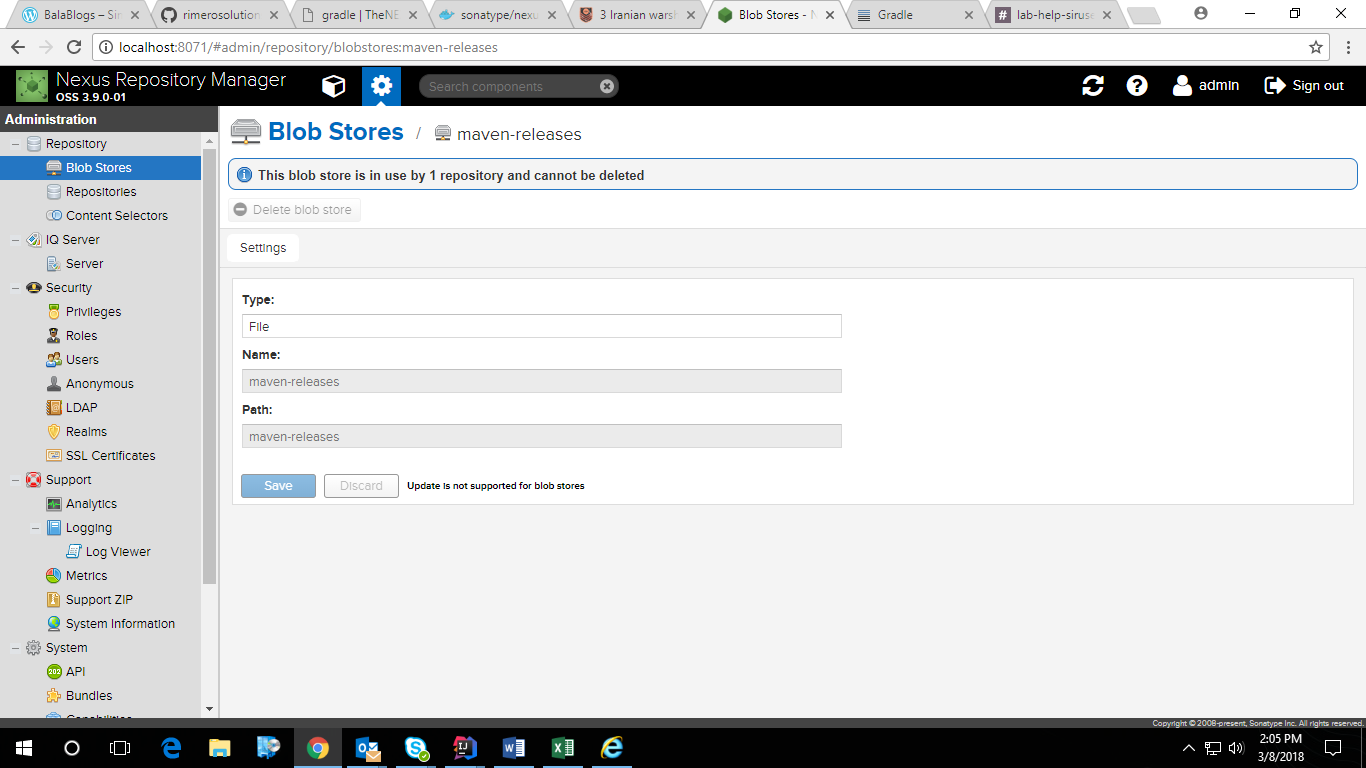


1. Create a maven-snapshots hosted repository

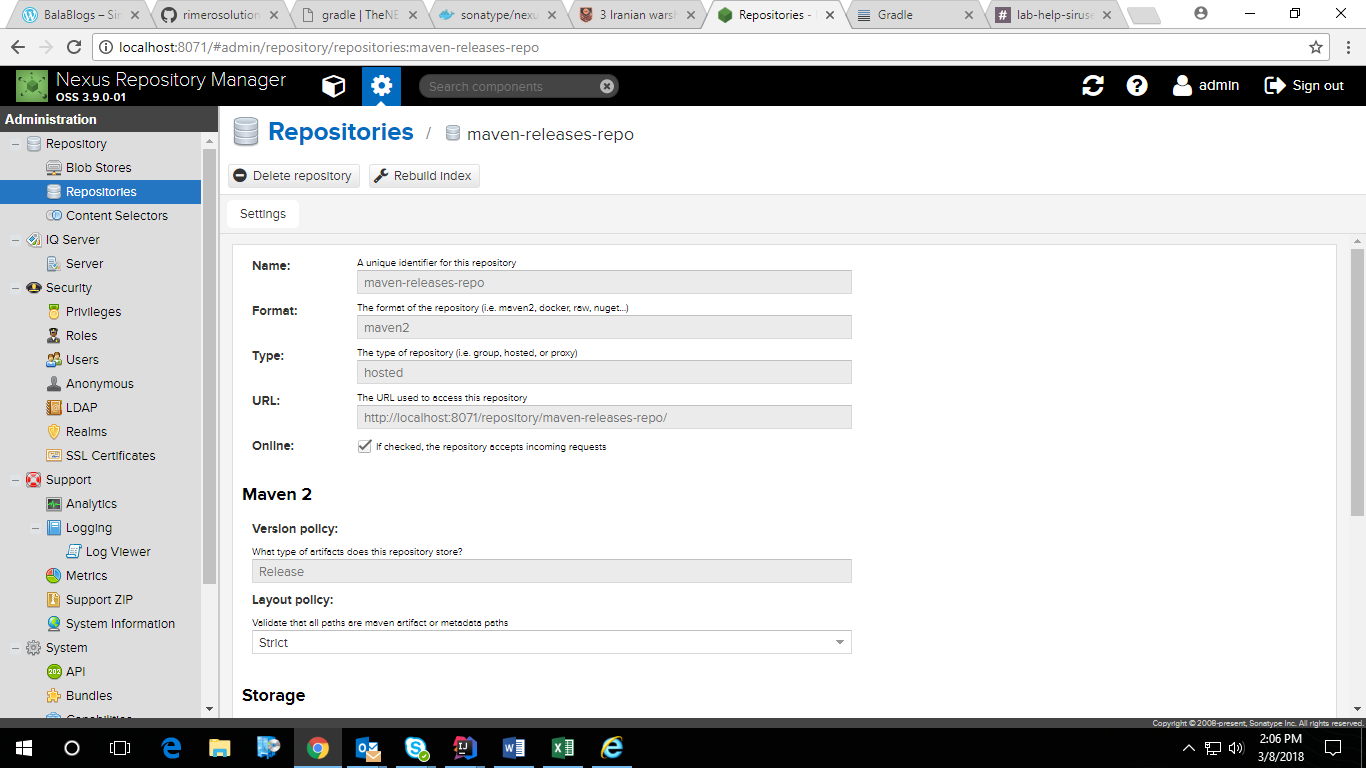


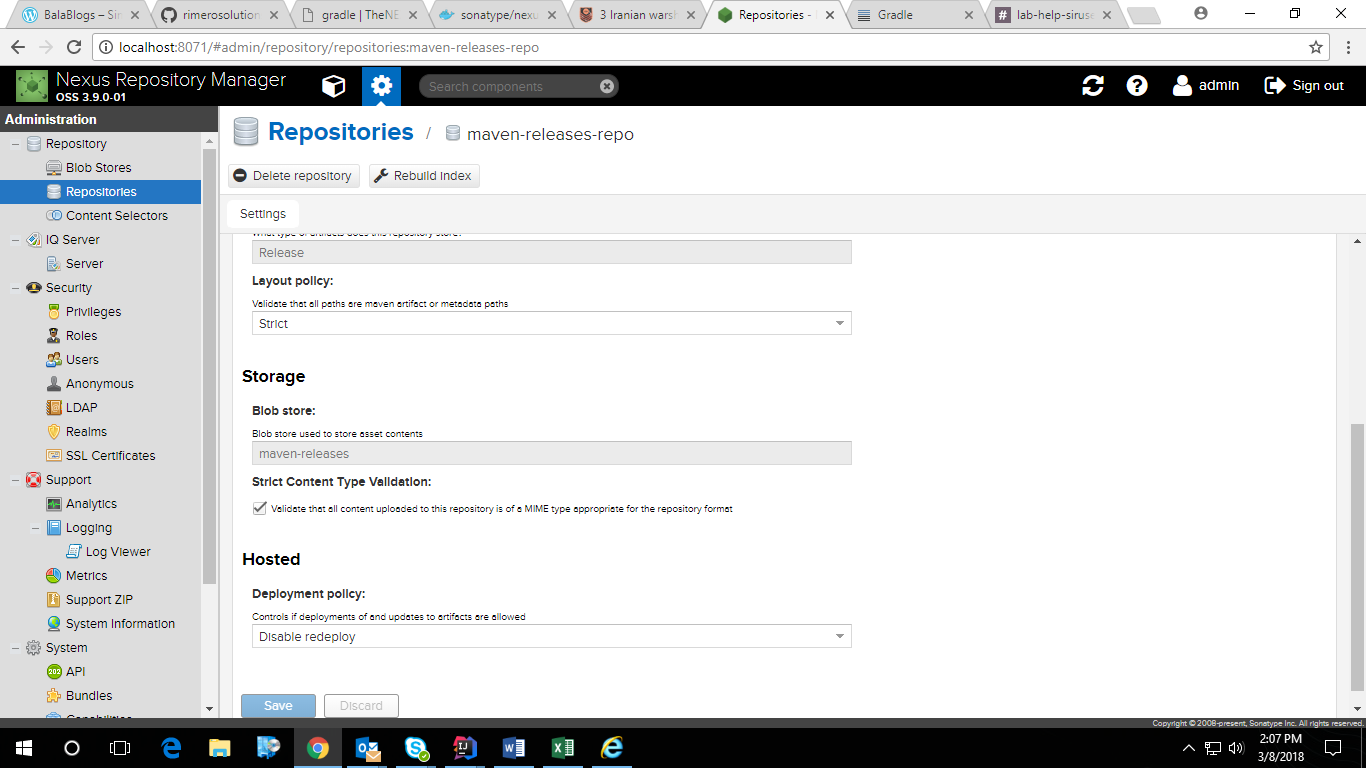


1. create a private (hosted) repository for our releases
2. Create a Blob Store named maven-releases

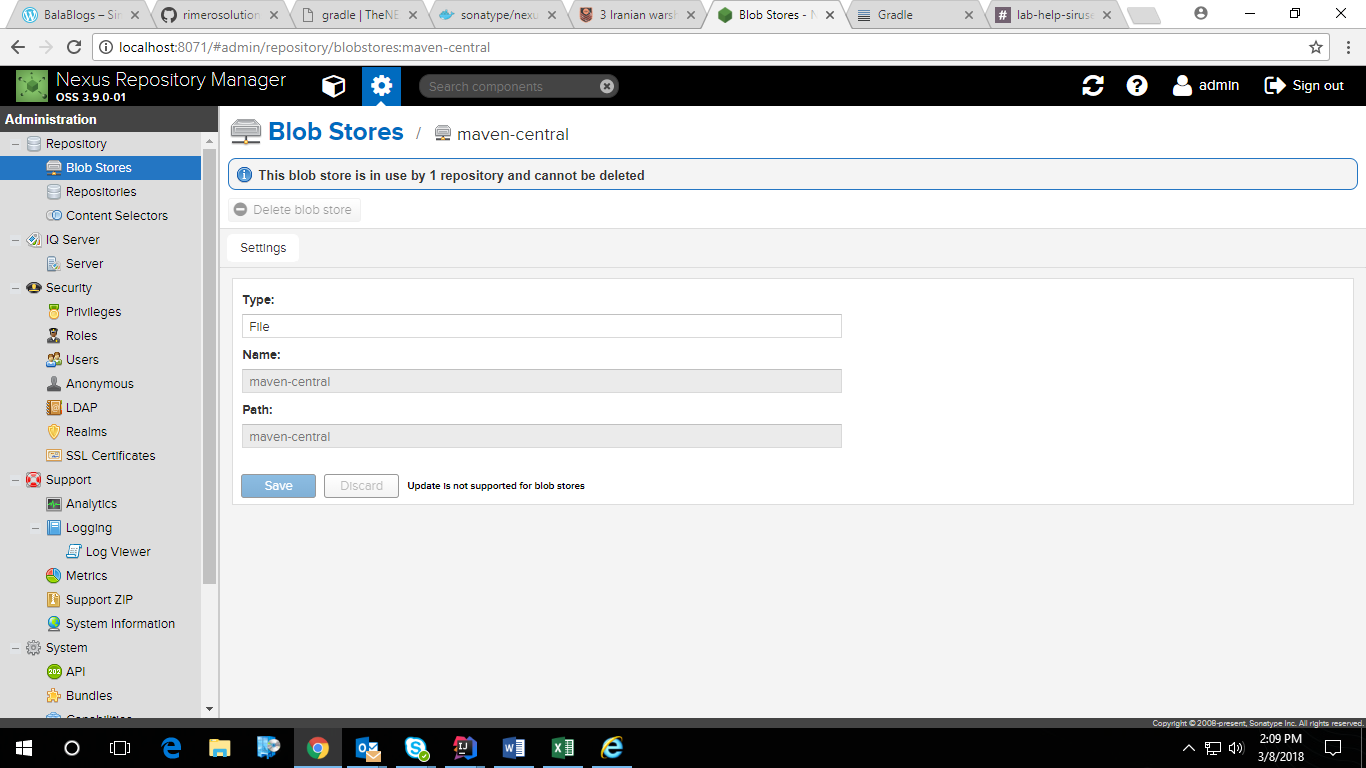


1. Create a maven-releases hosted repository

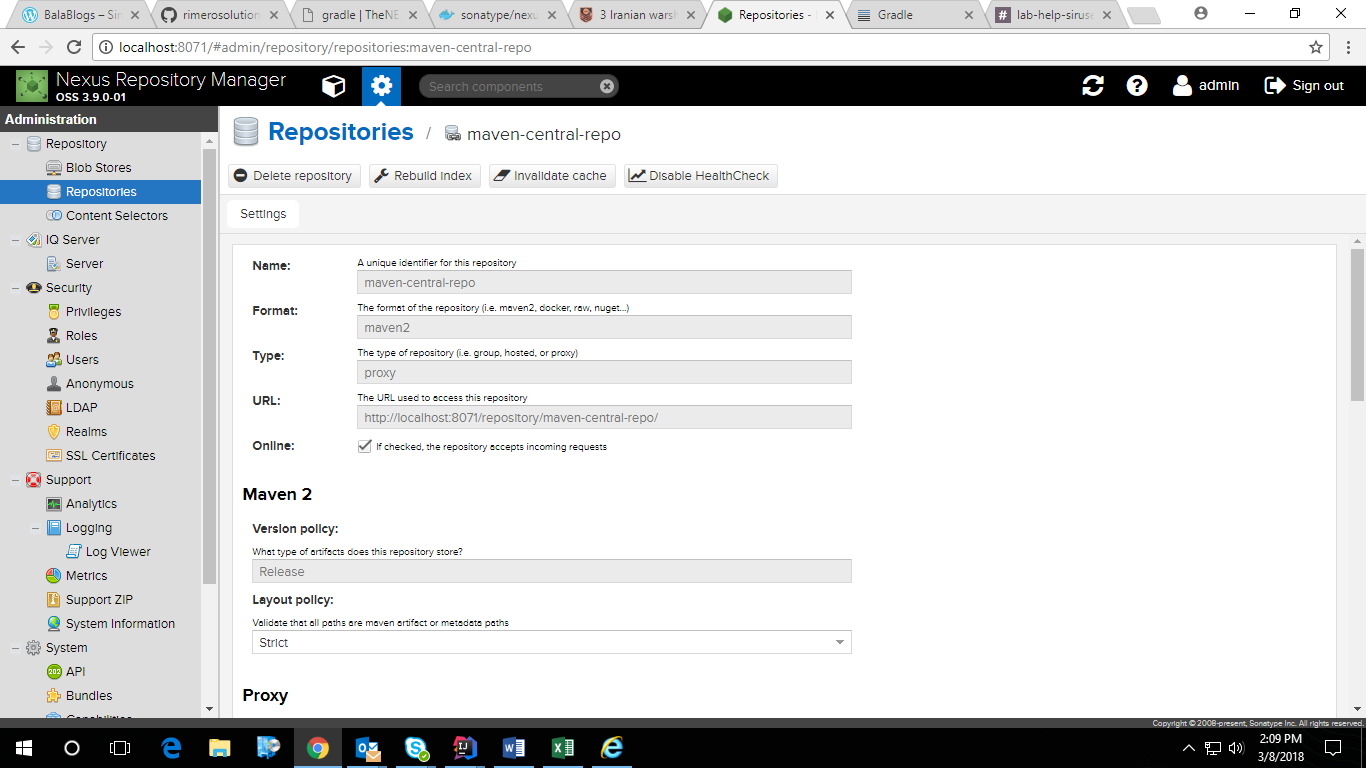


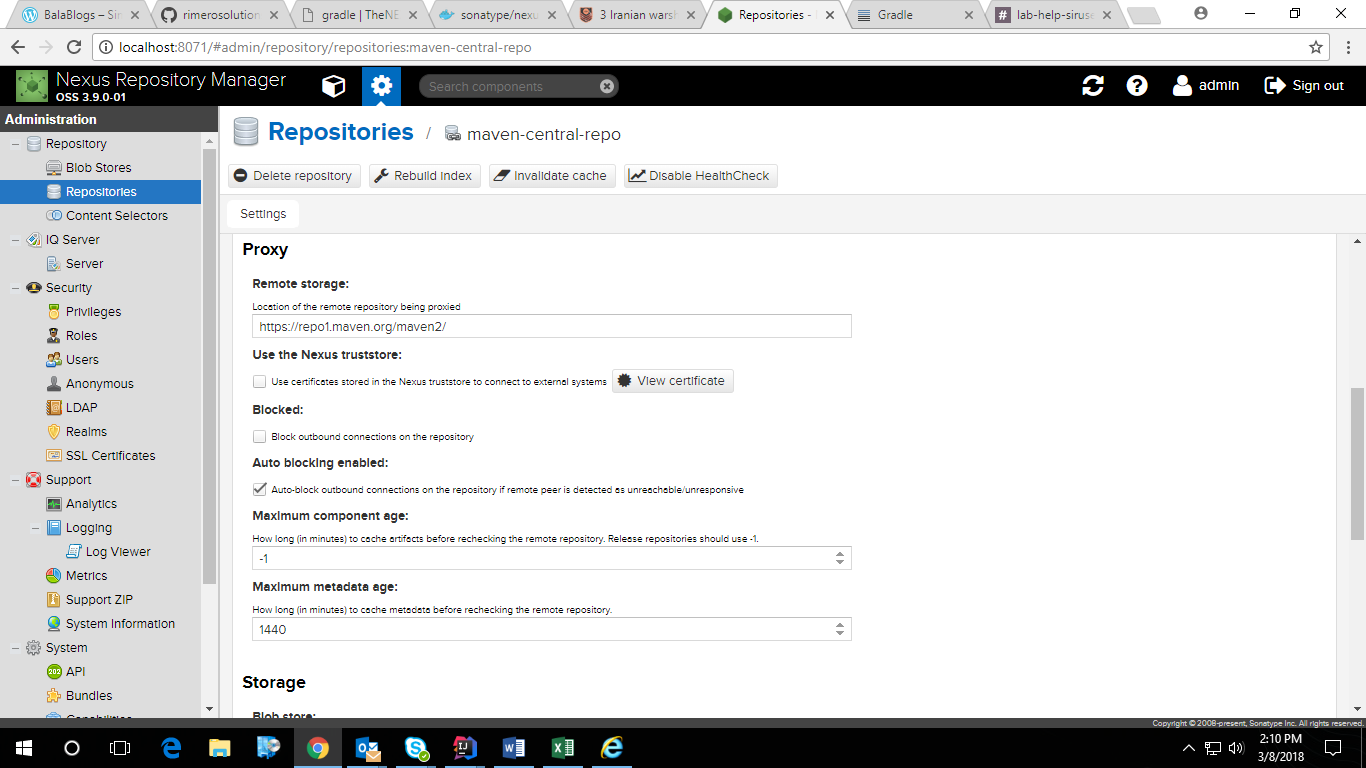


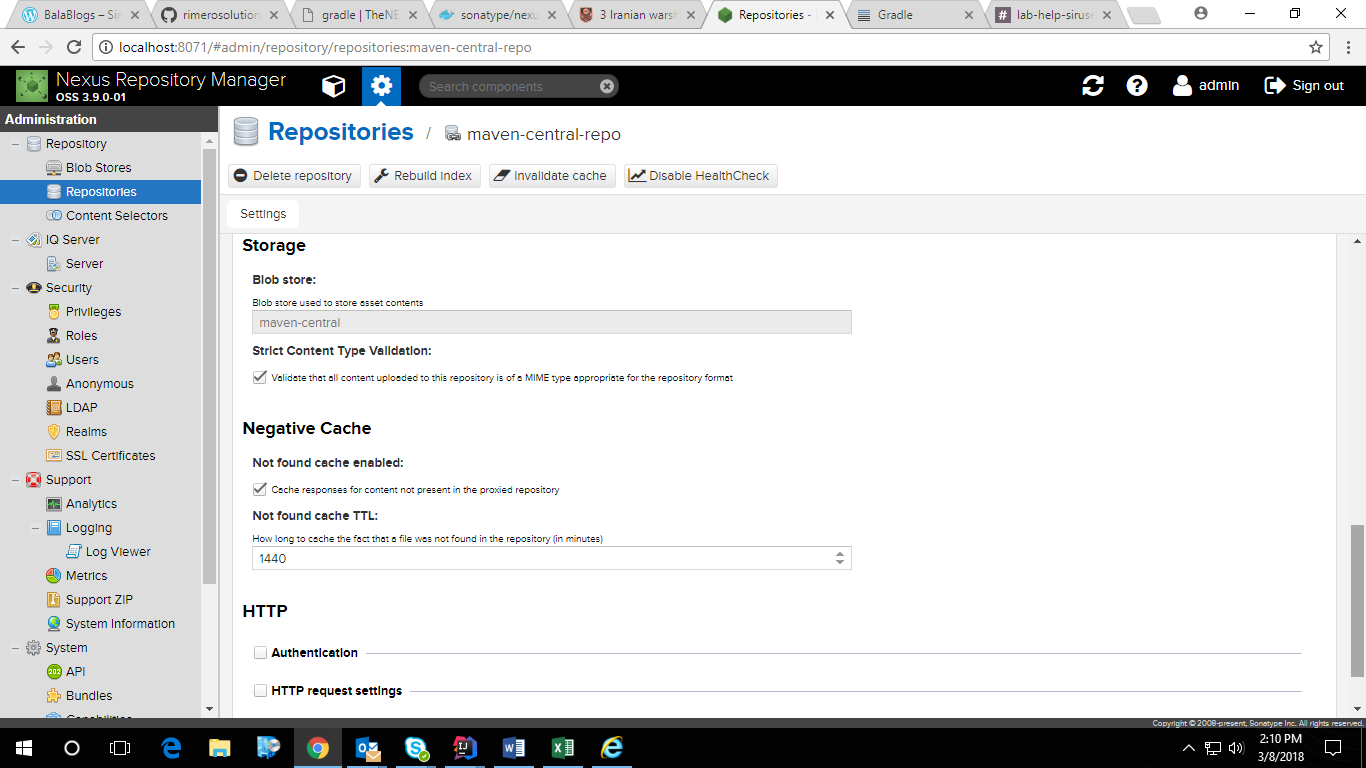
1. create a proxy repository pointing to Maven Central
2. Create a Blob Store named maven-releases

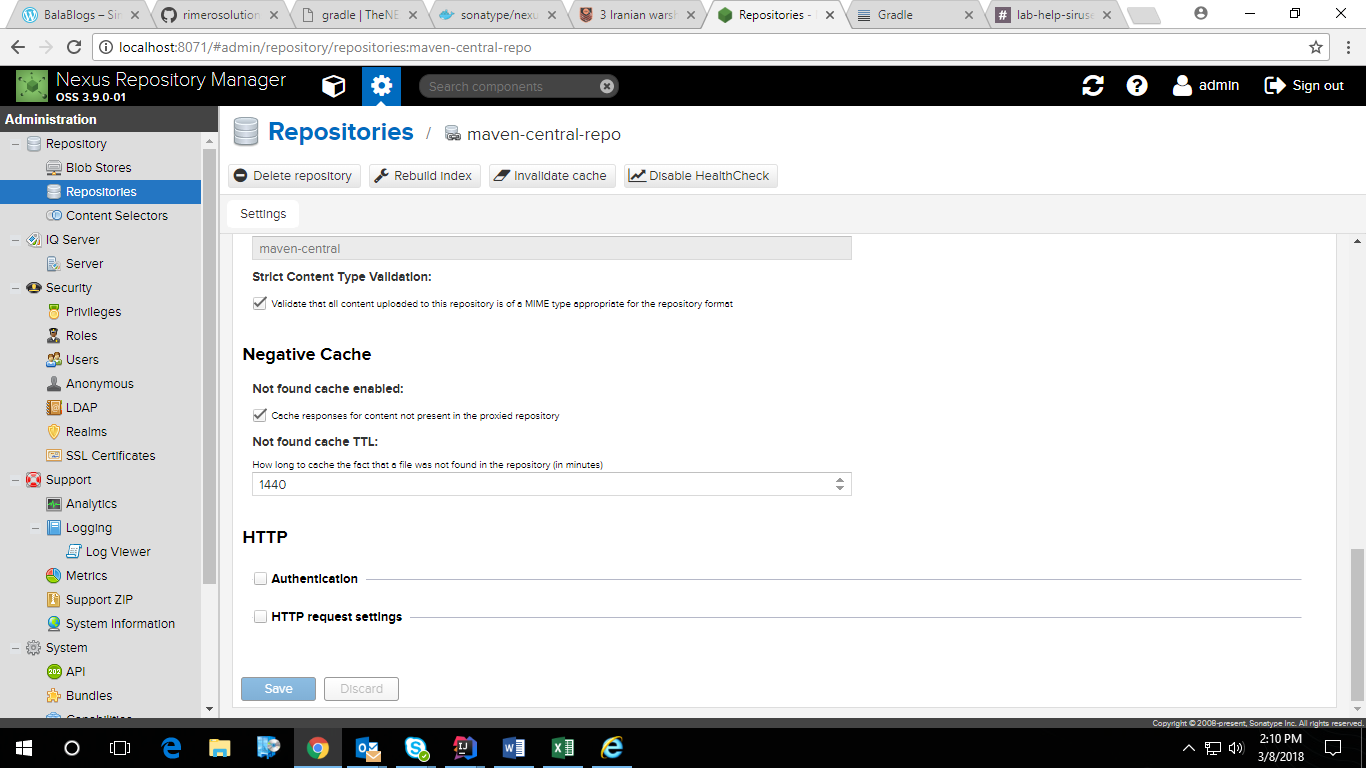


1. Create a proxy repository pointing to Maven Central

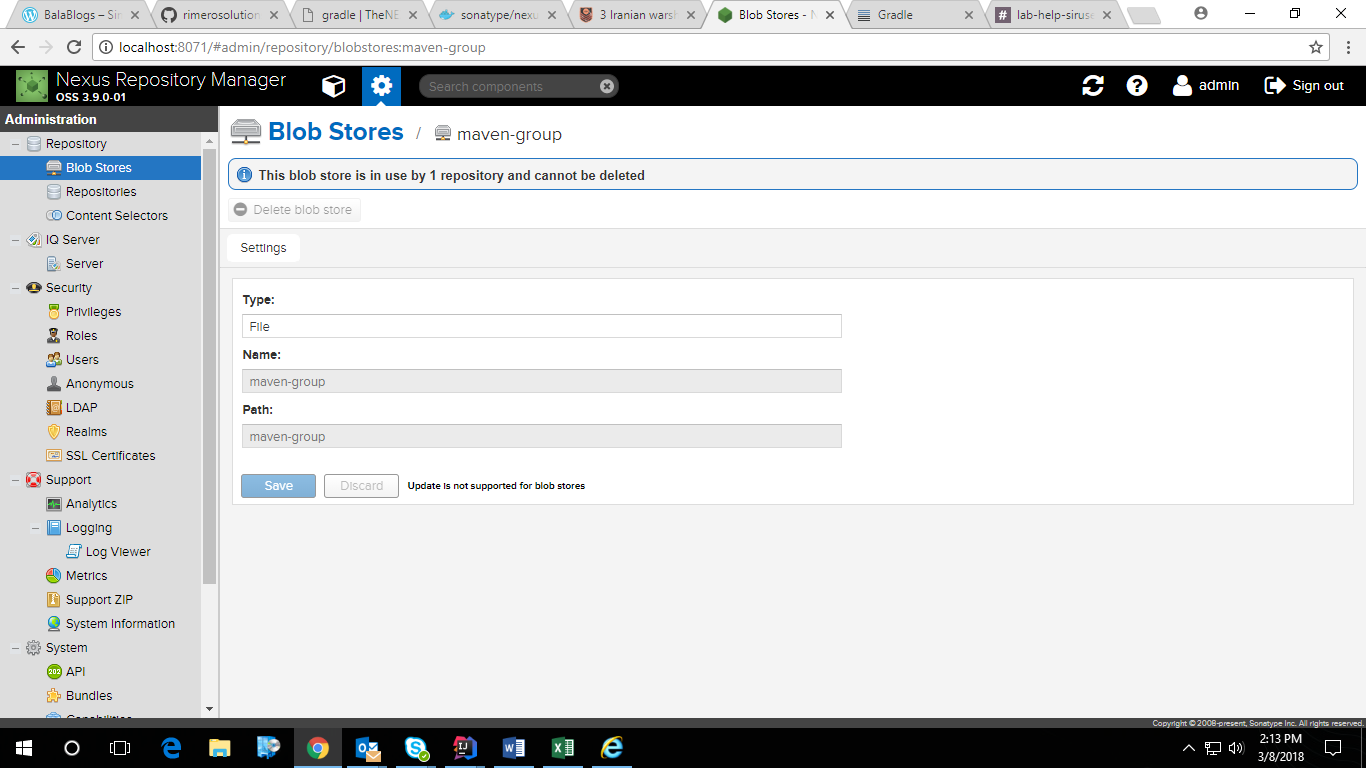




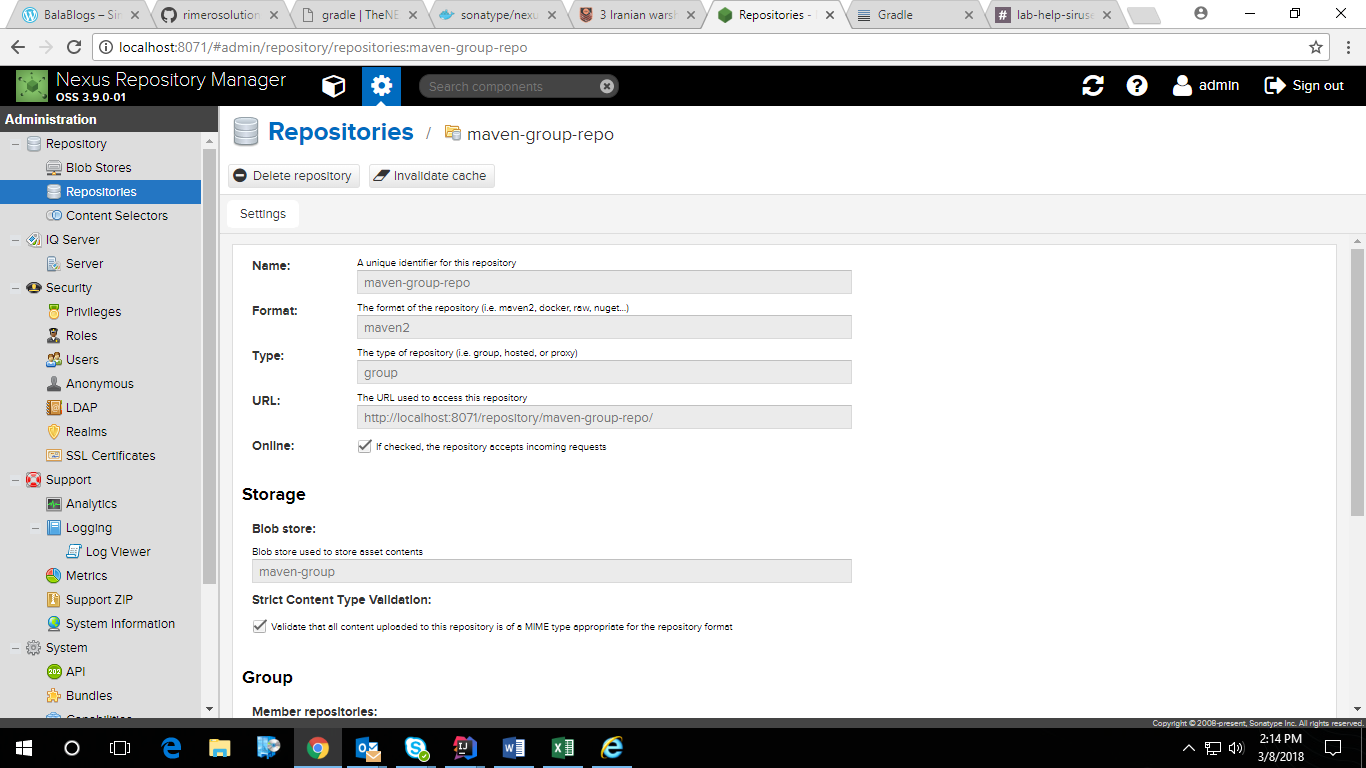


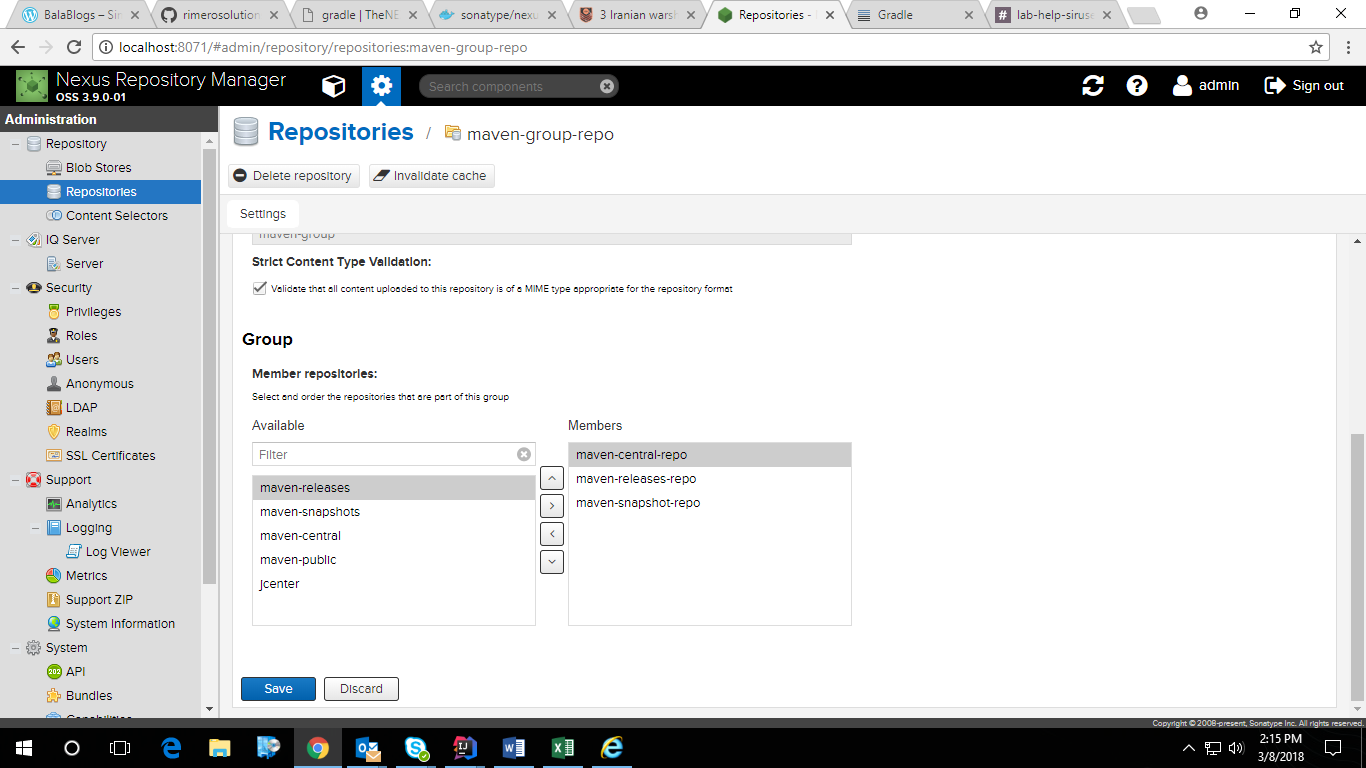


1. create a group repository to provide all of these repos under a single URL
2. Create a Blob Store named maven-releases

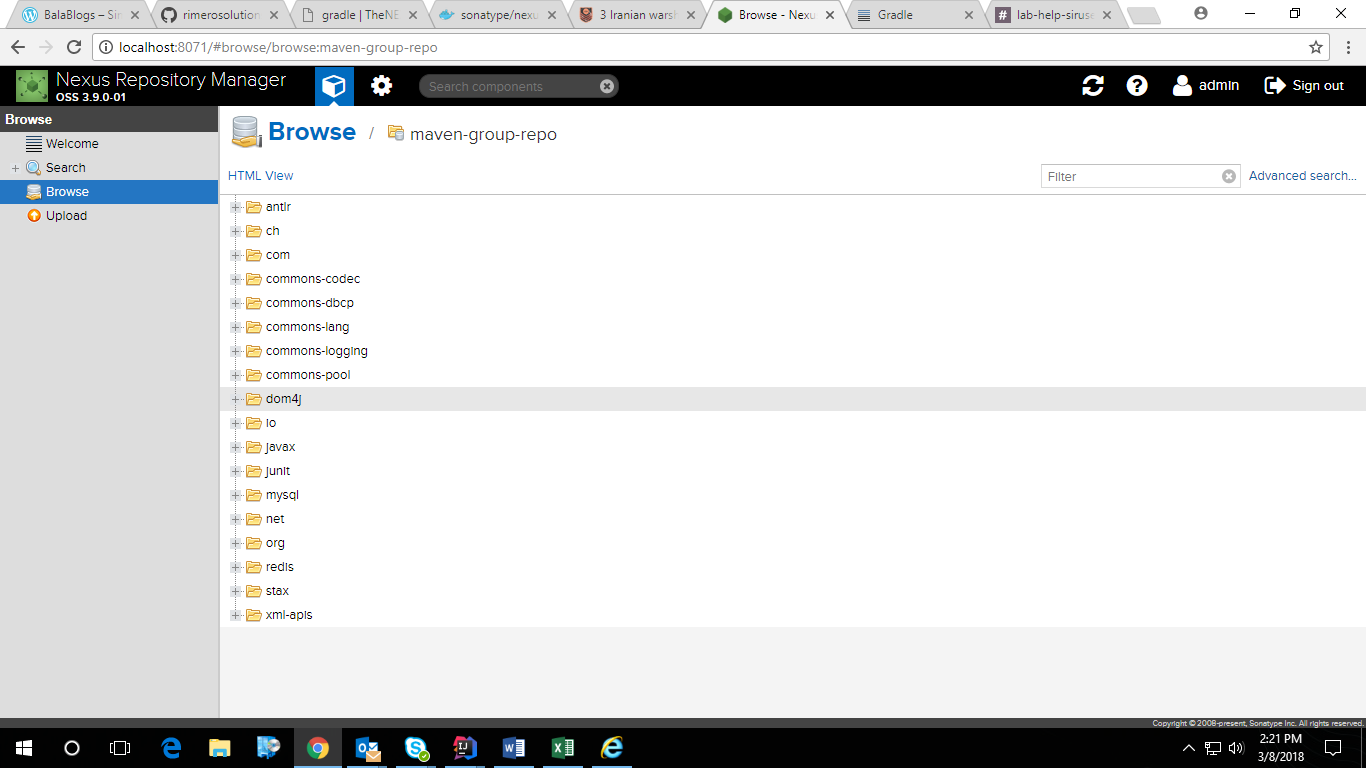


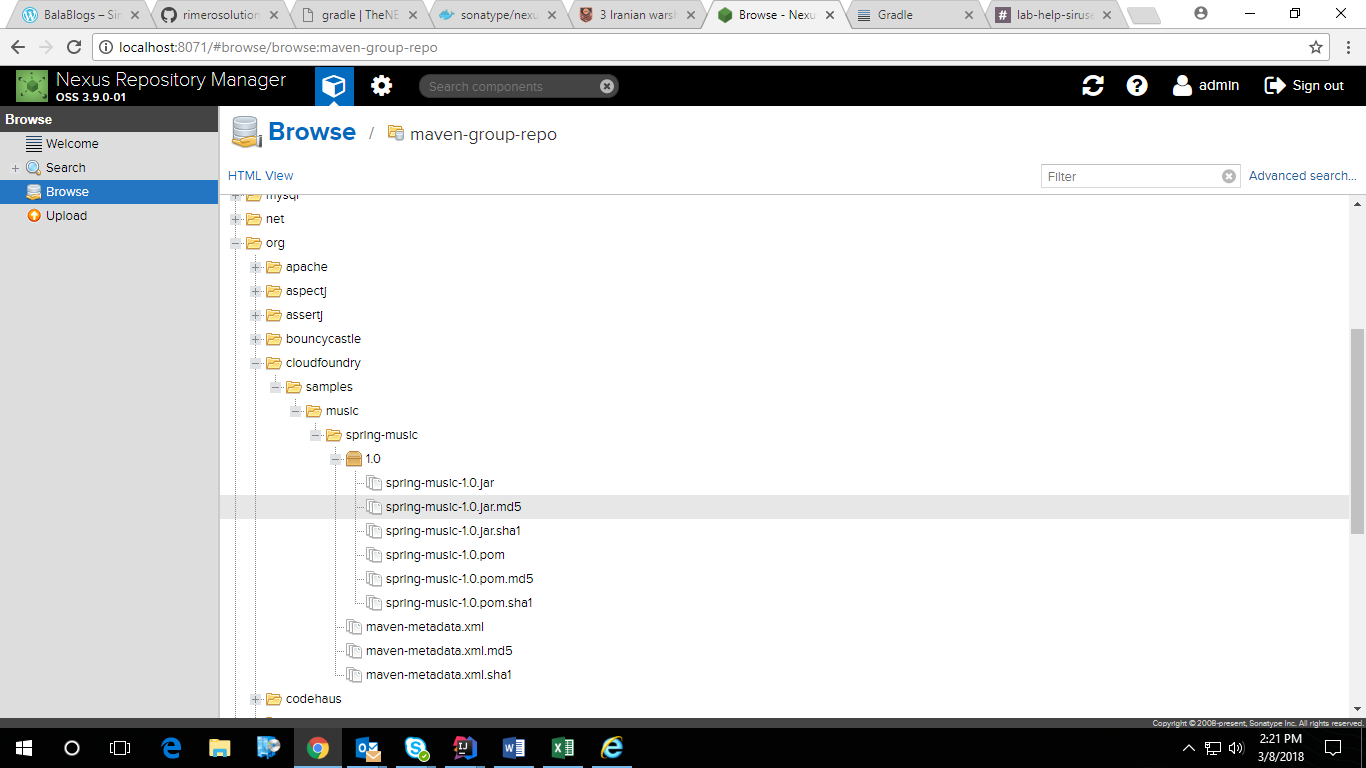
1. Create a group repository to provide all of these repos under a single URL





Step4 – Publishing the spring-music artifacts to Nexus maven repo





NOTE : Delete the files under folder .gradle/caches/module and then do

gradlew clean build –refresh-dependencies