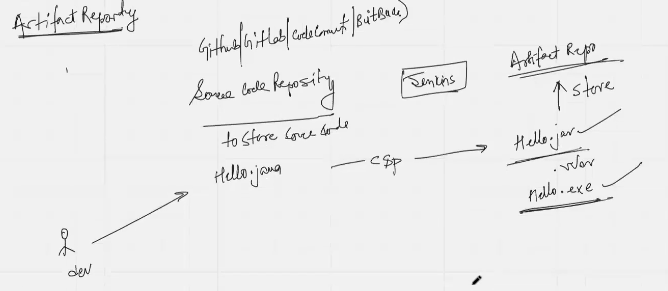
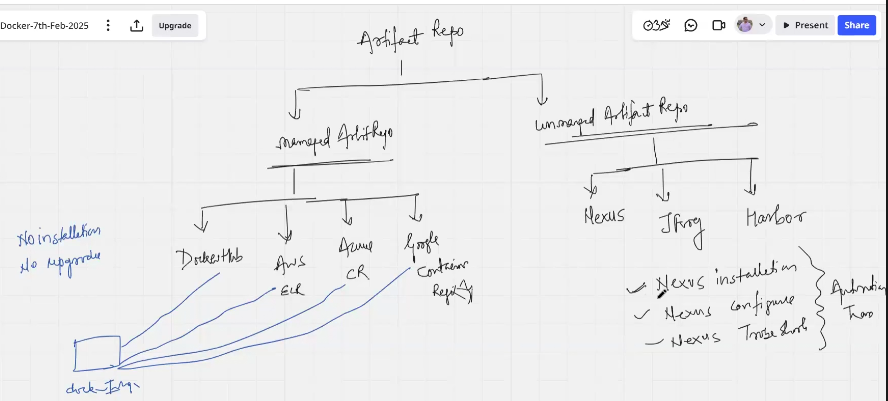
Preface to interviews

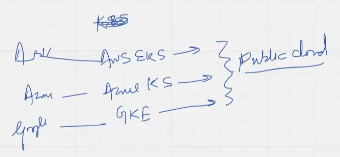
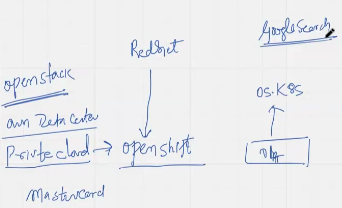
<https://snyk.io/blog/ten-git-hub-security-best-practices/>

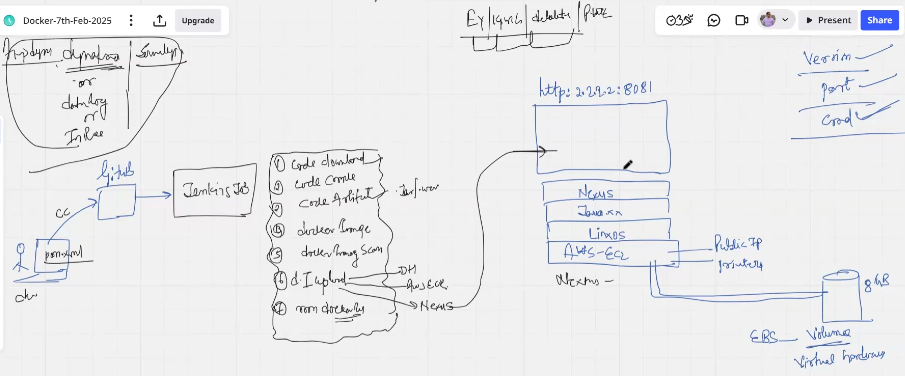
>> Always verify webhook, ip address of jenkins and ip address in servers and ip address everywhere between labs

Artefact Repository





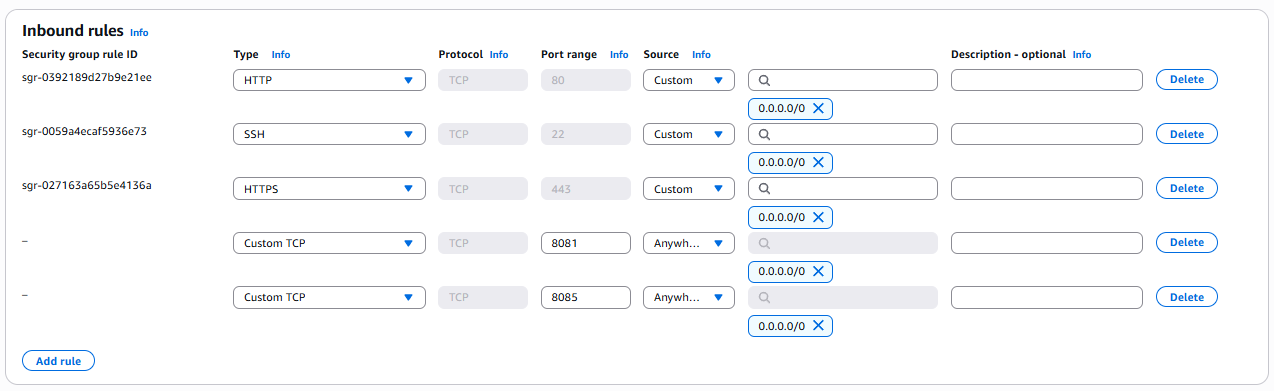
 



Steps to work with Nexus repository in Jenkins

Goto EC2 >> Launch instance >> Nexus-serverJune2025-amazonlinux-t2.medium-ping098(key) >> Create

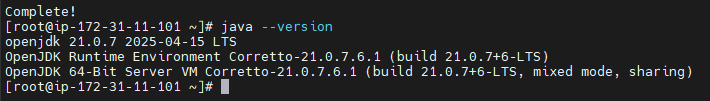
Goto Nexus-serverJune2025 >> Security >> Edit inbound rules >> Enable nexus-SG with port number open, 22, 80, 443, 8081, 8085



<https://www.redhat.com/en/topics/application-modernization/openjdk-vs-oracle-jdk>

*>> Install java version as jenkinsmaster to avoid java inconcurrency*

* sudo yum install java-21\*

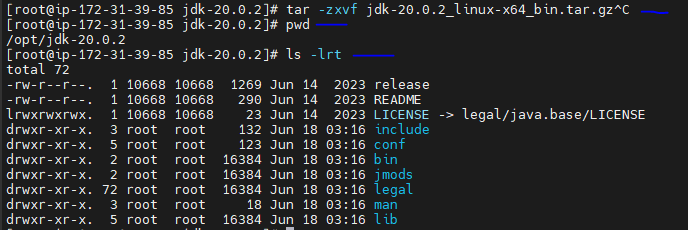


*>> Check java variables in devopsmaster for verification and create java variables in Nexus too*

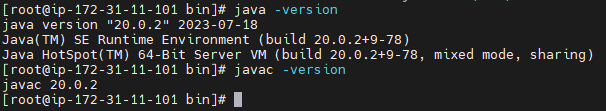
* cd /opt
* Wget -c --header "Cookie: oraclelicense=accept-securebackup-cookie" <https://download.oracle.com/java/20/archive/jdk-20.0.2_linux-x64_bin.tar.gz>

>> Downloads -rw-r--r--. 1 root root 192003505 Jun 15 2023 jdk-20.0.2\_linux-x64\_bin.tar.gz

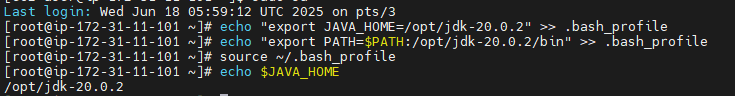
* tar -zxvf jdk-20.0.2\_linux-x64\_bin.tar.gz
* cd /opt/jdk-20.0.2
* ls –lrt



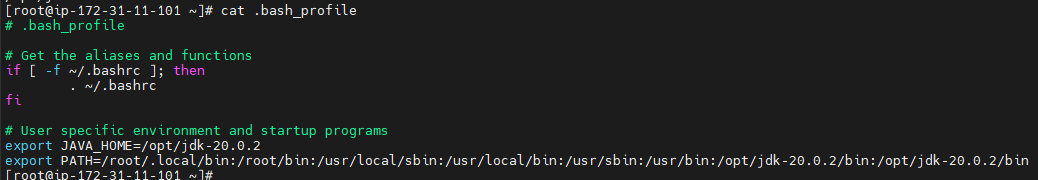
* alternatives --install /usr/bin/java java /opt/jdk-20.0.2/bin/java 2
* alternatives --config java
* alternatives --config javac
* alternatives --install /usr/bin/jar jar /opt/jdk-20.0.2/bin/jar 2
* alternatives --install /usr/bin/javac javac /opt/jdk-20.0.2/bin/javac 2
* alternatives --set jar /opt/jdk-20.0.2/bin/jar
* alternatives --set javac /opt/jdk-20.0.2/bin/javac
* cd
* echo "export JAVA\_HOME=/opt/jdk-20.0.2" >> .bash\_profile
* echo "export PATH=$PATH:/opt/jdk-20.0.2/bin" >> .bash\_profile
* java –version
* javac –version



* source ~/.bash\_profile
* echo $JAVA\_HOME



* cat .bash\_profile ……verify path being set in bash profile

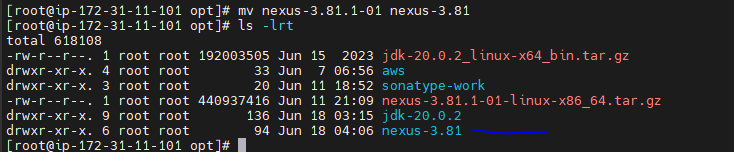


*>> Download nexus, extract it and renamed to nexus-3.81*

* wget <https://download.sonatype.com/nexus/3/nexus-3.81.1-01-linux-x86_64.tar.gz>

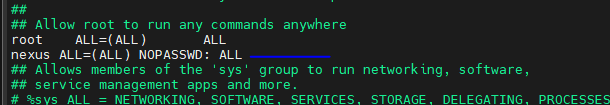
>> Downloads -rw-r--r--. 1 root root 440937416 Jun 11 21:09 nexus-3.81.1-01-linux-x86\_64.tar.gz

* tar -zxvf nexus-3.81.1-01-linux-x86\_64.tar.gz
* mv /opt/nexus-3.81.1-01 /opt/nexus-3.81



*>> Add the user nexus and give root privilege*

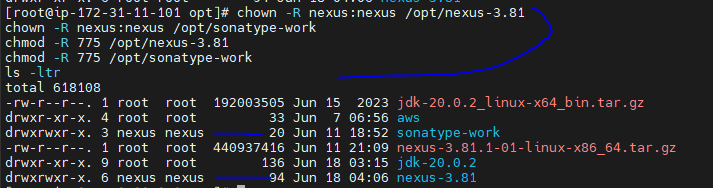
* useradd nexus
* visudo
* nexus ALL=(ALL) NOPASSWD: ALL



* id nexus

*>> Change the ownership and permissions recursively for the directories nexus-3.81 and sonatype-work*

* chown -R nexus:nexus /opt/nexus-3.81
* chown -R nexus:nexus /opt/sonatype-work
* chmod -R 775 /opt/nexus-3.81
* chmod -R 775 /opt/sonatype-work
* ls –ltr



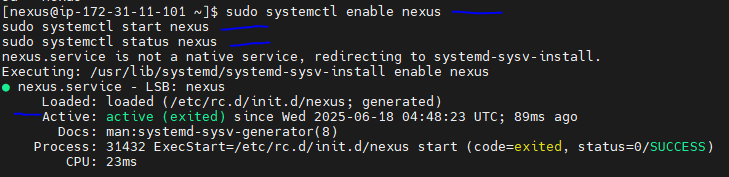
*>> Open /opt/nexus-3.81/bin/nexus.rc file and uncomment run\_as\_user*

* vi /opt/nexus-3.81/bin/nexus.rc
* run\_as\_user="nexus"
* :wq!

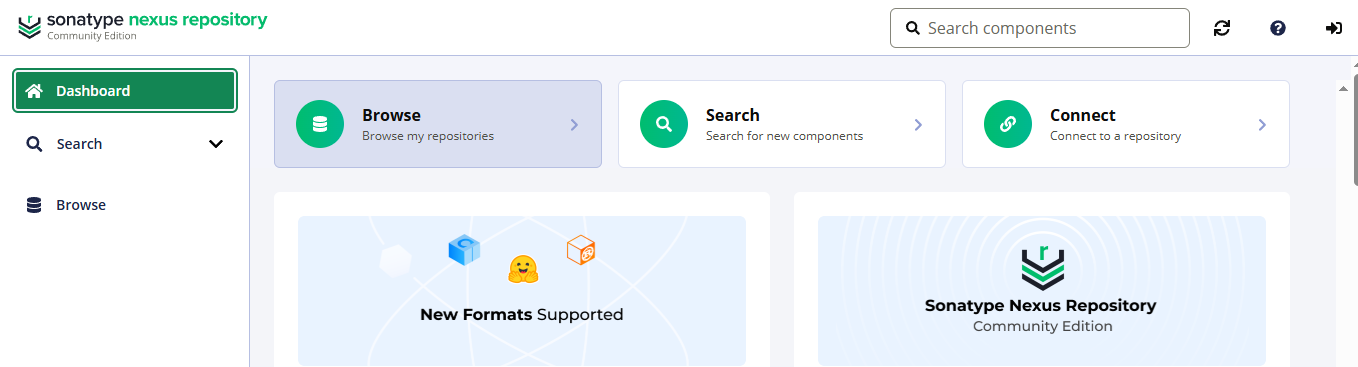


*>> create nexus as a service (to run the application as nexus user)*

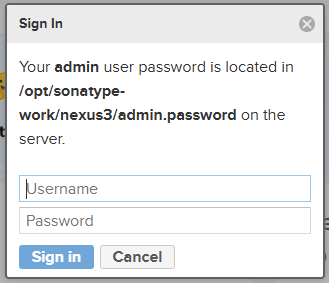
* ln -s /opt/nexus-3.81/bin/nexus /etc/init.d/nexus
* su – nexus
* whoami
* sudo systemctl enable nexus
* sudo systemctl start nexus
* sudo systemctl status nexus



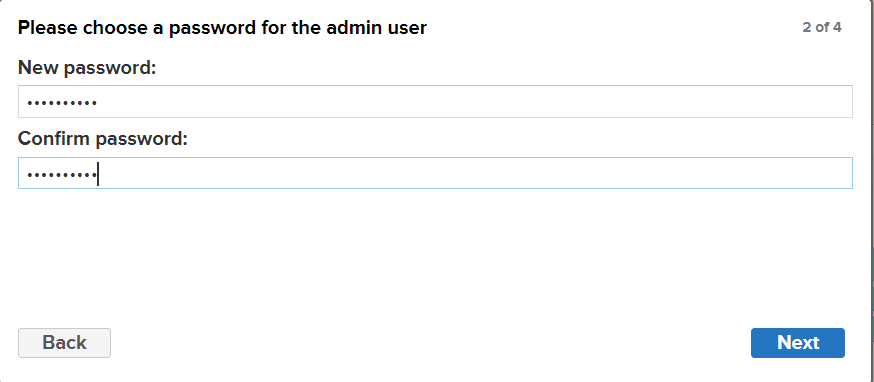
*Go to browser and type publicip:8081*



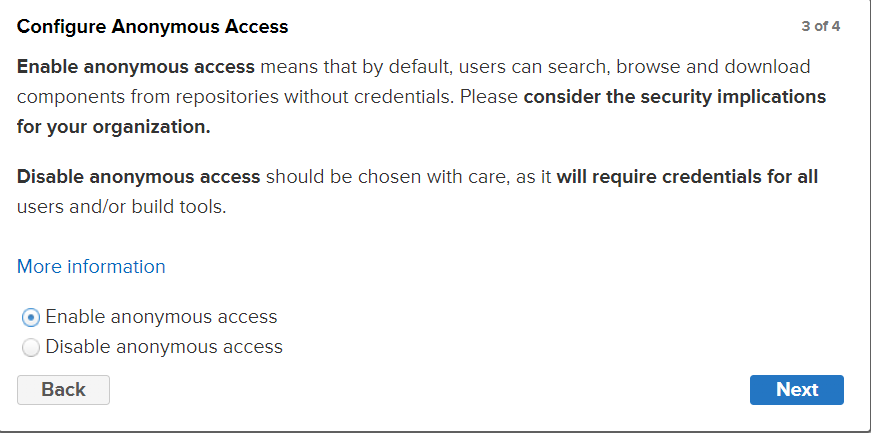
Click on sign-in



Click on next and give new password=nexus@123



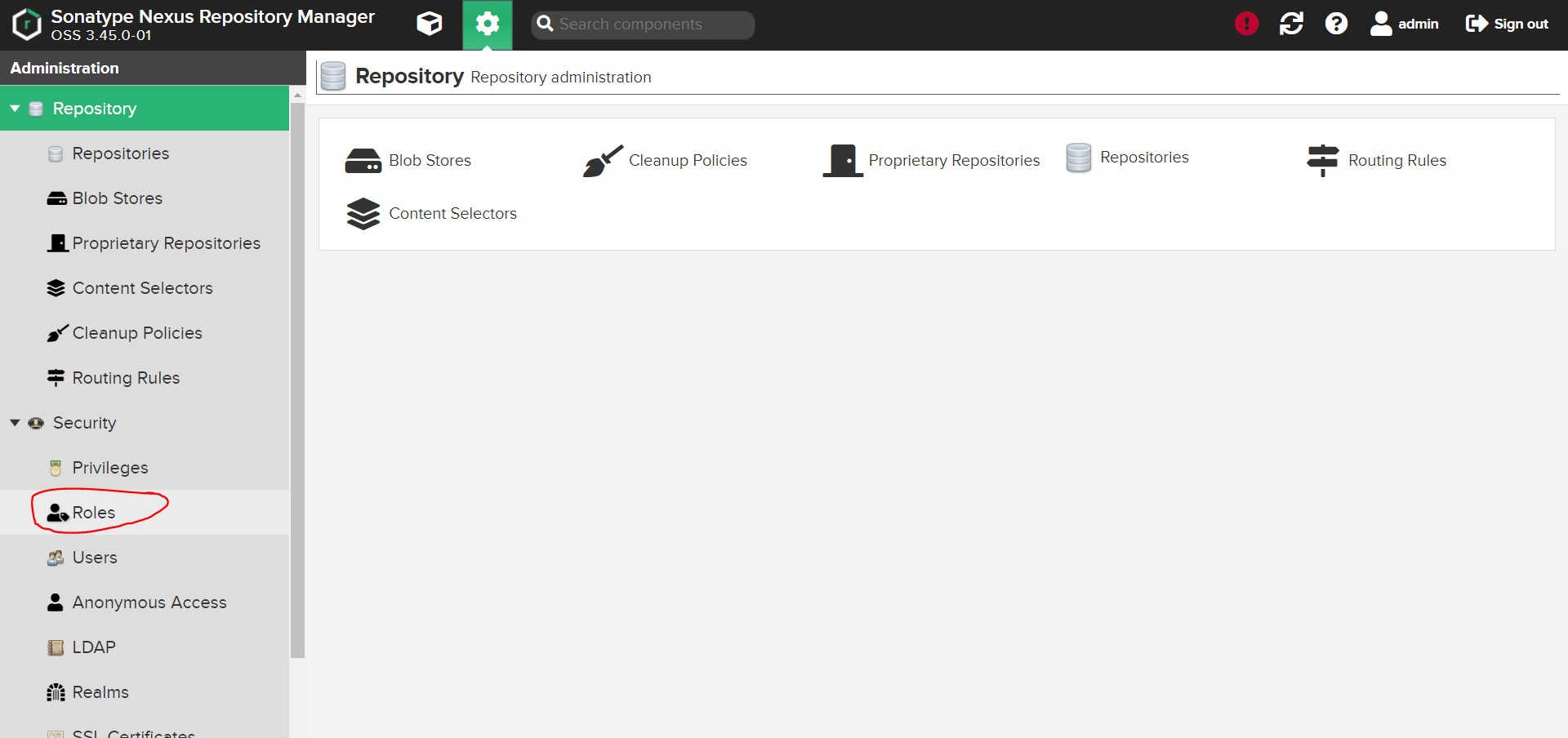
Click on enable anonymous access



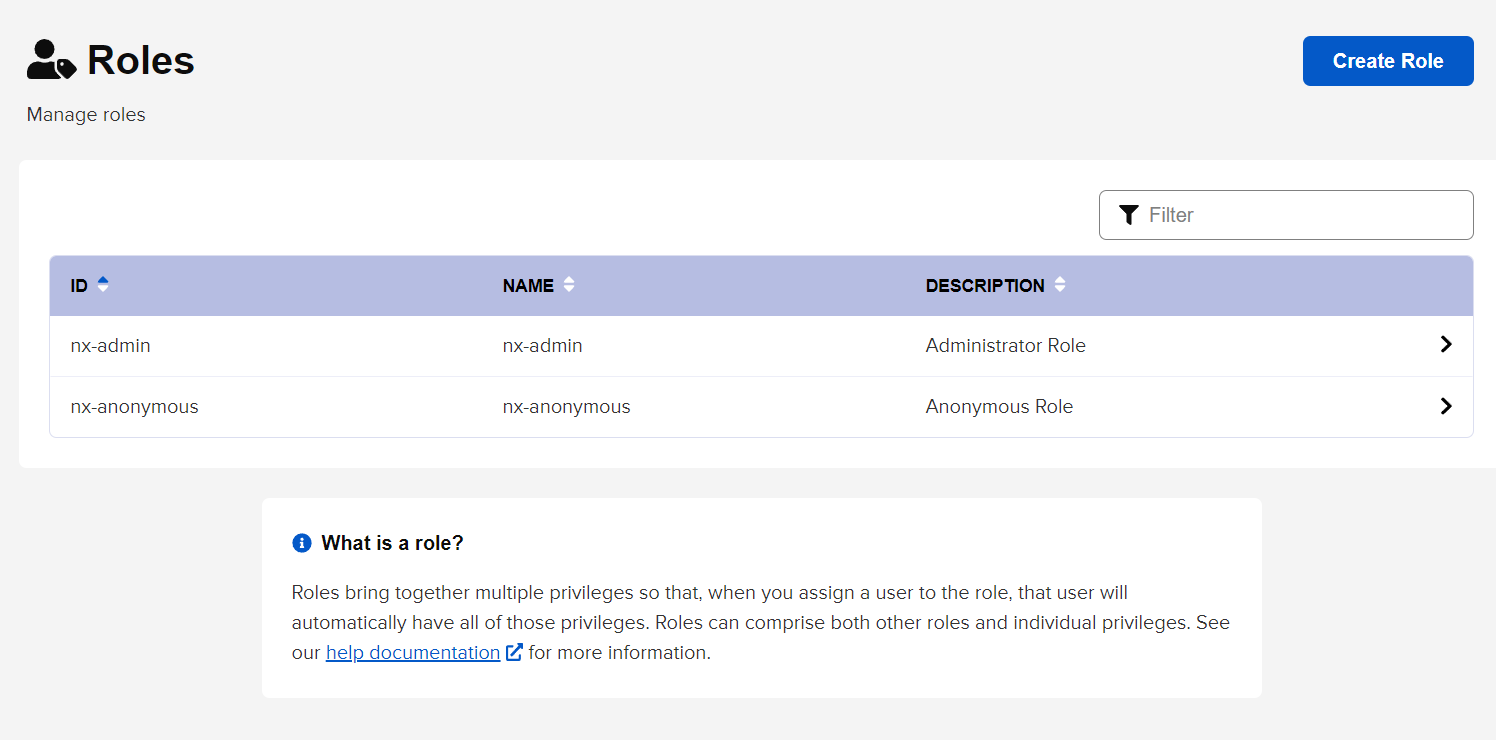
Then click on next and finish.

Nexus is normally integrated with LDAP servers so we can add the roles in nexus and assign to the users.

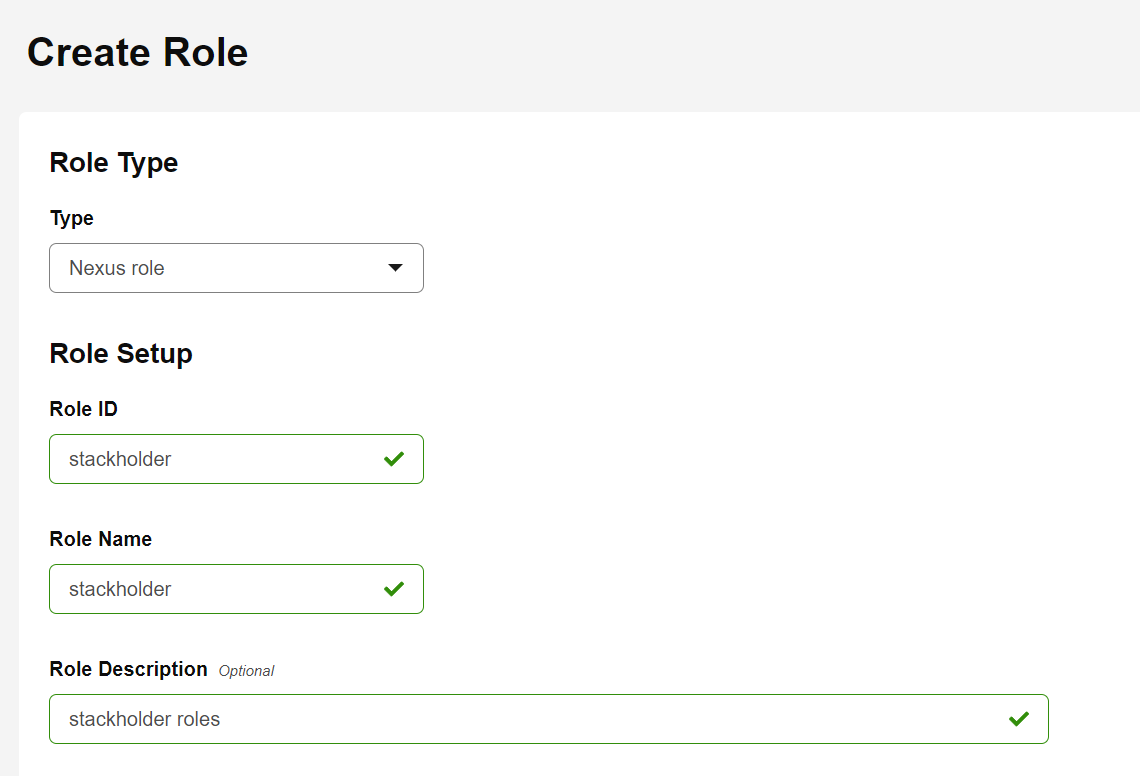
Go to server administration and configuration and click on the roles

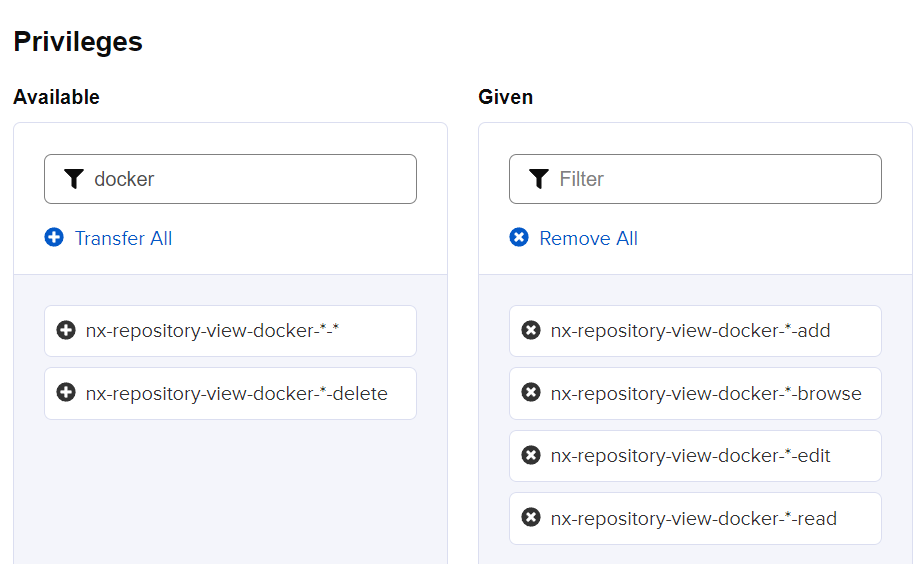


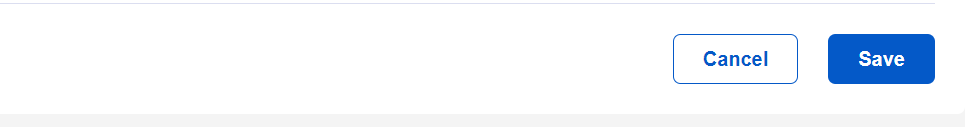
Click on create a role



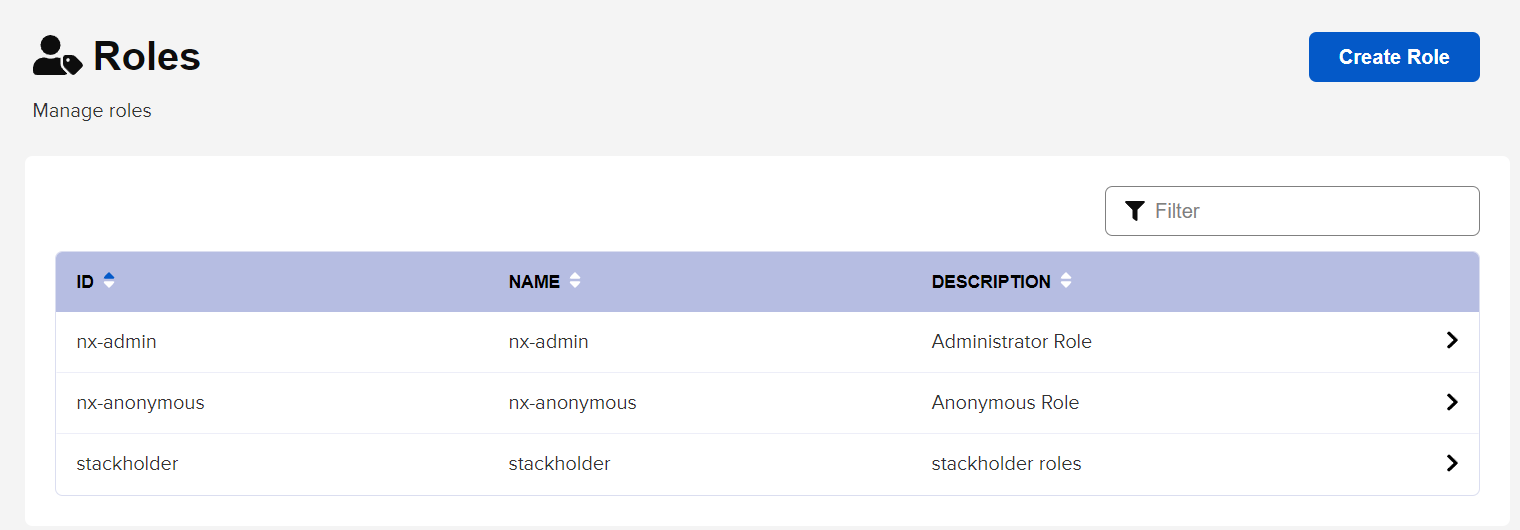
Add a role for an example. Give some docker privileges and click on save to create the role.



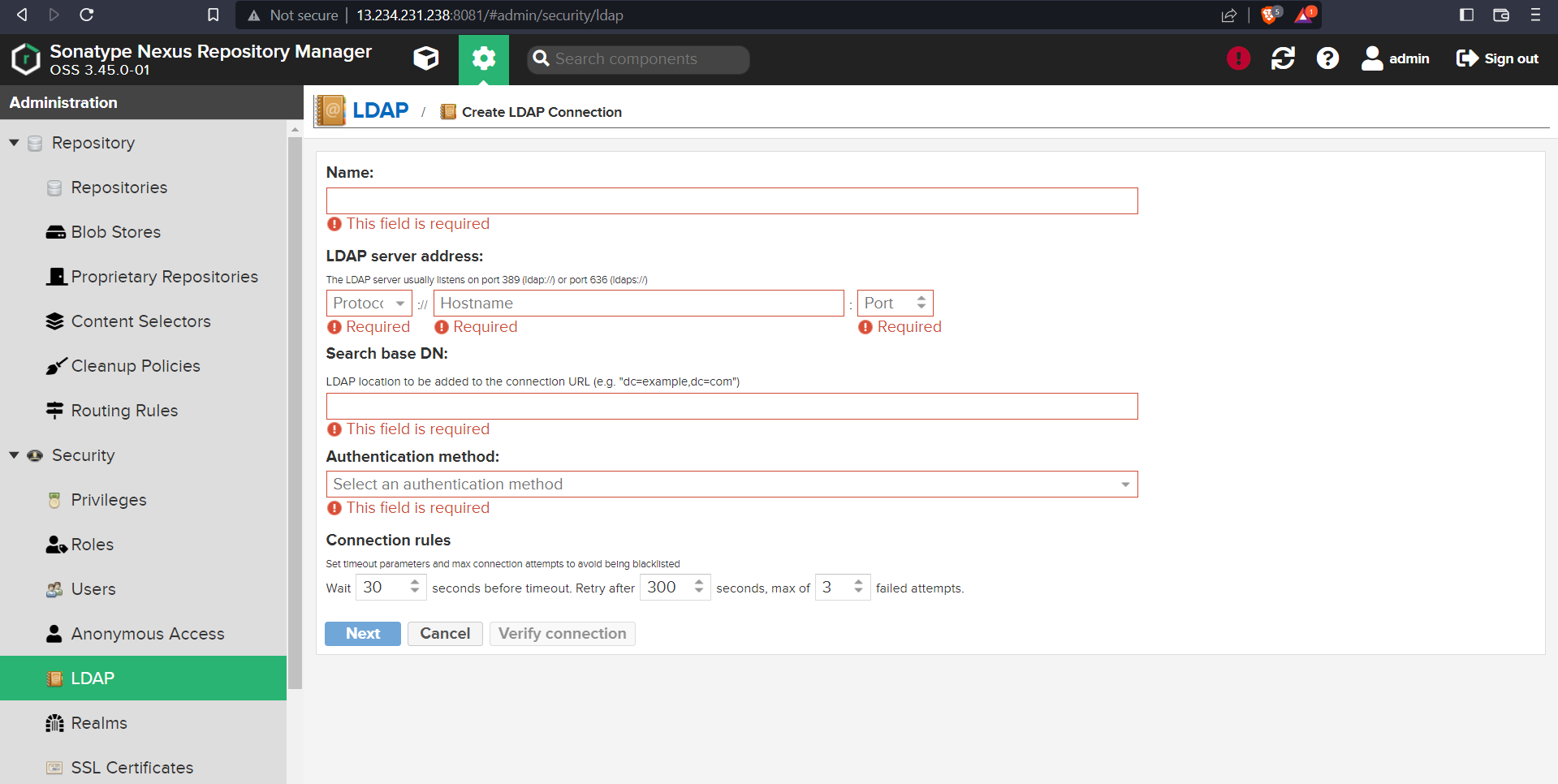




Now you can see the role has been added



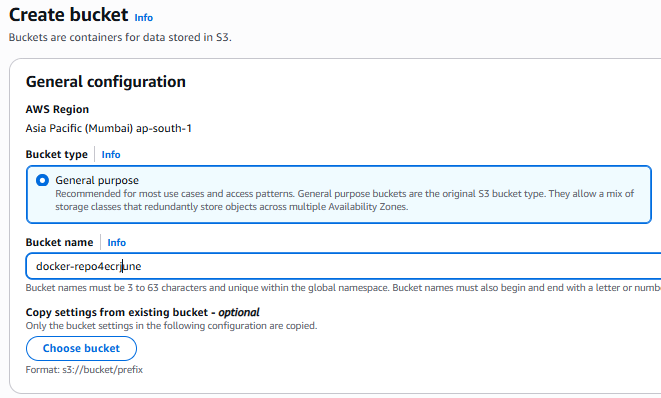
If we are integrated with LDAP server, we can add the LDAP connection ad add the roles to users



Currently we are not using LDAP server.

>> Goto AWS S3 and click on create bucket

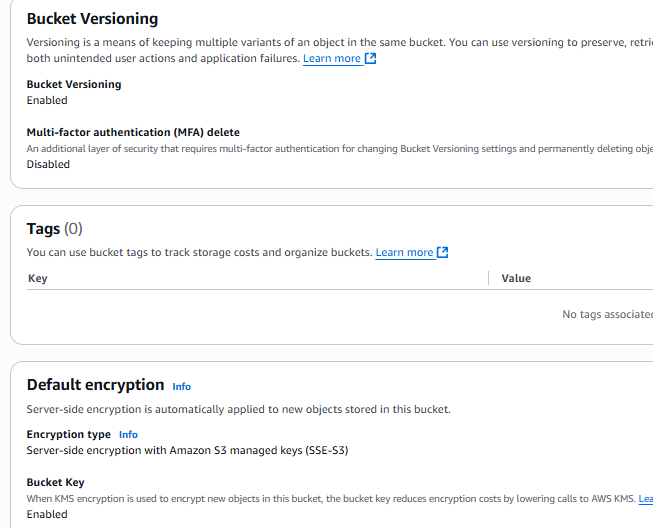
>> Enter the Bucket name, Region, and select ACL enabled



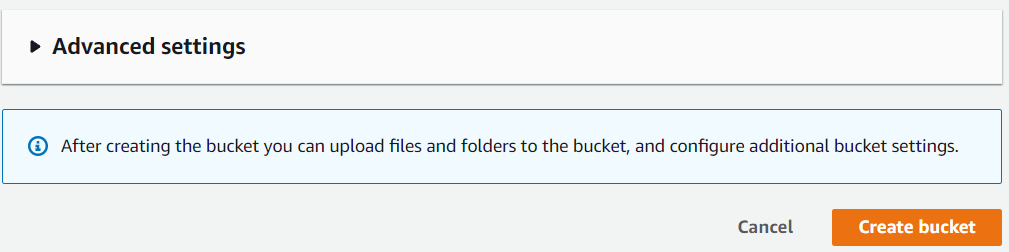
Untick block all public access and tick on I acknowledge that the current settings might result in this bucket and the objects within becoming public.



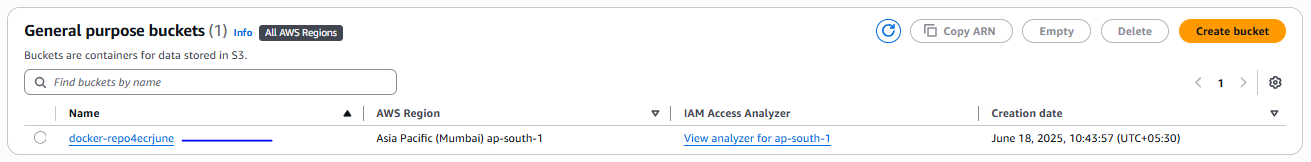
Select bucket versioning enable



Then click on create bucket.

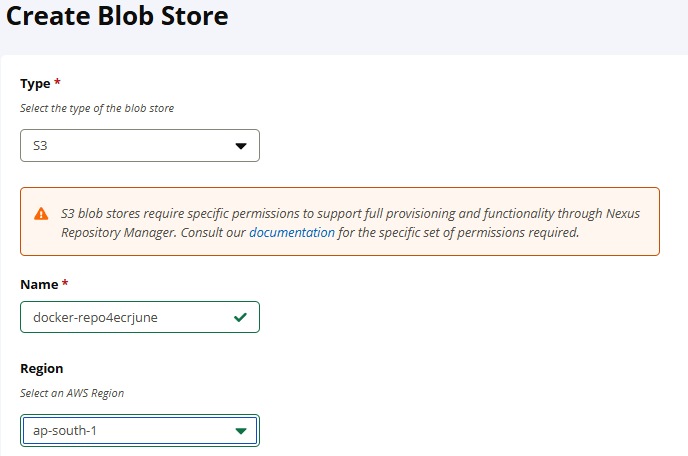


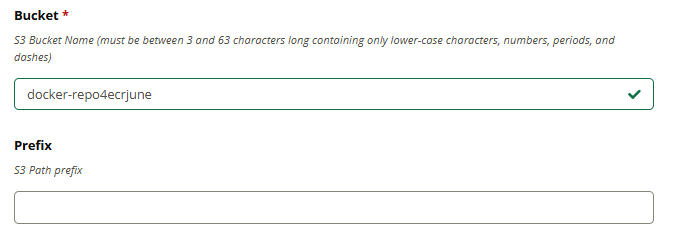
Now you can see the S3 bucket has been created



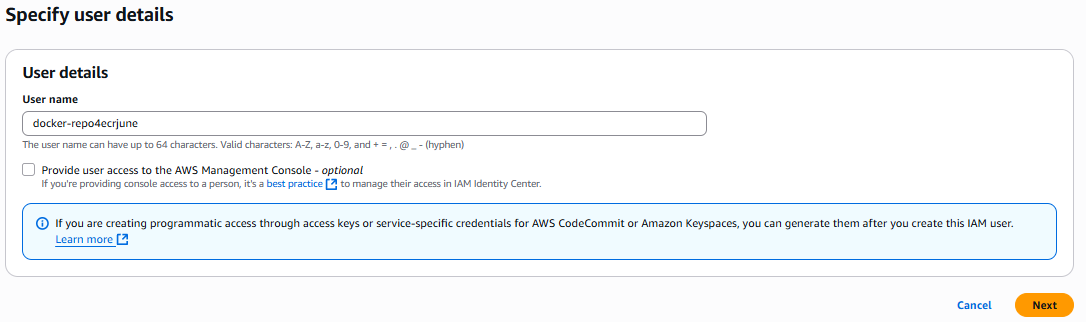
>> Go to nexus >> settings >> Repository >> Blob stores

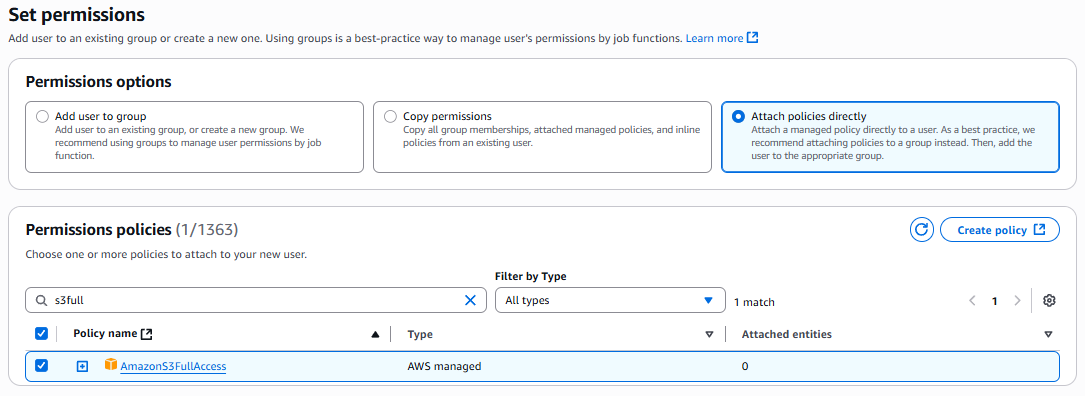
Click on create blob store

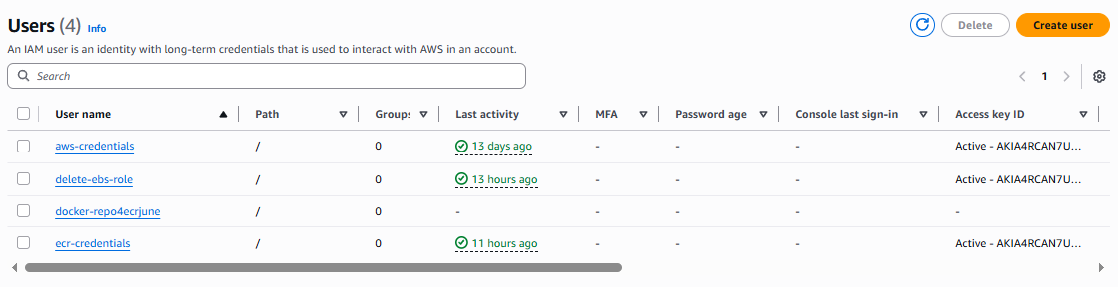




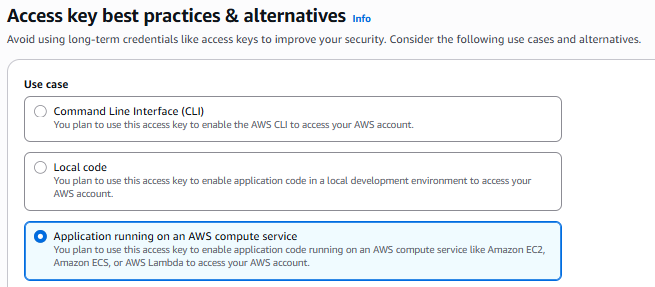
>> Goto AWS IAM and create a new user with amazonS3fullaccess



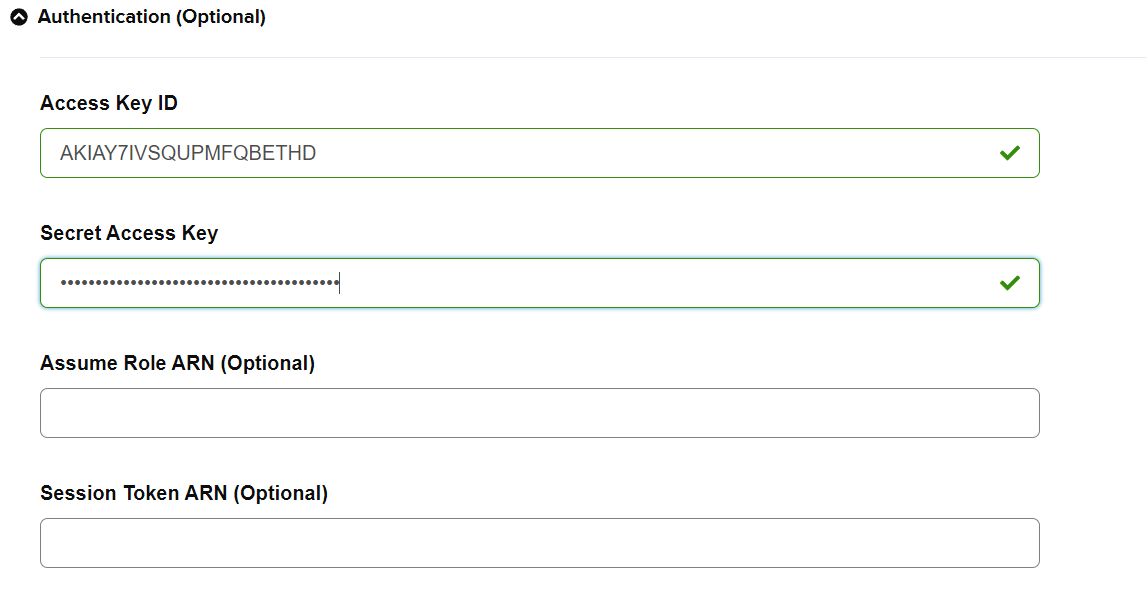




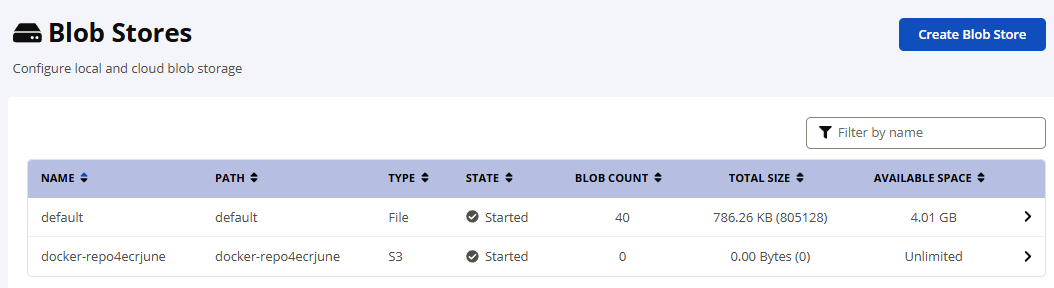
>> Create access key and secret access key



>> Copy the access key and secret access key to nexus

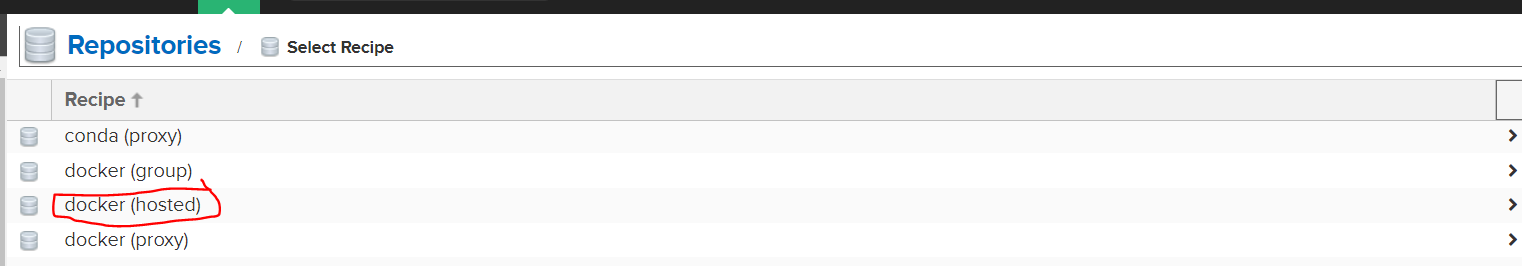


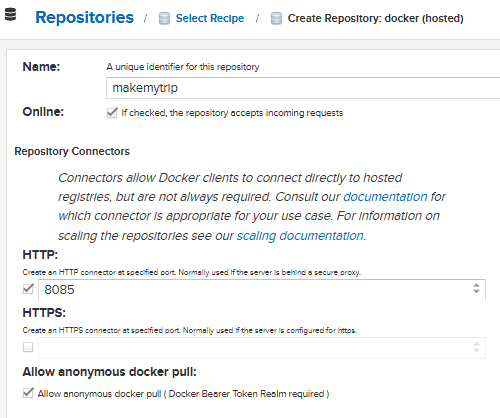
>> Then Click on Save.

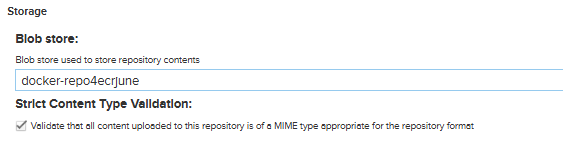


Goto Nexus >> Repositories >> Create repositories

Click on docker (hosted)

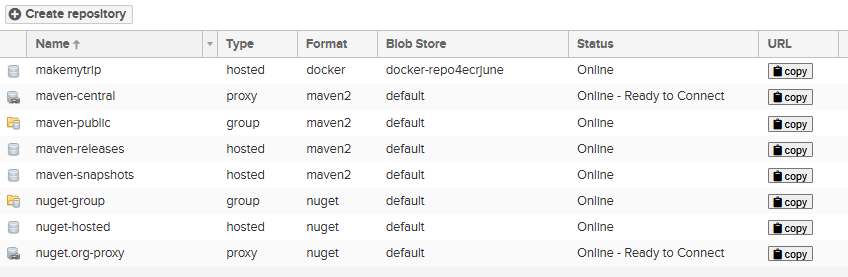




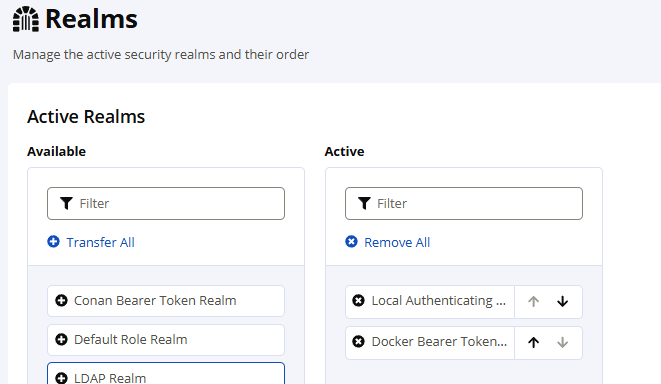




Then click on Create Repository

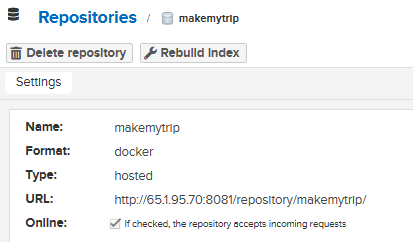


>>Goto realms >> Activate >> Local Authenticating realm >> Local authorizing realm >> Docker bearer token realm

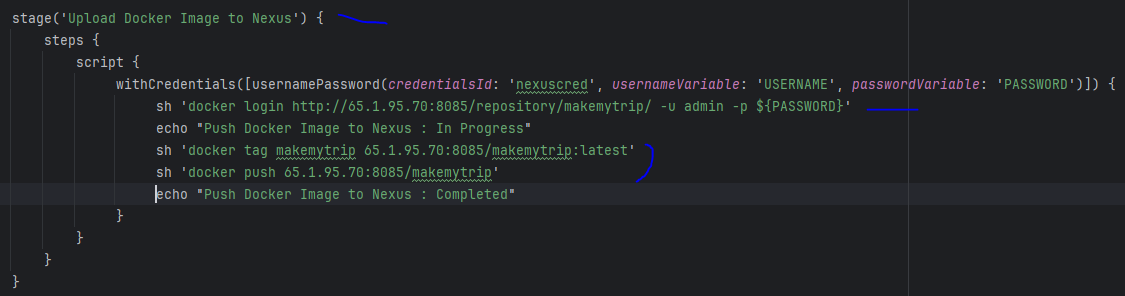


>> Save

>> Go to repositories and open yatra to get the details of yatra repository

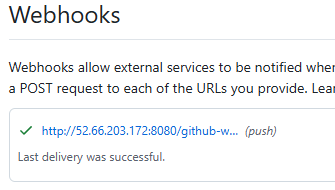
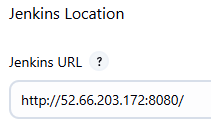


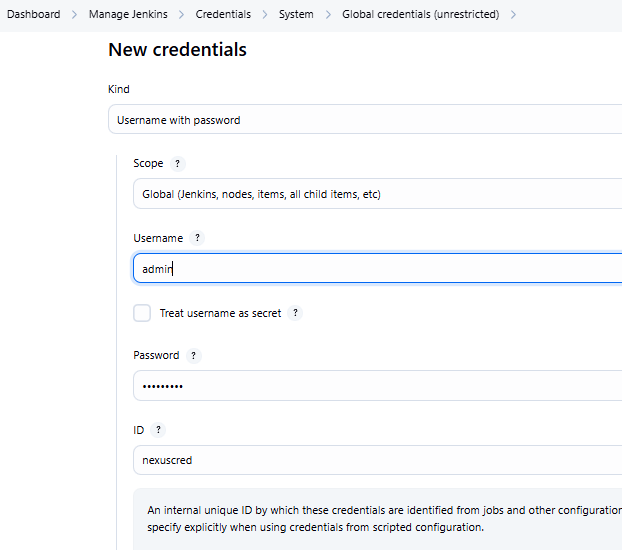
>> Edit the Jenkins file by adding the URL with the port number 8085.



Go to jenkins console and add the nexus credential. Ensure the credential ID mentioned here and in Jenkins file are same.

Update webhook and jenkins location

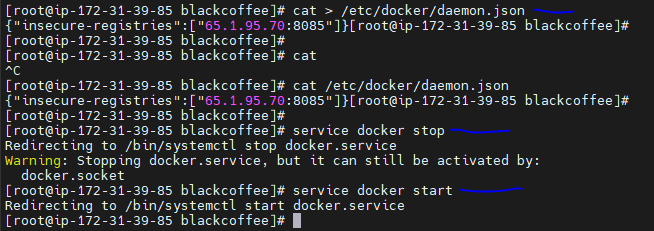


*>> Go to Jenkins Master terminal >> Add the file daemon.json and enter the nexus ip.*

* cat > /etc/docker/daemon.json

{ "insecure-registries" : [ " 65.2.150.176:8085" ] }

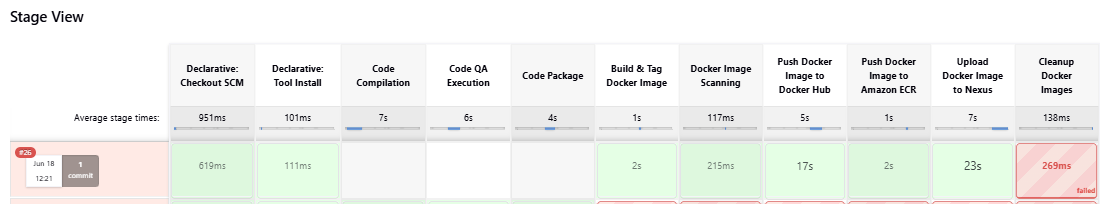
* Ctrl+d
* Service docker stop
* Service docker start



Push to git hub.

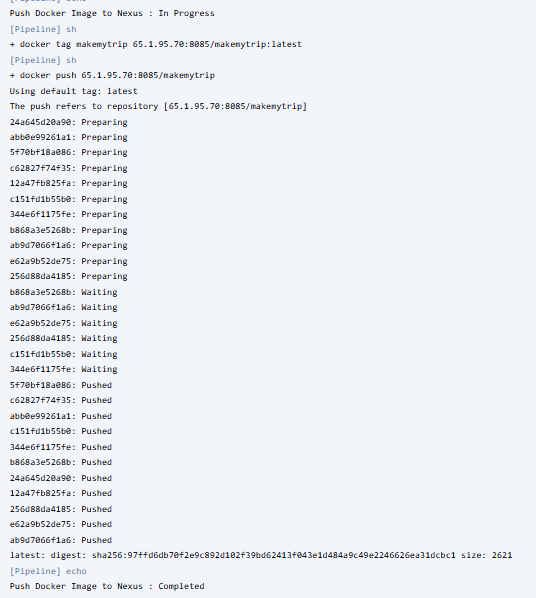
* Git add –all
* Git commit -m “ coderefactor”
* Git push origin dev

Go to Jenkins dash board and run the project yatra.

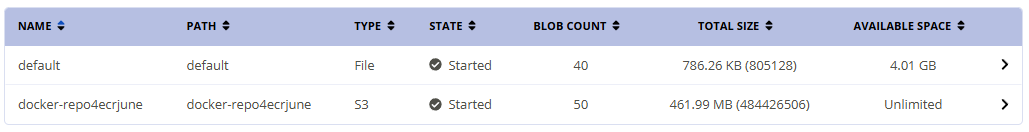


We can see the job was success until the image upload to nexus.

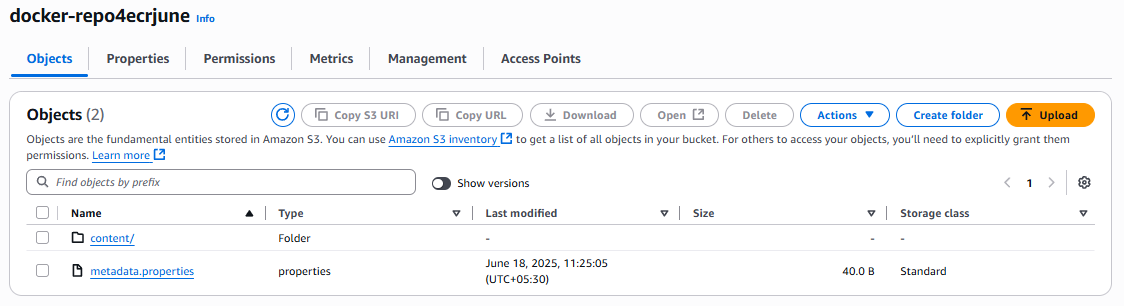
We can see the logs in console output.



You can check the blob store in nexus repository. There we can see the size was changed.



You can go to amazon s3 and view the files that we uploaded to nexus repository.



Click on the content and you can see the uploads.

