**Create a nexus-SG with port number open , 22,80,443,8081,8085**

**create an instance with t2.medium/t2.small**

#Install java 1.8

sudo yum install java-1.8.0-openjdk.x86\_64 -y

or

cd /opt

wget wget -c --header "Cookie: oraclelicense=accept-securebackup-cookie" <http://download.oracle.com/otn-pub/java/jdk/8u131-b11/d54c1d3a095b4ff2b6607d096fa80163/jdk-8u131-linux-x64.tar.gz>

tar -zxvf jdk-8u131-linux-x64.tar.gz

ls -ltr

ls -ltr

cd /opt/jdk1.8.0\_131

pwd

alternatives --install /usr/bin/java java /opt/jdk1.8.0\_131/bin/java 2

alternatives --config java

alternatives --install /usr/bin/jar jar /opt/jdk1.8.0\_131/bin/jar 2

alternatives --install /usr/bin/javac javac /opt/jdk1.8.0\_131/bin/javac 2

alternatives --set jar /opt/jdk1.8.0\_131/bin/jar

alternatives --set javac /opt/jdk1.8.0\_131/bin/javac

java -version

javac -version

cd

echo "export JAVA\_HOME=/opt/jdk1.8.0\_131" >> .bash\_profile

echo "export JRE\_HOME=/opt/jdk1.8.0\_131/jre" >> .bash\_profile

echo "export PATH=$PATH:/opt/jdk1.8.0\_131/bin:/opt/jdk1.8.0\_131/jre/bin" >> .bash\_profile

source ~/.bash\_profile

echo $JAVA\_HOME

#Download nexus, extract it and renamed to nexus-3.45

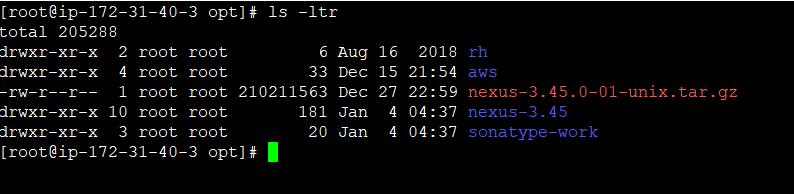
cd /opt

wget https://download.sonatype.com/nexus/3/nexus-3.45.0-01-unix.tar.gz

tar -zxvf nexus-3.45.0-01-unix.tar.gz

mv /opt/nexus-3.45.0-01 /opt/nexus-3.45

ls -ltr

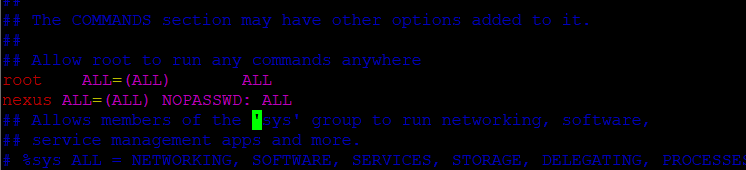


#Add the user nexus and give root privilege

useradd nexus

visudo

nexus ALL=(ALL) NOPASSWD: ALL



id nexus

#Change the ownership and permissions recursively for the directories nexus-3.45 and sonatype-work

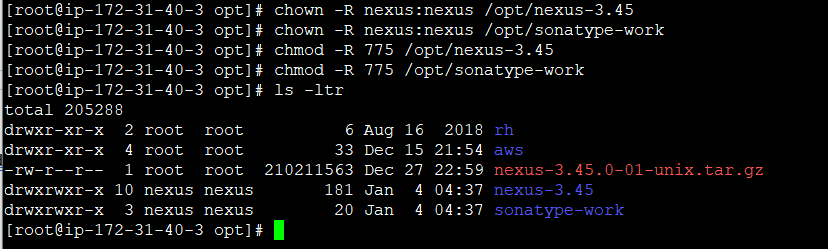
chown -R nexus:nexus /opt/nexus-3.45

chown -R nexus:nexus /opt/sonatype-work

chmod -R 775 /opt/nexus-3.45

chmod -R 775 /opt/sonatype-work

ls -ltr



#open /opt/nexus-3.45/bin/nexus.rc fine and uncomment run\_as\_user

vi /opt/nexus-3.45/bin/nexus.rc

run\_as\_user="nexus"

:wq!

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

ln -s /opt/nexus-3.45/bin/nexus /etc/init.d/nexus

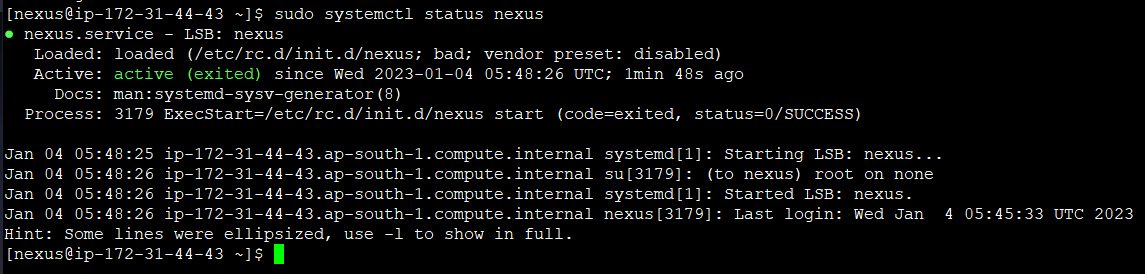
su - nexus

whoami

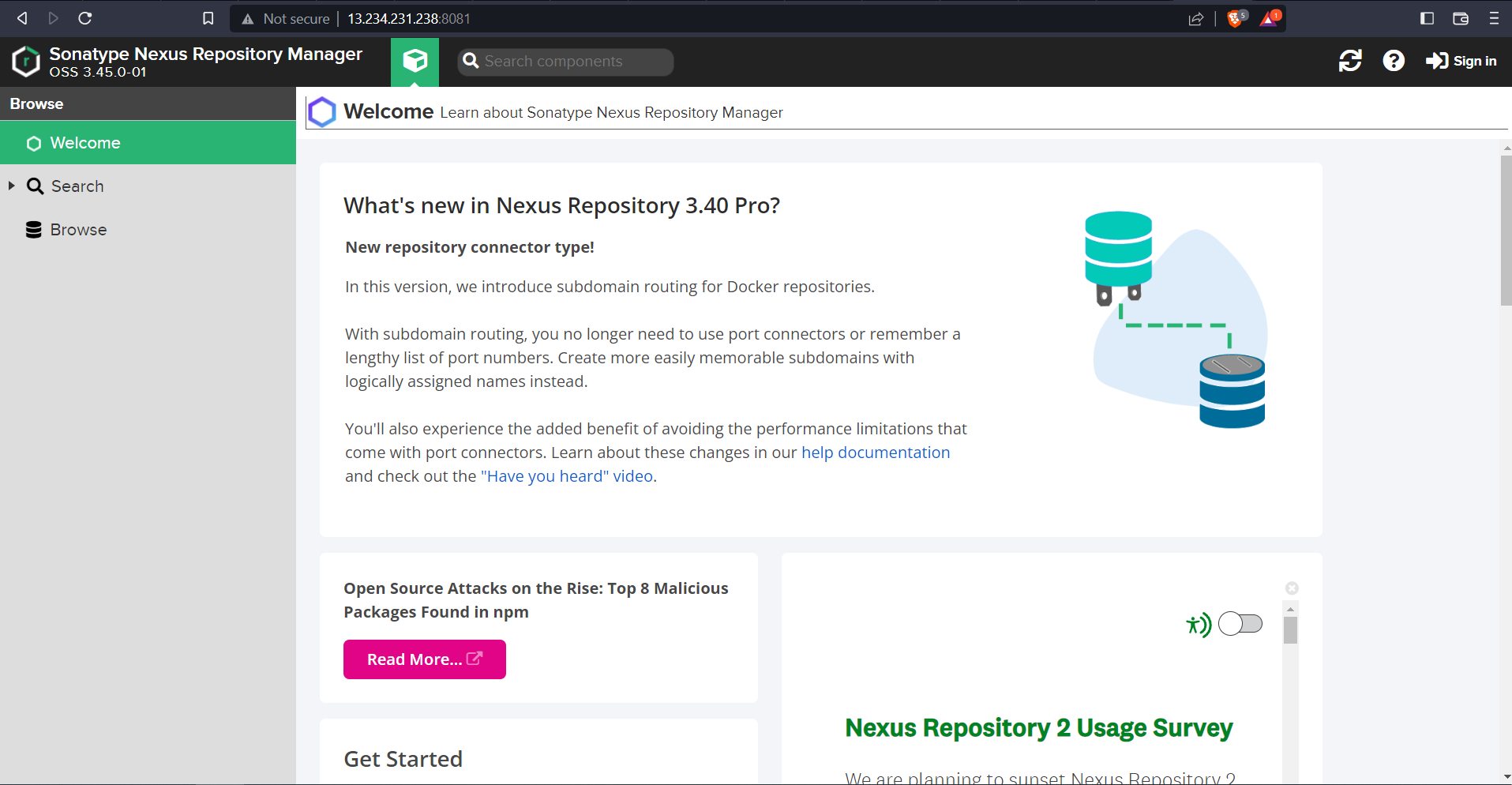
sudo systemctl enable nexus

sudo systemctl start nexus

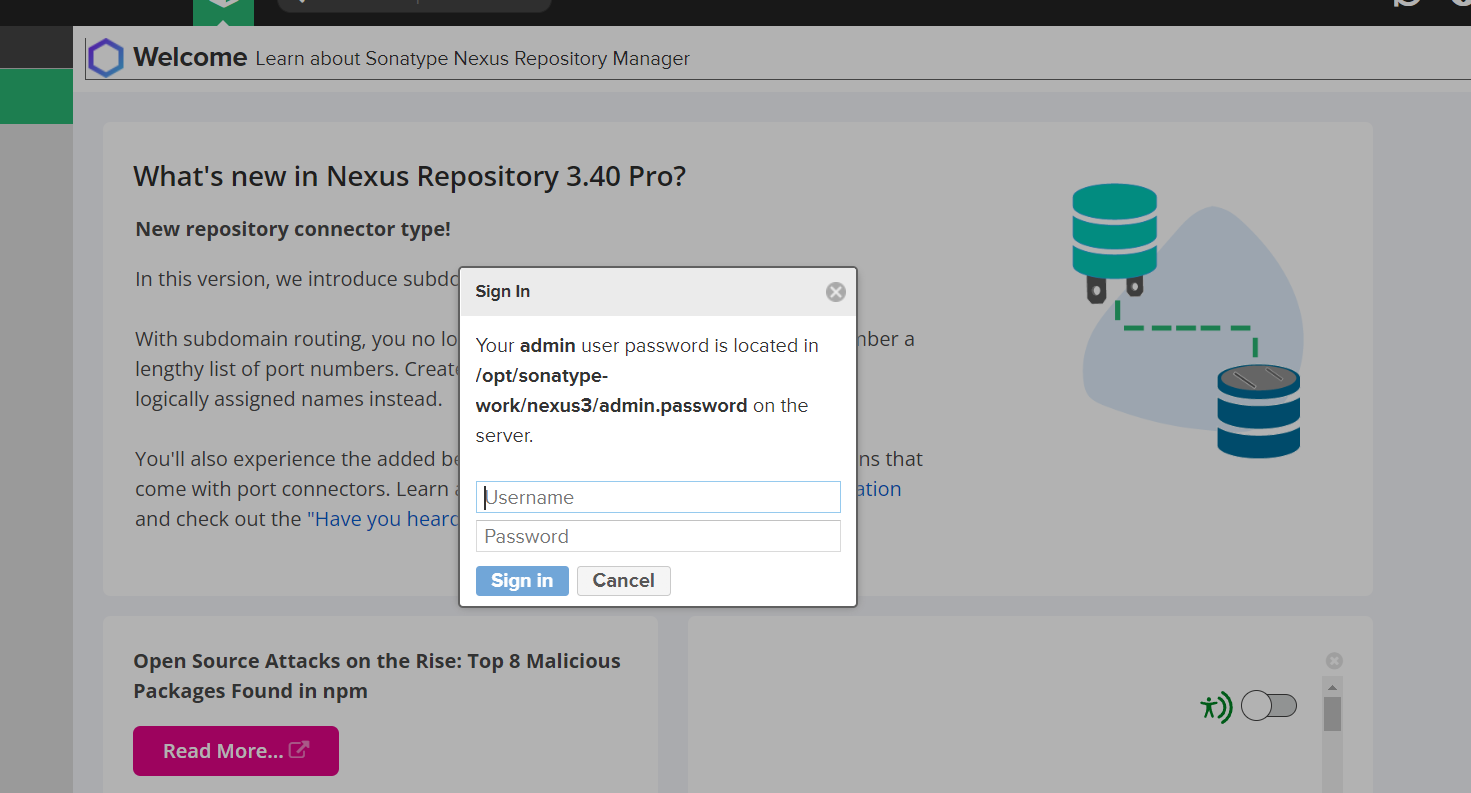
sudo systemctl status nexus



Go to browser and type publicip:8081

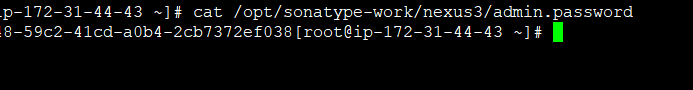


Click on Sign in

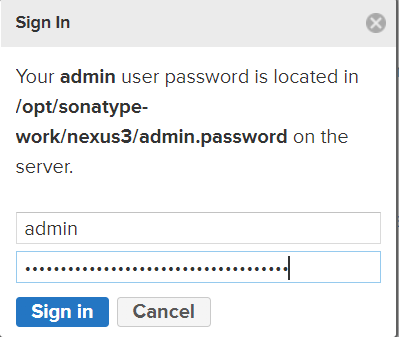


Go to instance to get the password

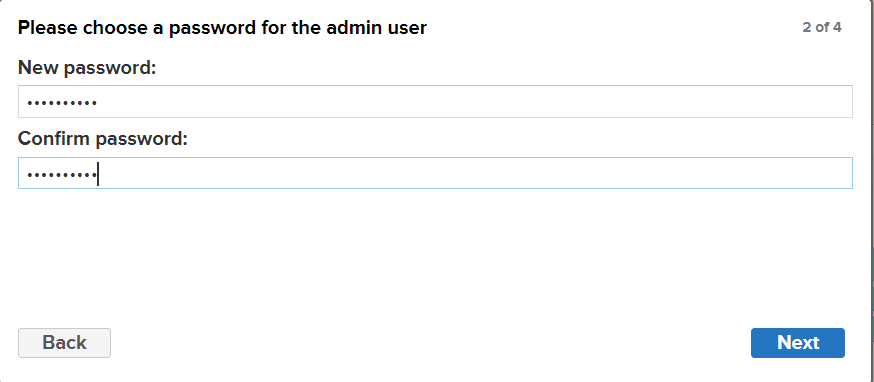
cat /opt/sonatype-work/nexus3/admin.password



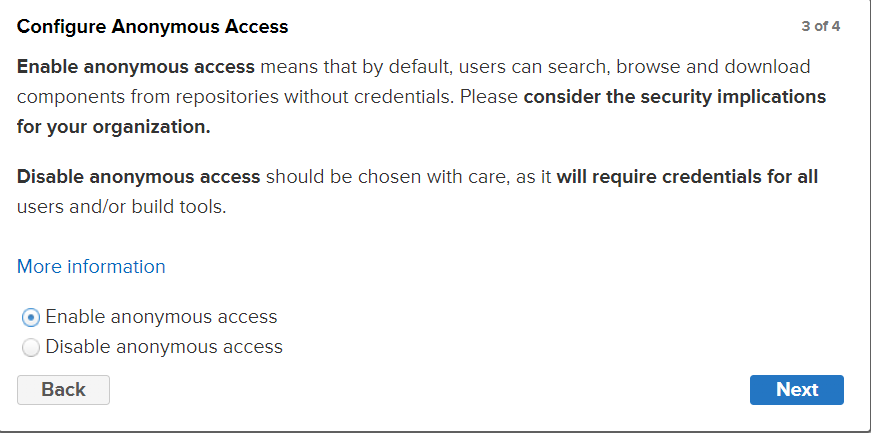
Type the username and password and login



Click on next and give new password



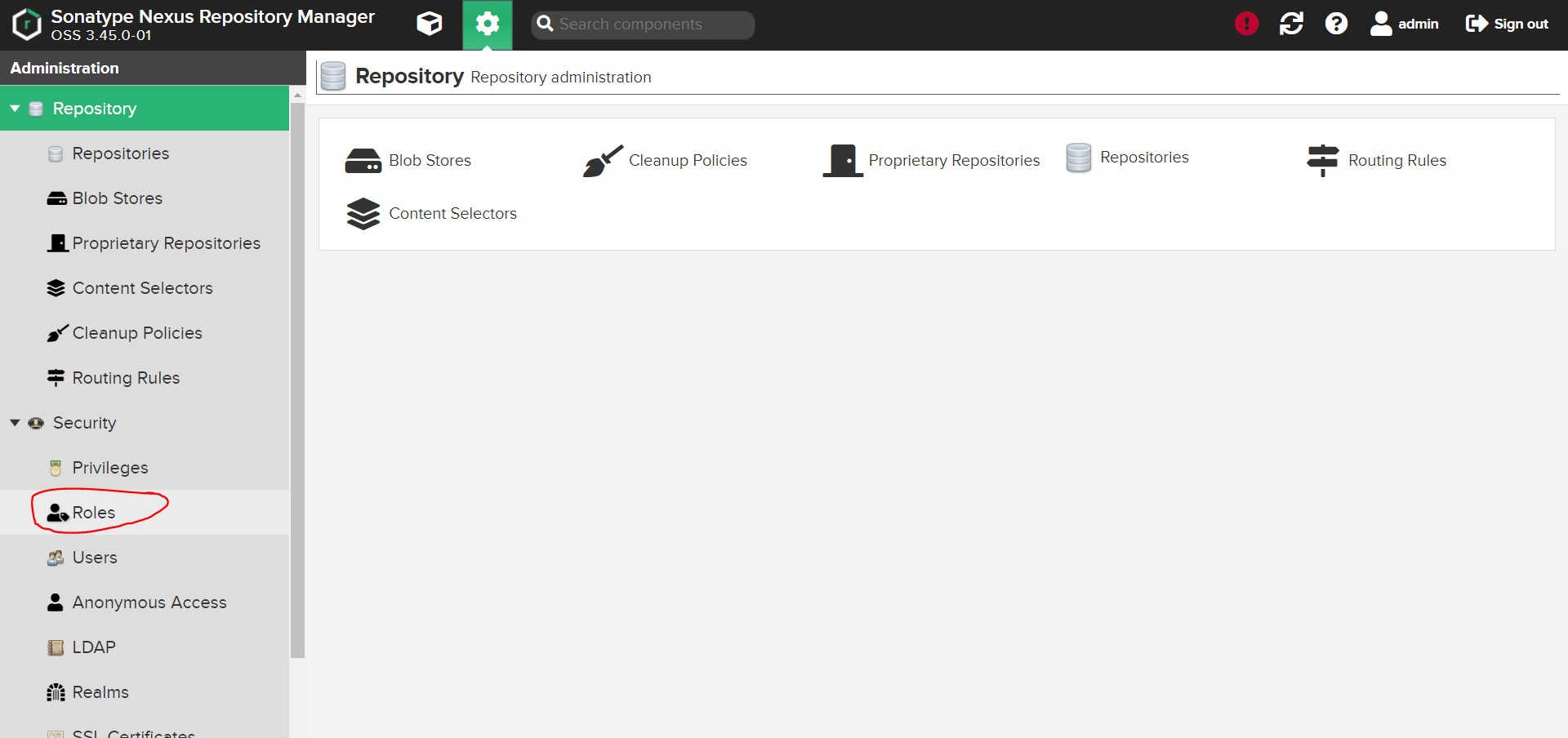
Click on enable anonymous access



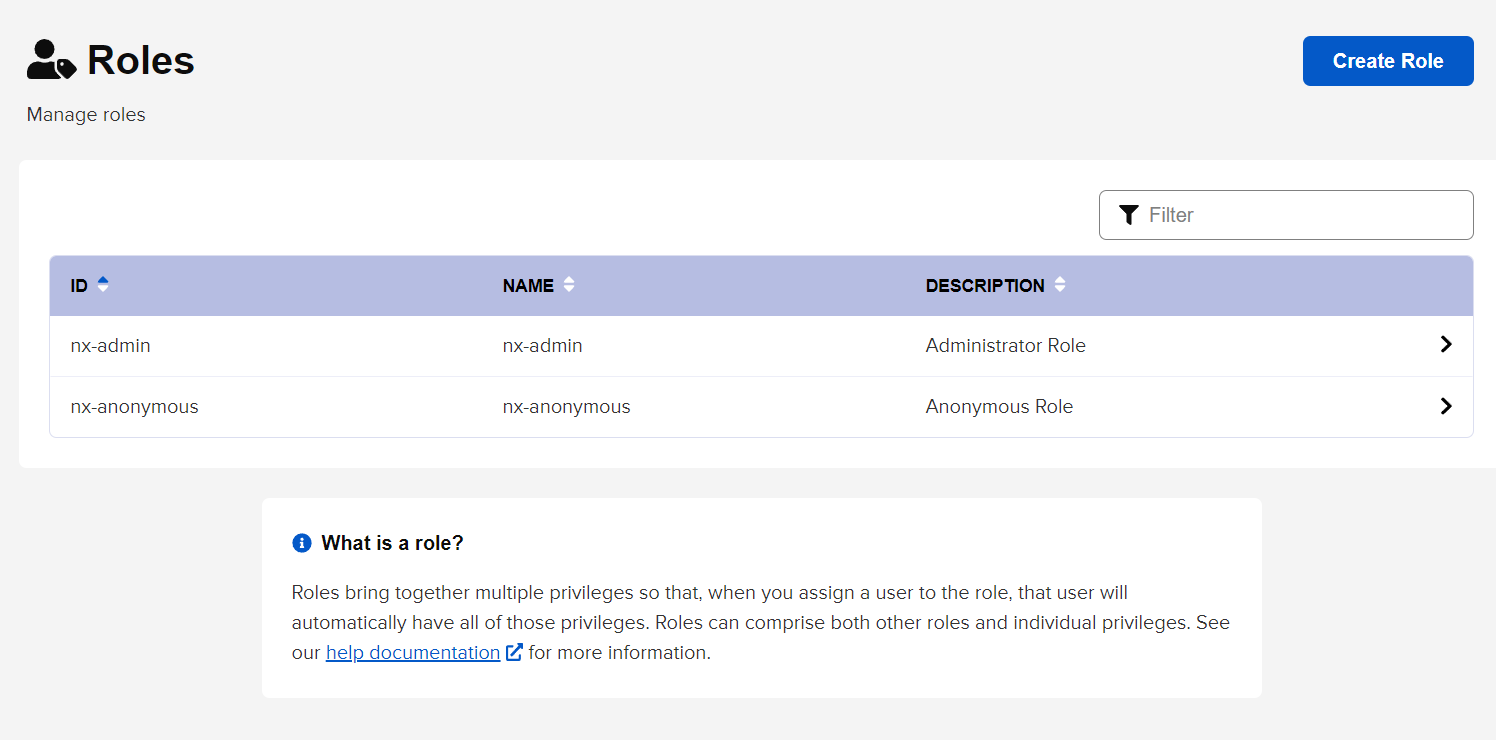
Then click on next and finish.

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

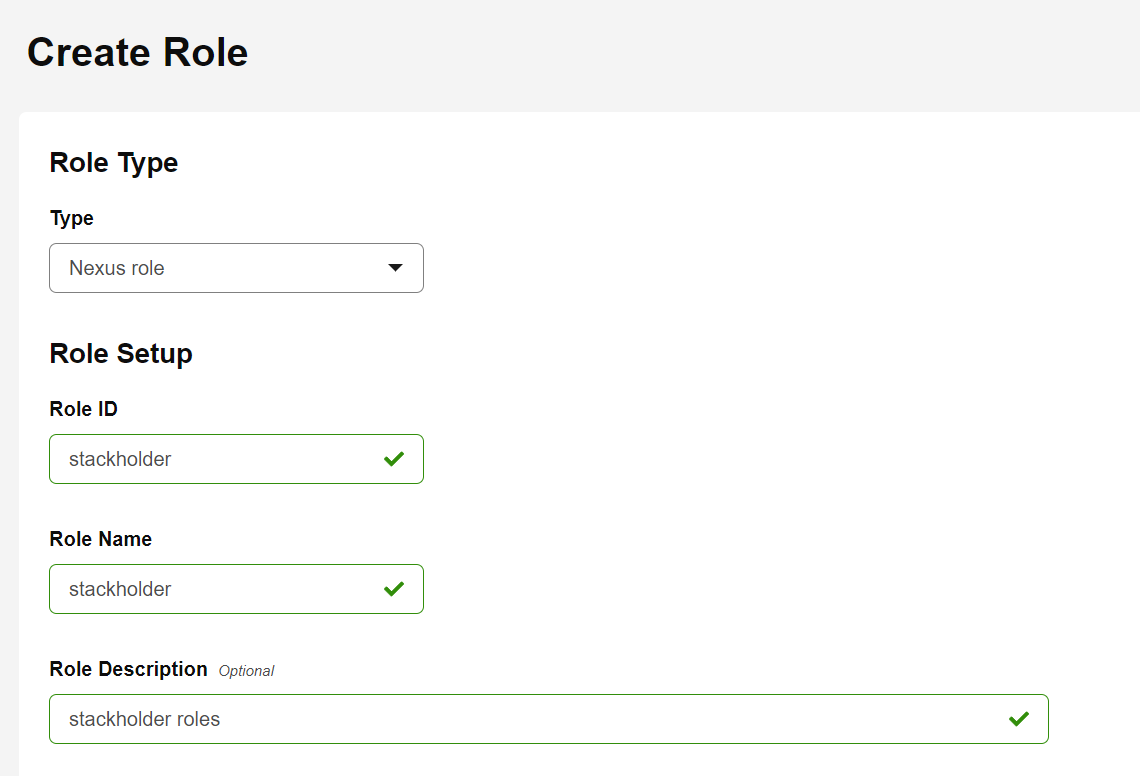
Go to server administration and configuration and click on the roles

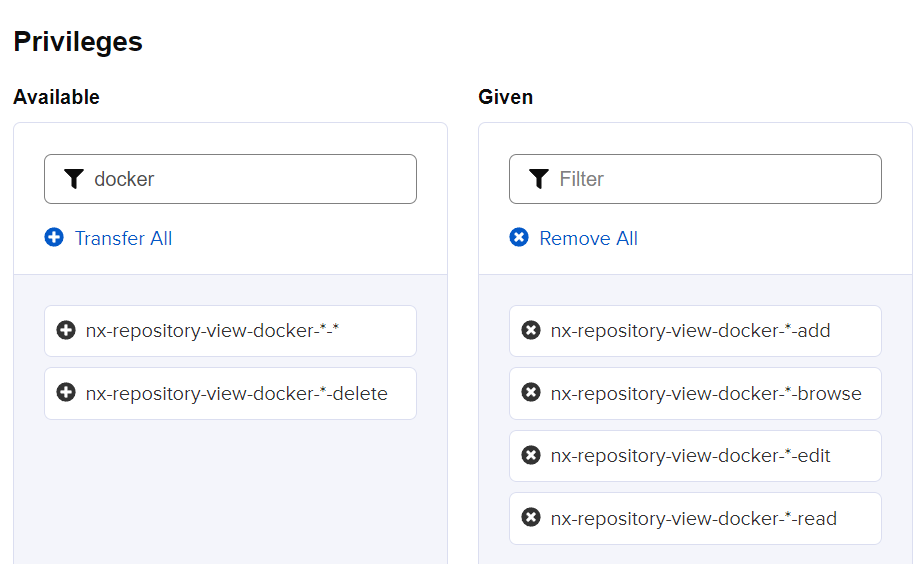


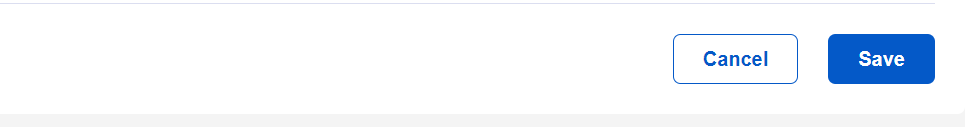
Click on create a role



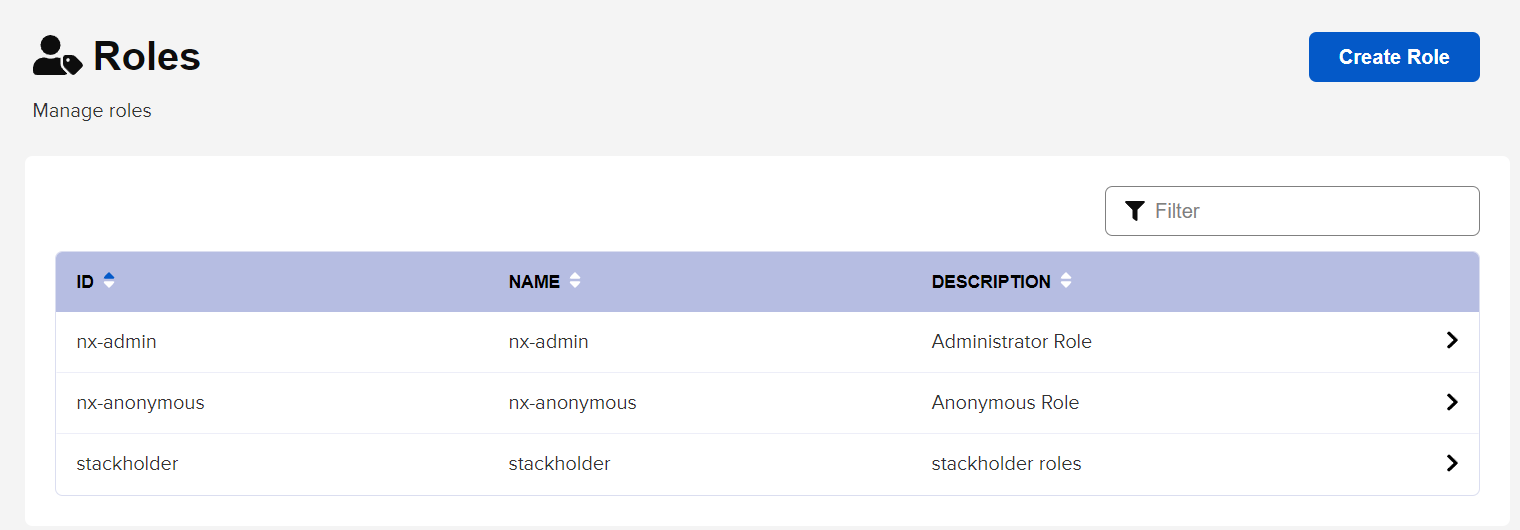
Add a role for an example. Give some docker privileges and click on save to create the role.



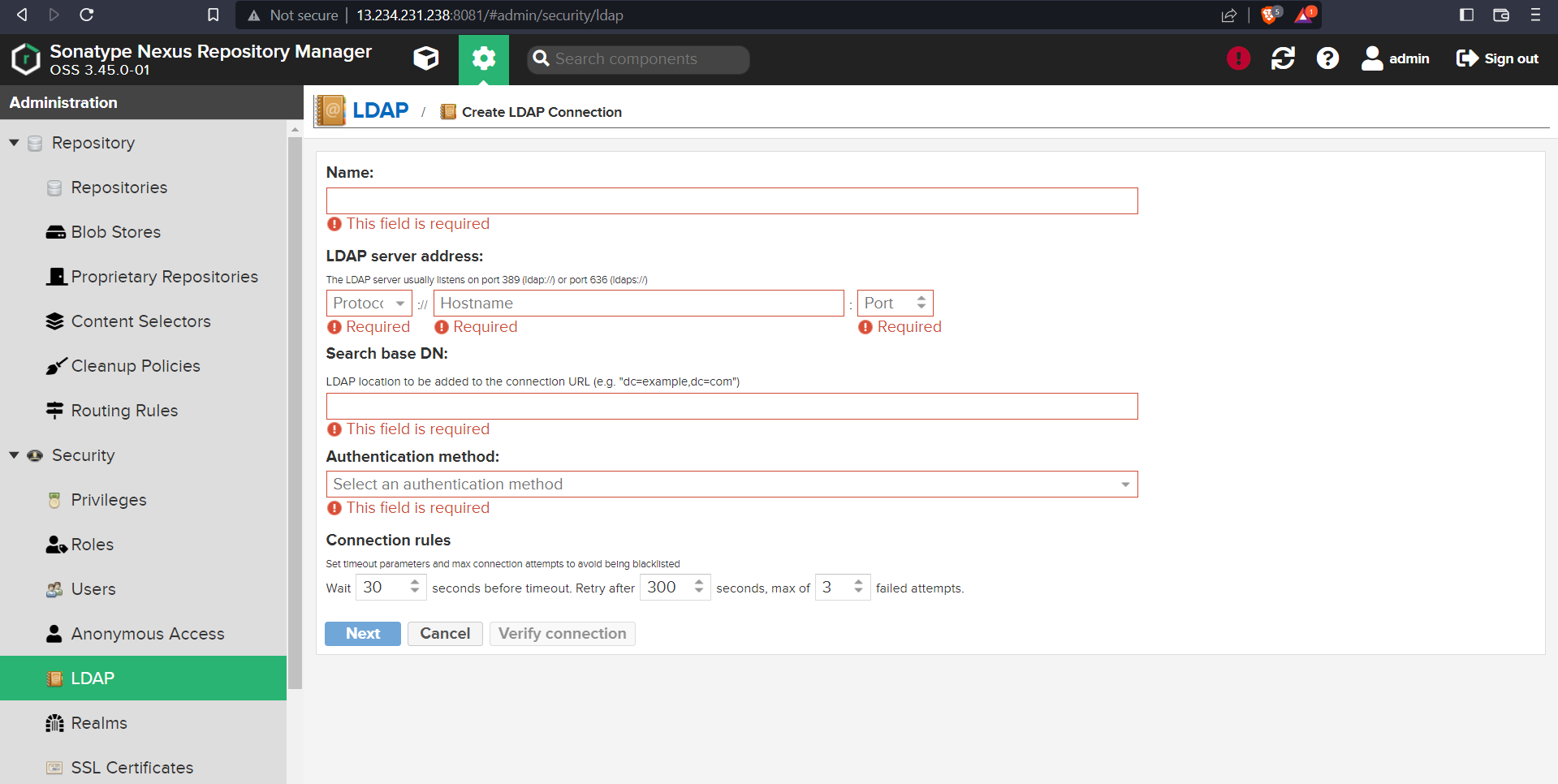




Now you can see the role has been added

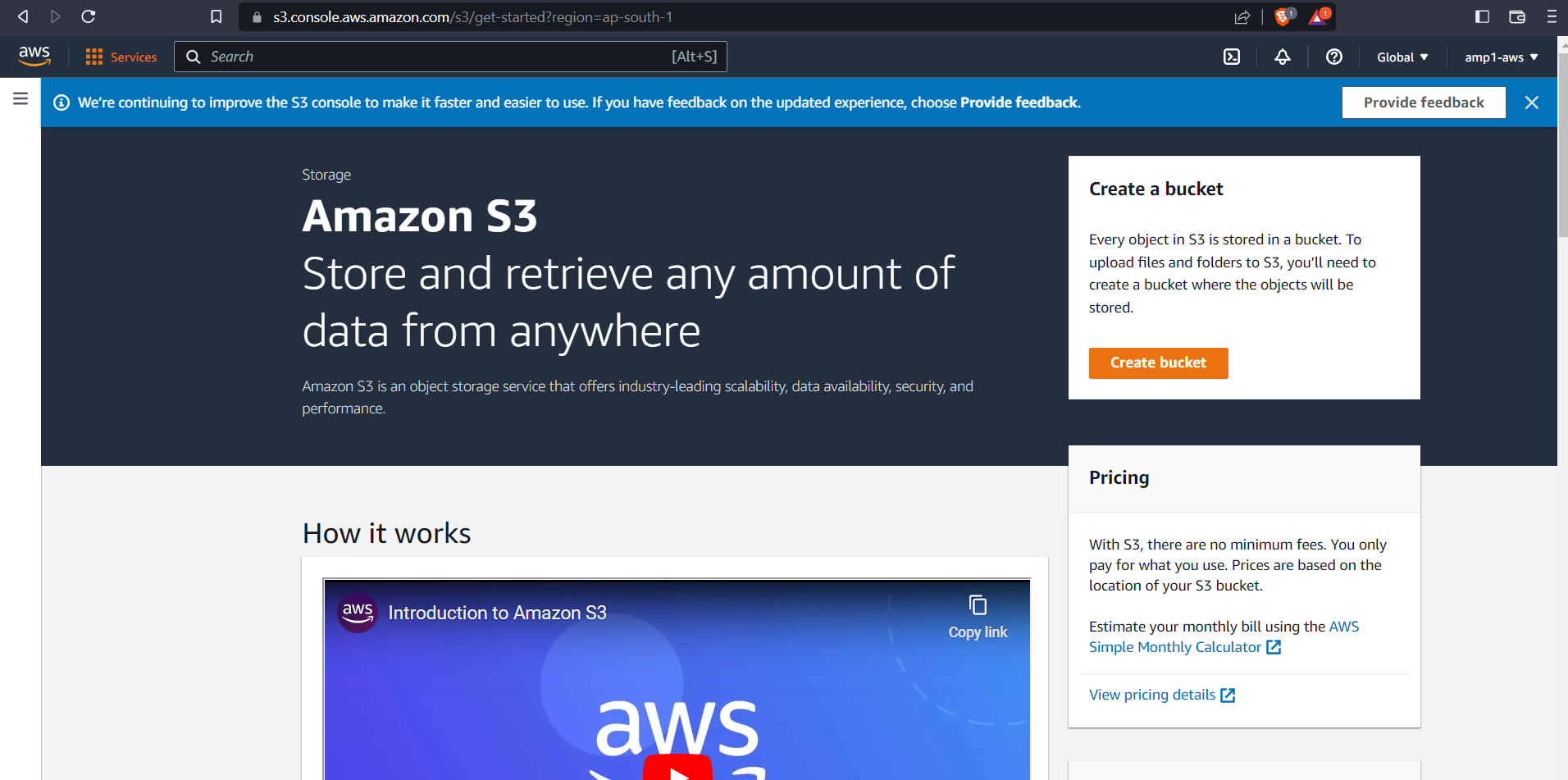


If we are integrated with LDAP server, we can add the LDAP connection ad add the roles to users

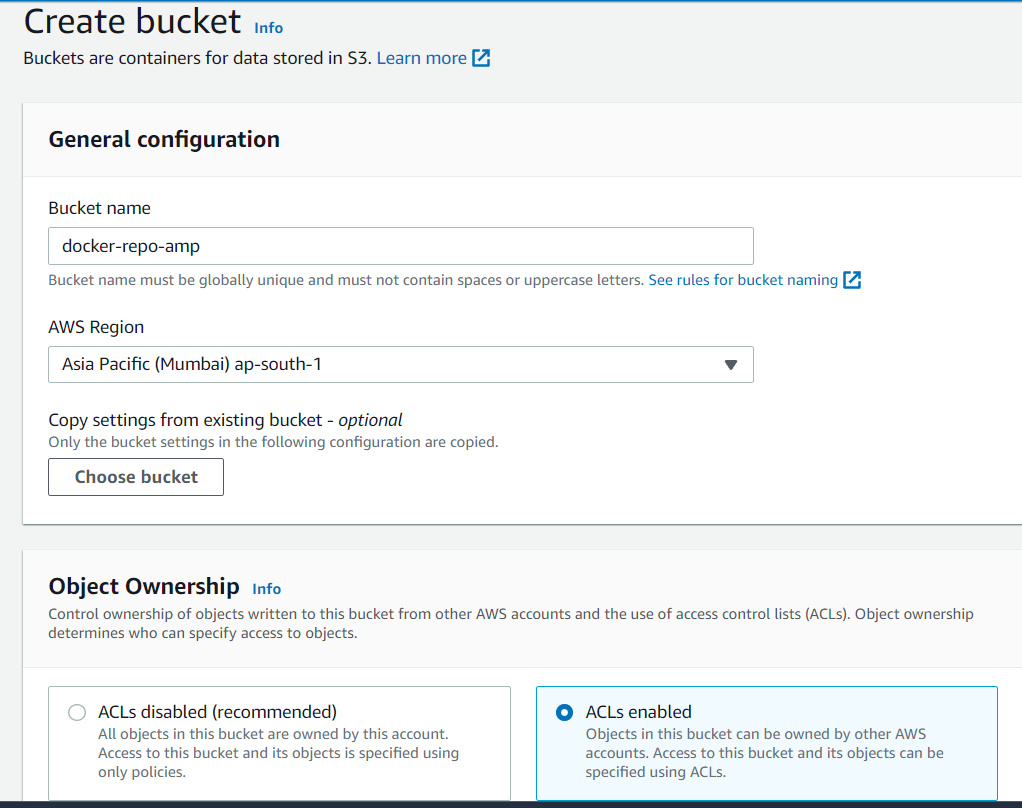


Currently we are not using LDAP server.

Go to AWS S3 and click on create bucket



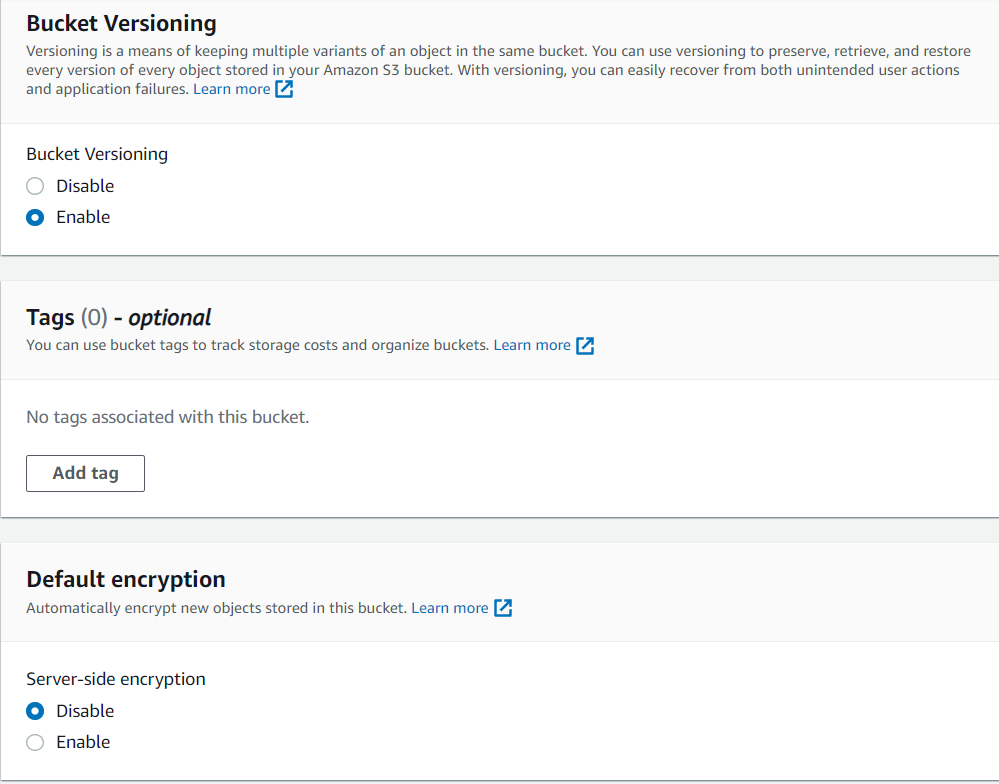
Enter the Bucket name, Region, and select ACL enabled



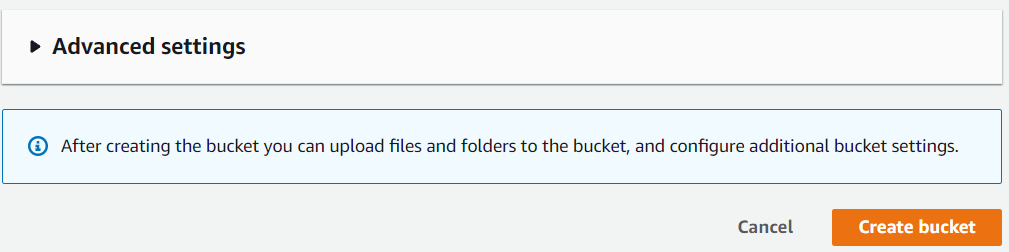
Untick block all public access and tick on I acknowledge that the current settings might result in this bucket and the objects within becoming public.



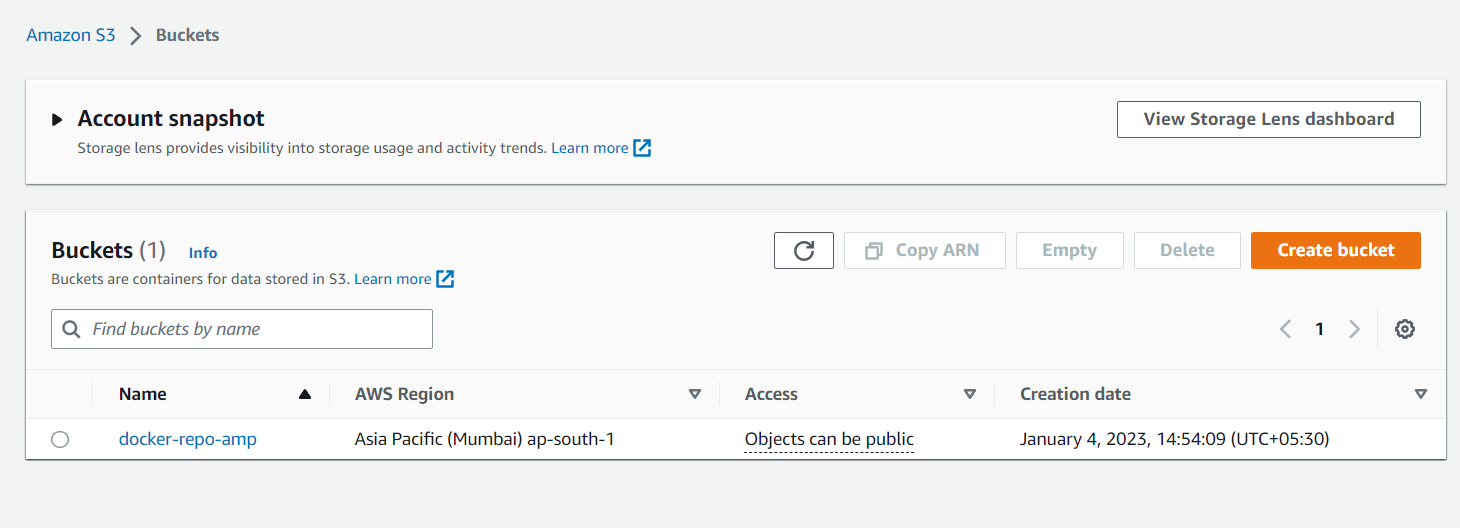
Select bucket versioning enable

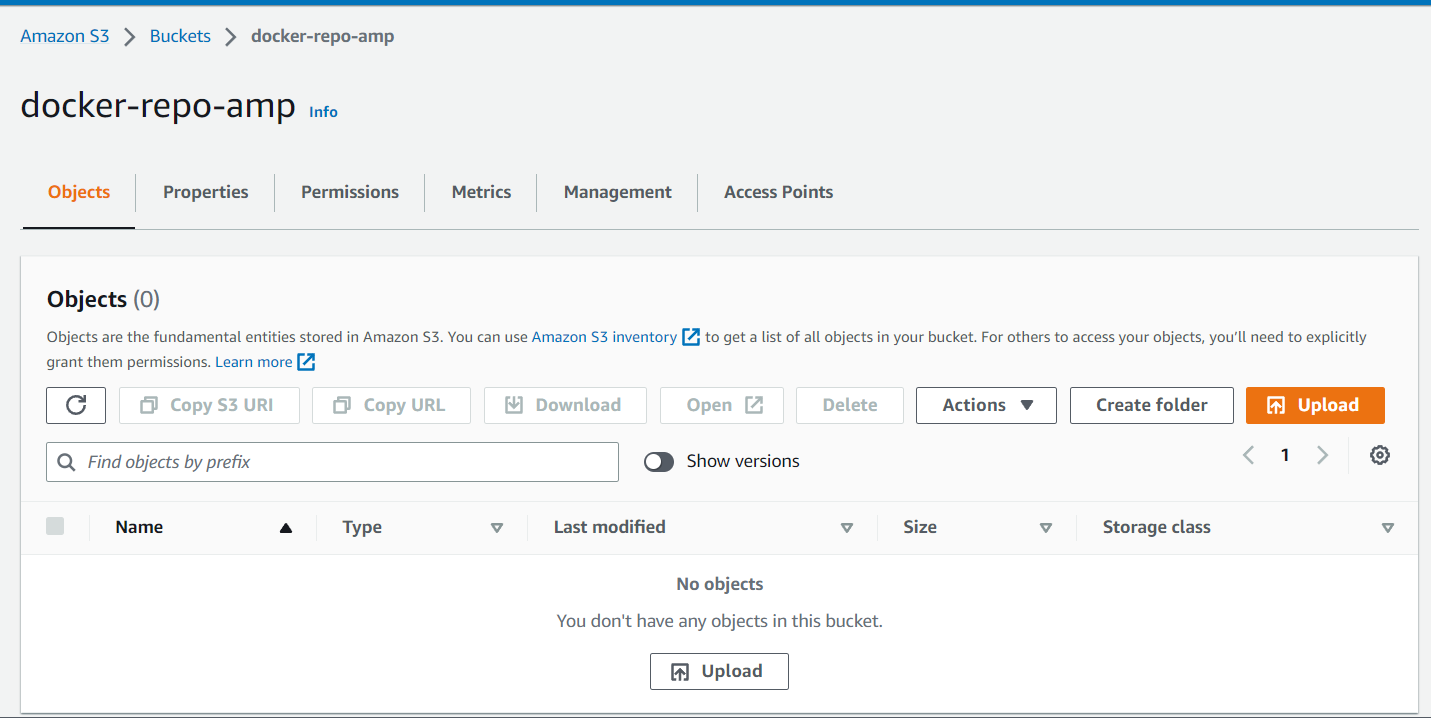


Then click on create bucket.



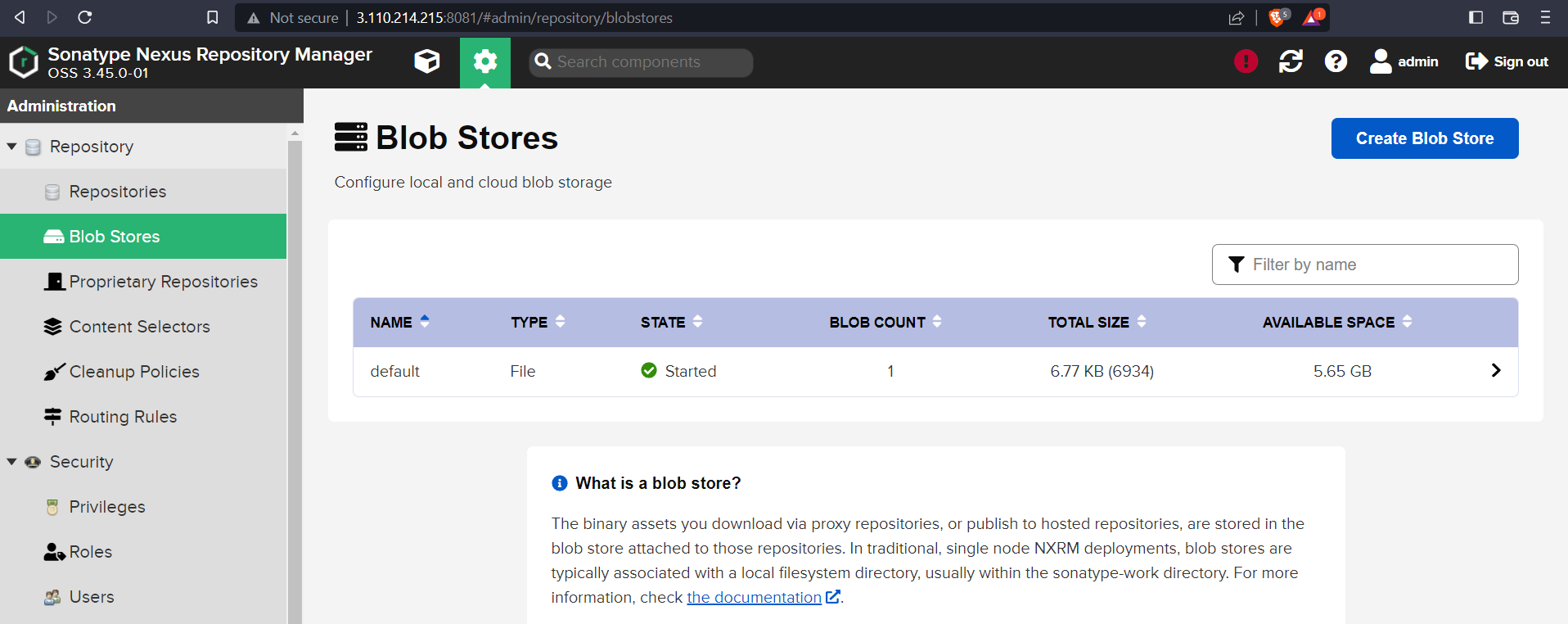
Now you can see the S3 bucket has been created

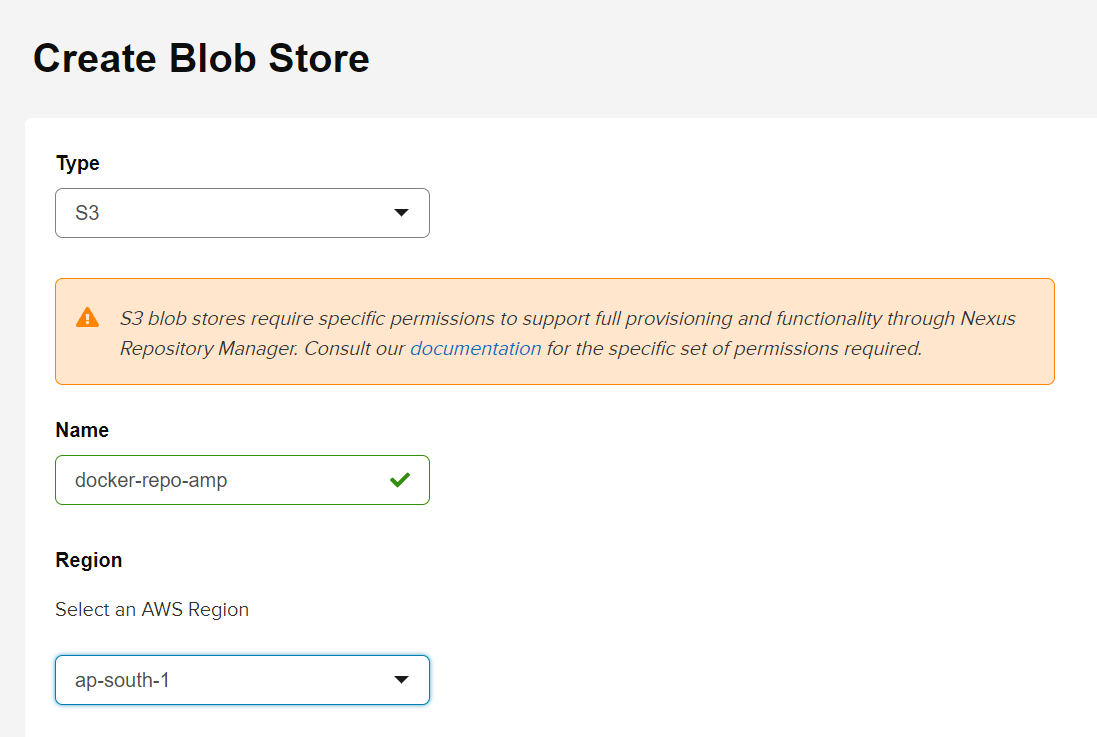


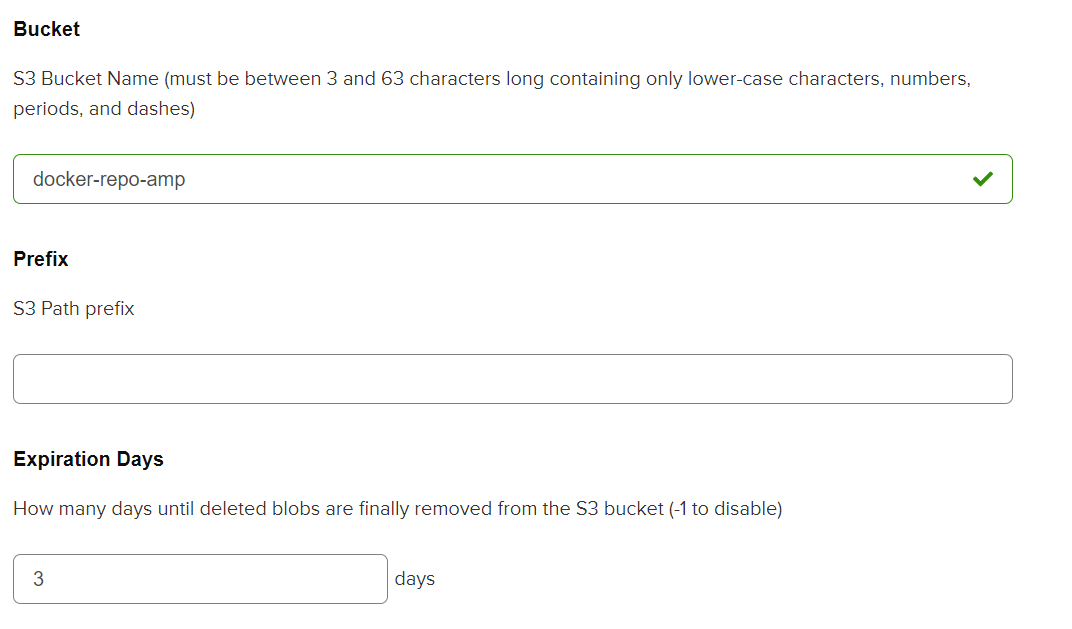


Go to nexus and click on blob stores

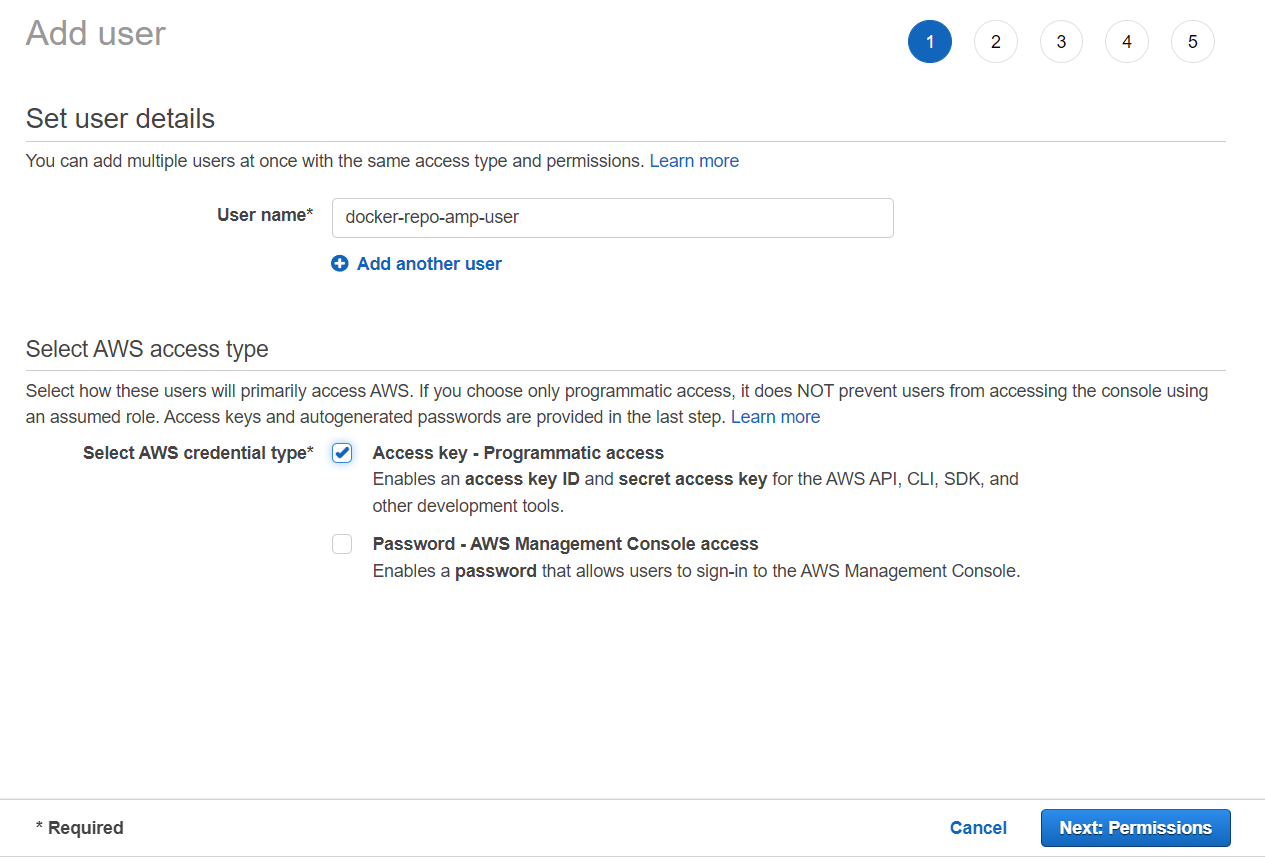
Click on create blob store

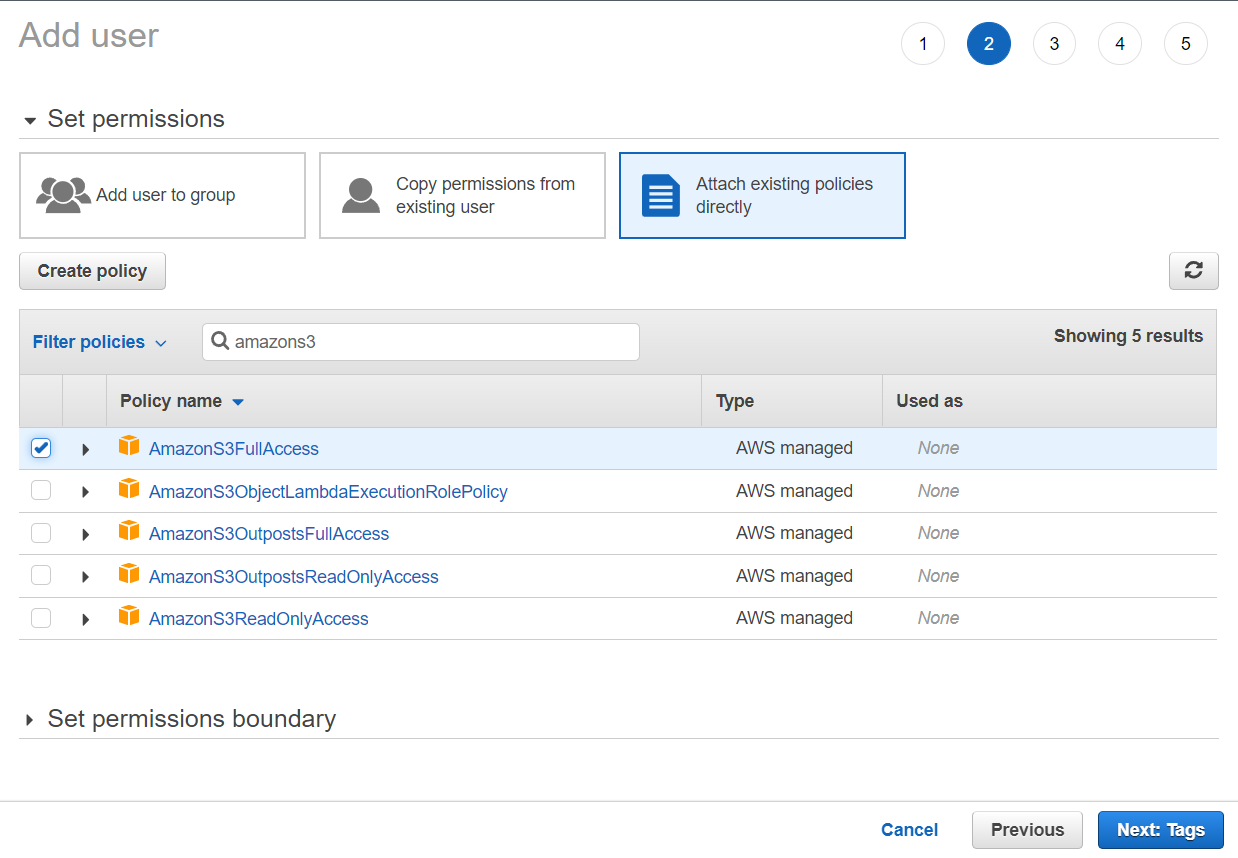


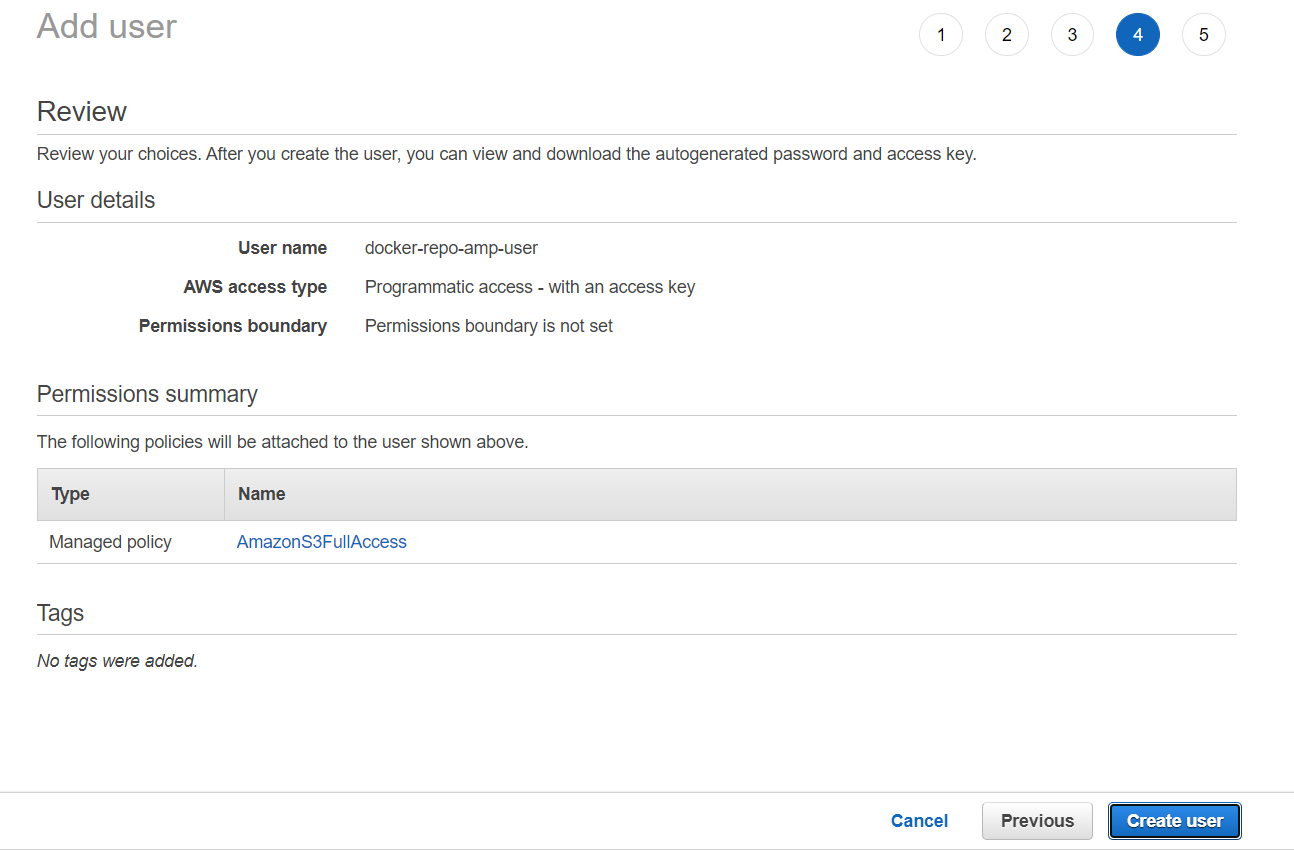


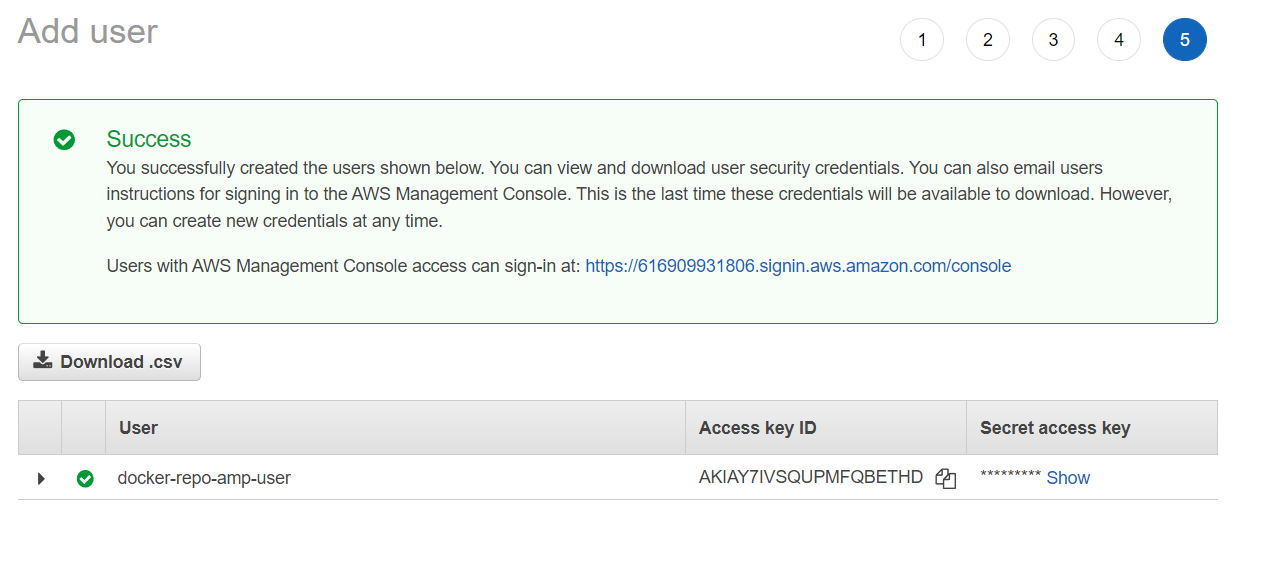


Go to AWS IAM and create a new user with amazonS3fullaccess

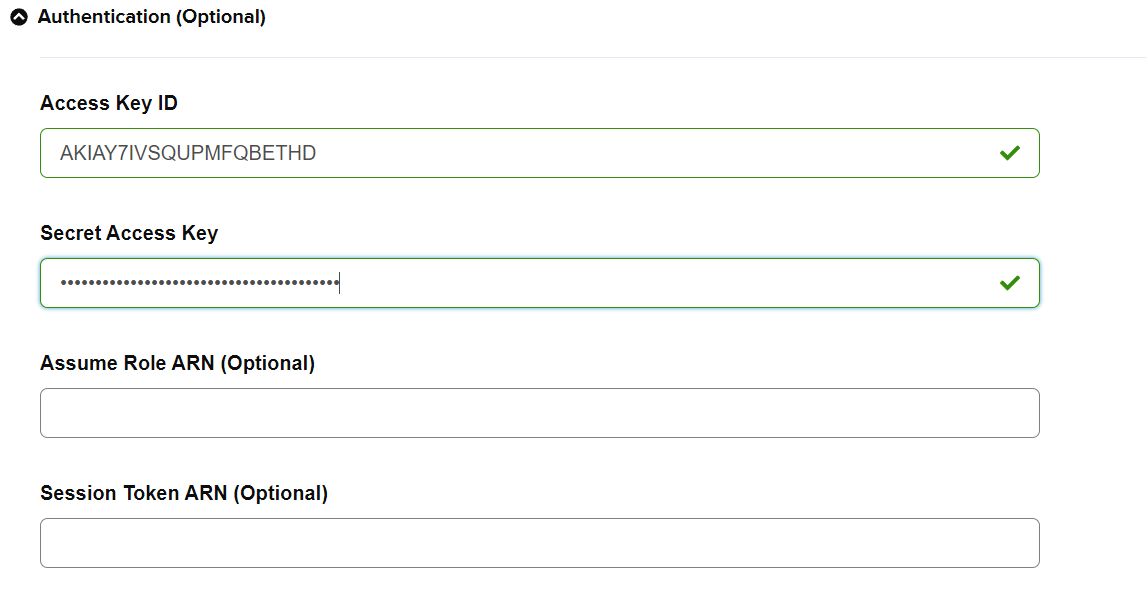




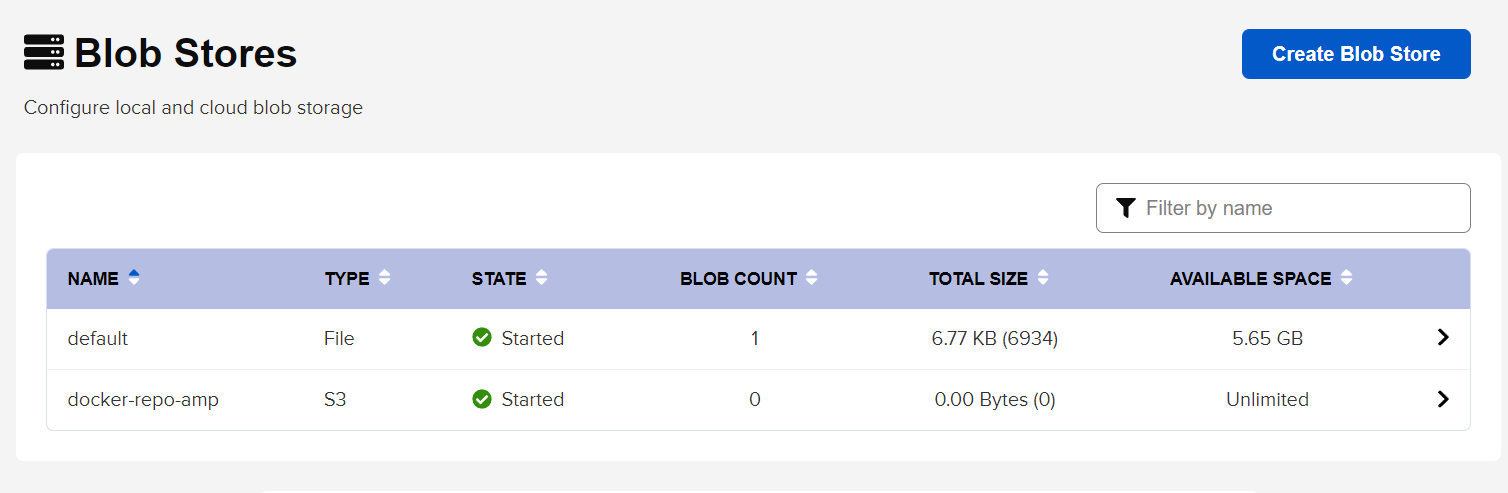




Copy the access key and secret access key to nexus

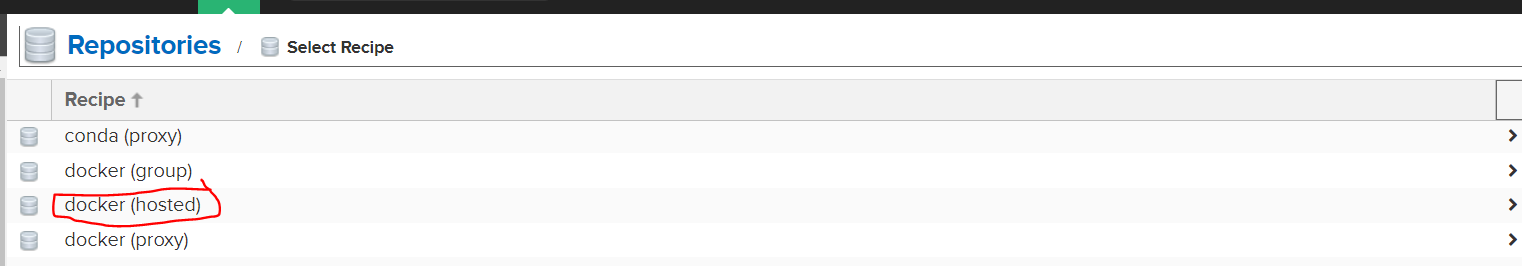


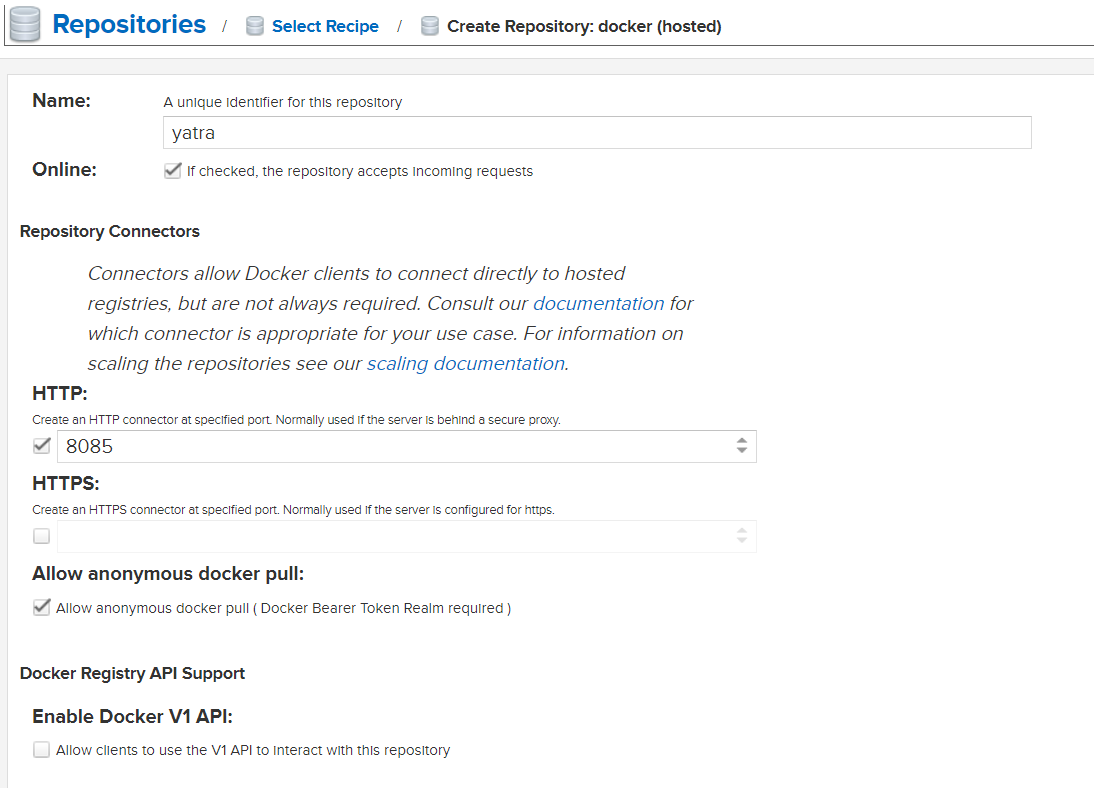
Then Click on Save.

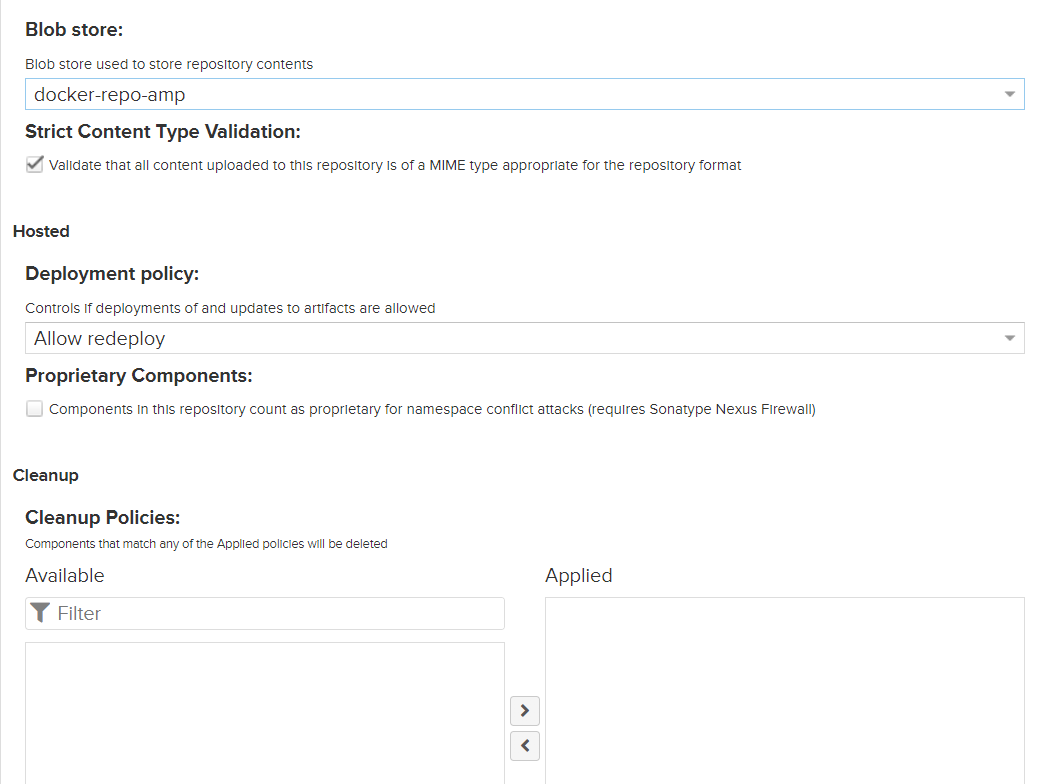


Go to repositories and click on create repository

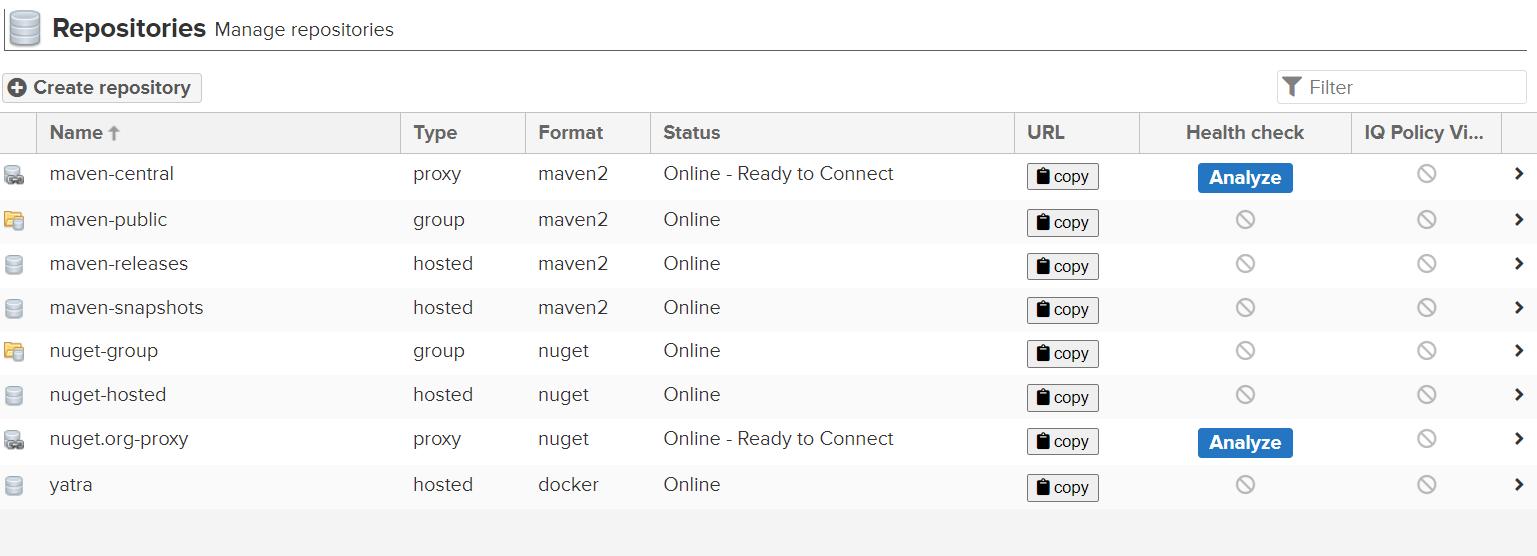
Click on docker (hosted)







Then click on create repository



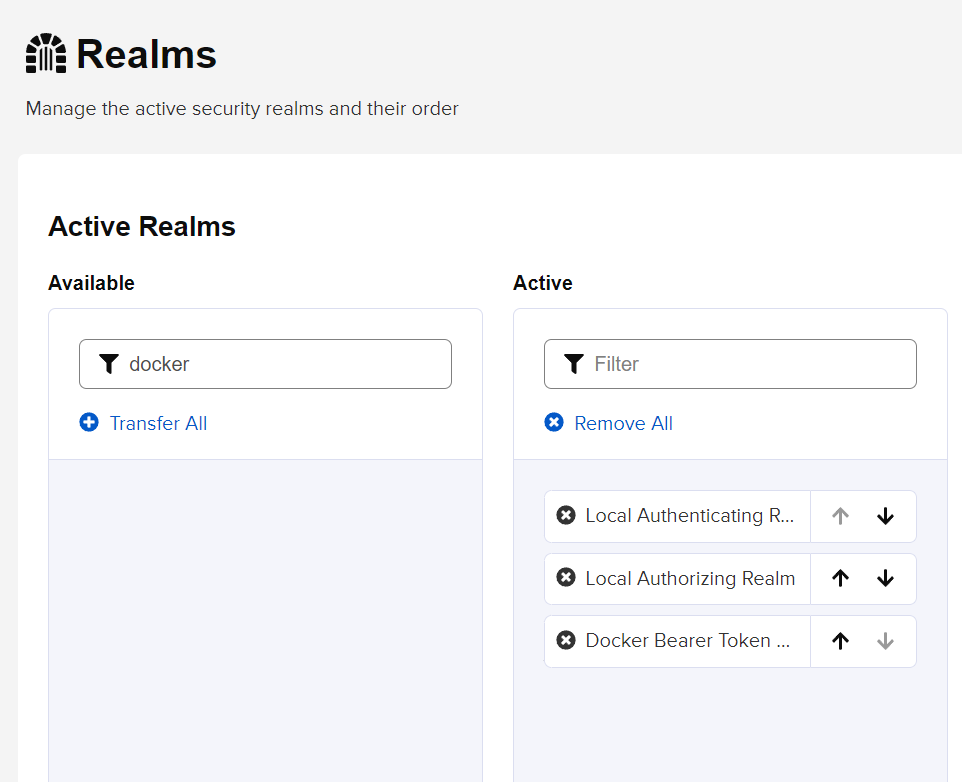
Click on realms

Activate

Local Authenticating realm

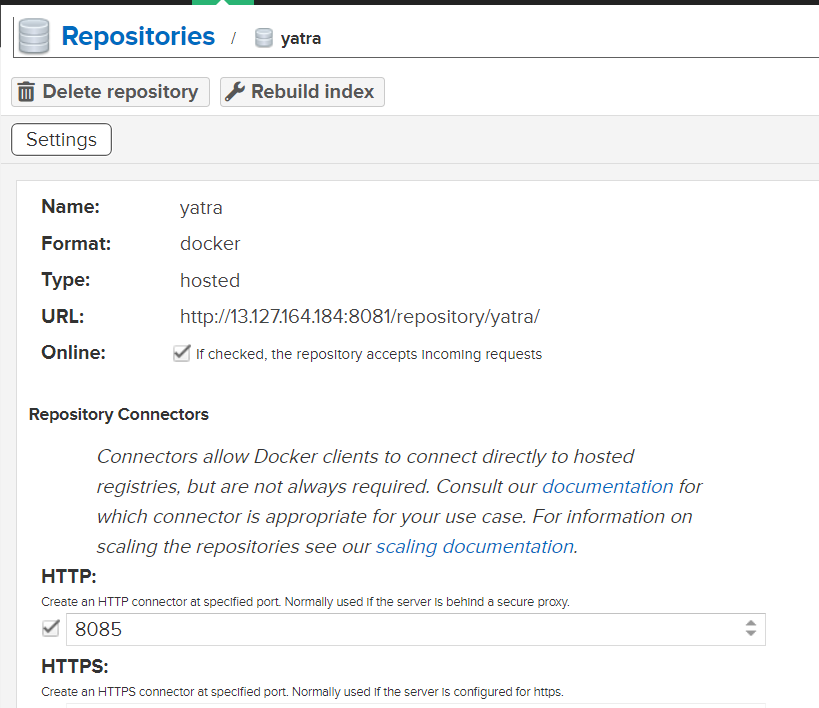
Local authorizing realm

Docker bearer token realm

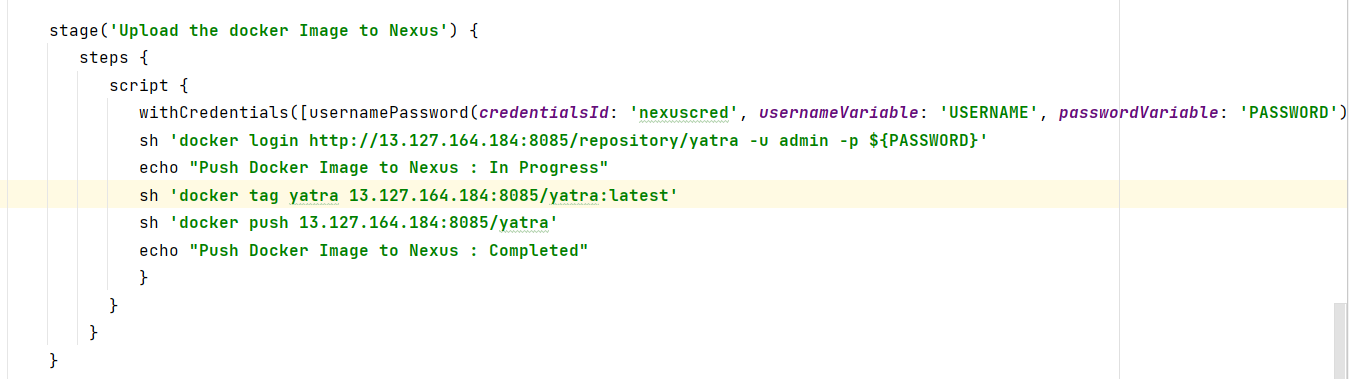


Click on save

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.



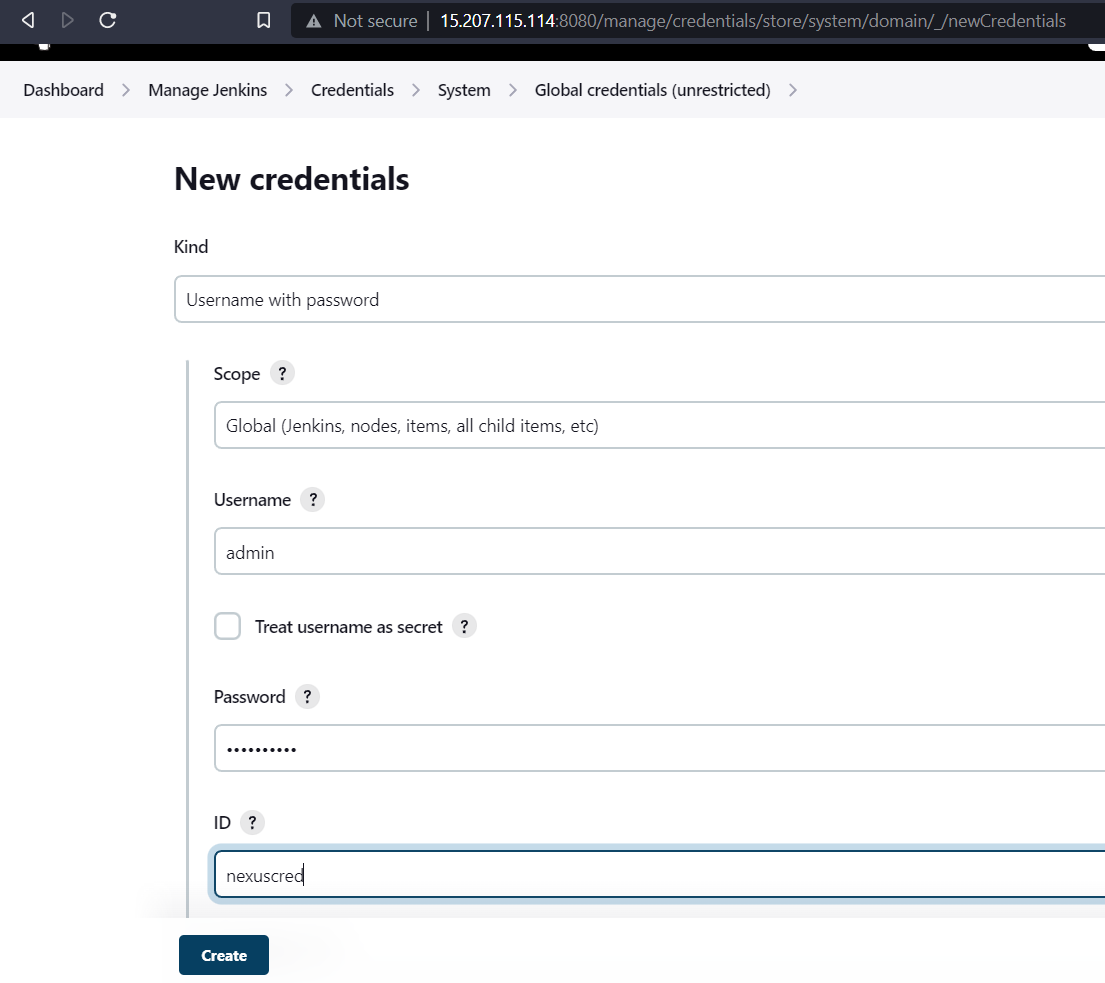
Push to git hub.

Git add --all

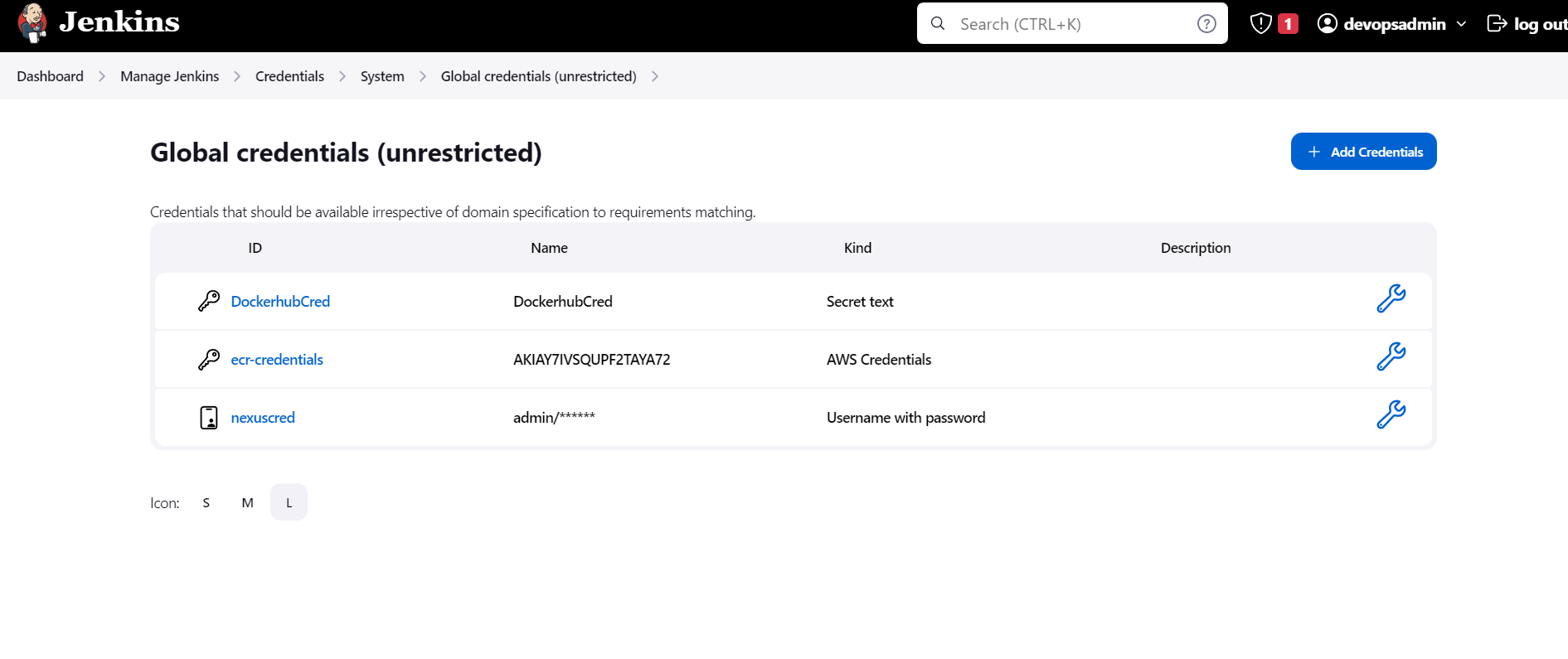
Git commit -m “ coderefactor”

Git push origin master

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



Click on create. And you can see the credentials.



Go to Jenkins Master terminal

#Add the file daemon.json and enter the nexus ip.

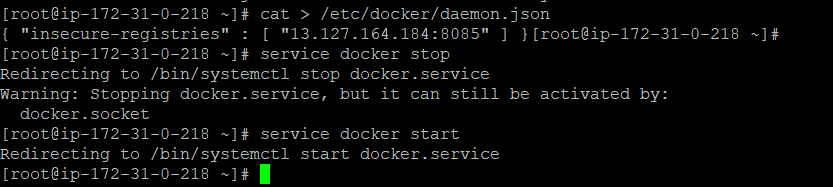
cat > /etc/docker/daemon.json

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

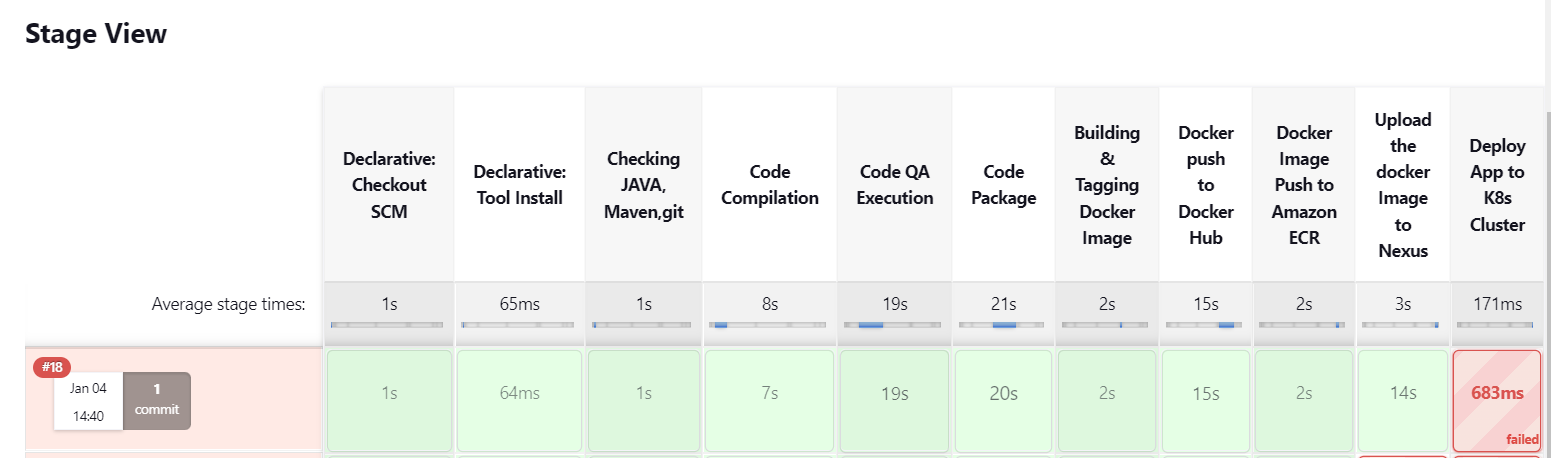
Ctrl+d

Service docker stop

Service docker start

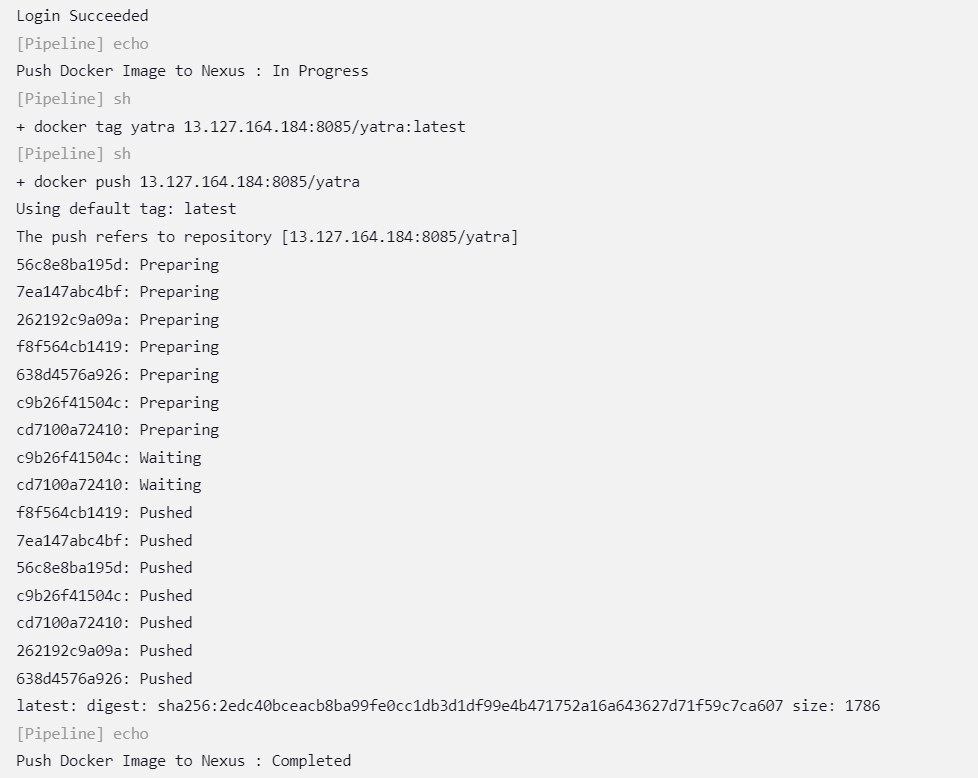


Go to Jenkins dash board and run the project yatra.

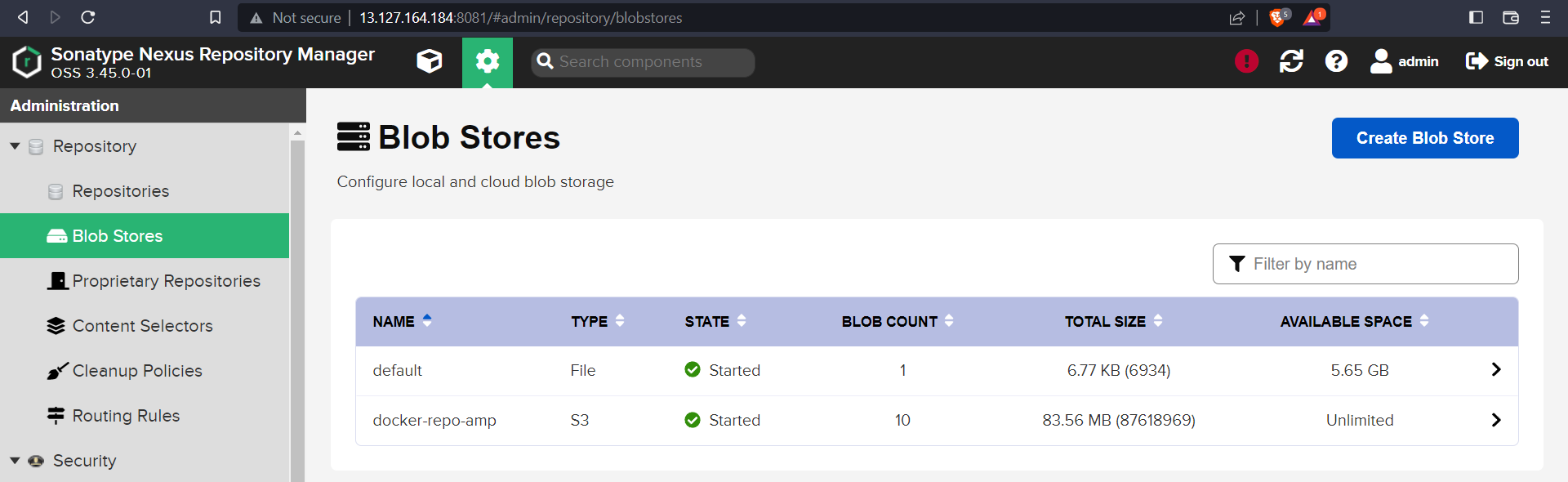


We can see the job was success until the image upload to nexus.

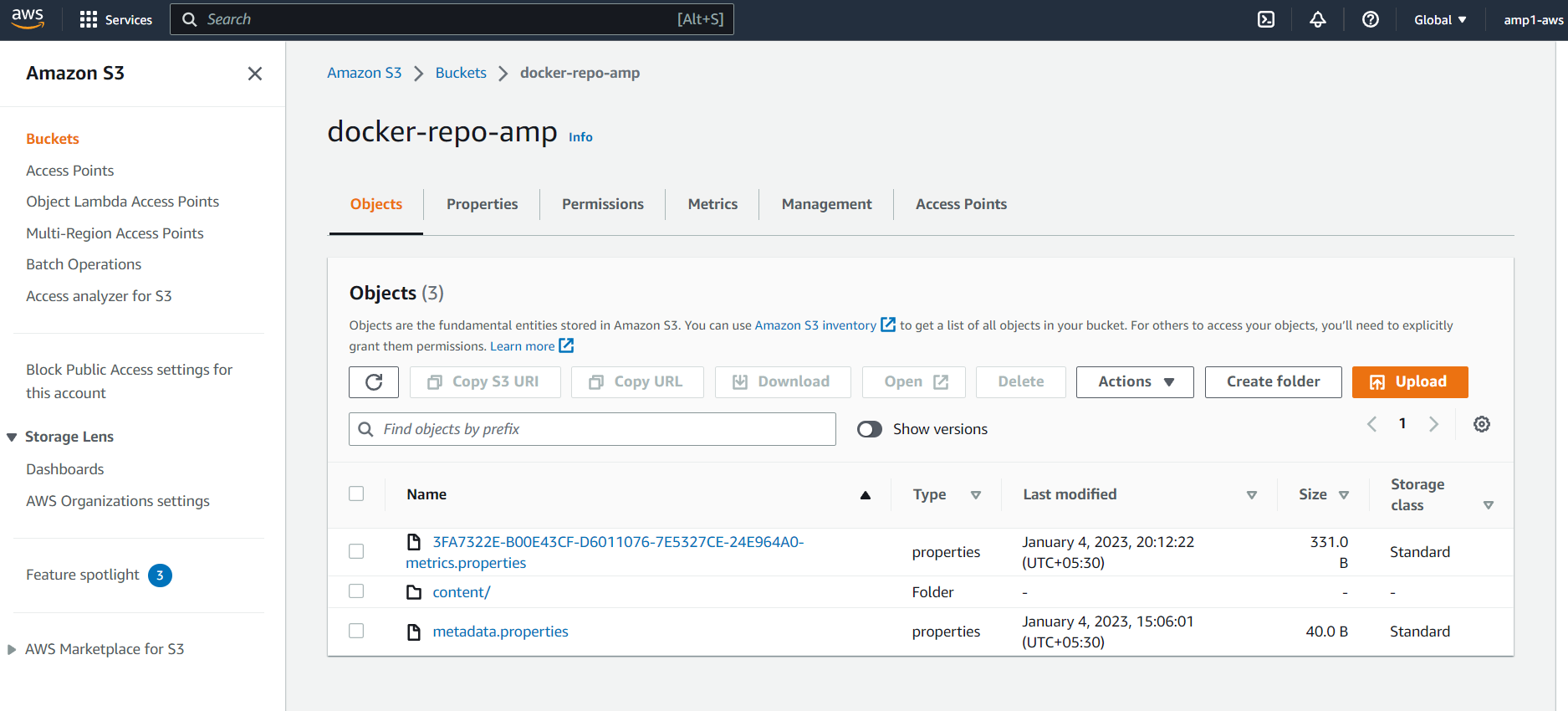
We can see the logs in console output.



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.

