

# NEXUS SETUP AND INTEGRATED TO JENKINS

STEP-1: LAUNCH INSTANCE WITH T2.MEDIUM AND 8081 PORT

STEP-2: INSTALL JAVA1.8.0 AS THE NEXUS DEPENDENCY

STEP-3: CREATE A FOLDER (mkdir /app)

STEP-4: GO TO FOLDER (cd /app)

STEP-5: DOWNLOAD THE NEXUS (**sudo wget -O nexus.tar.gz**  
**<https://download.sonatype.com/nexus/3/latest-unix.tar.gz>**)

STEP-6: UNTAR THE FILE (tar -zxvf filename)

STEP-7: RENAME THE NEXUS FILE (from nexus-3\* to nexus)

STEP-8: ADD NEXUS USER (useradd nexus)

STEP-9: CHANGE THE OWNERS OF THE FILES (chmod -R nexus:nexus file\_names)

STEP-10: CHANGE nexus.rc file (vim nexux/bin/nexus.rc)

run\_as\_user="nexus"

STEP-11: WRITE A SCRIPT TO RUN NEXUS (vi /etc/systemd/system/nexus.service)

[Unit]

Description=nexus service

After=network.target

[Service]

Type=forking

LimitNOFILE=65536

User=nexus

Group=nexus

ExecStart=/app/nexus/bin/nexus start

ExecStop=/app/nexus/bin/nexus stop

User=nexus

Restart=on-abort

[Install]

WantedBy=multi-user.target

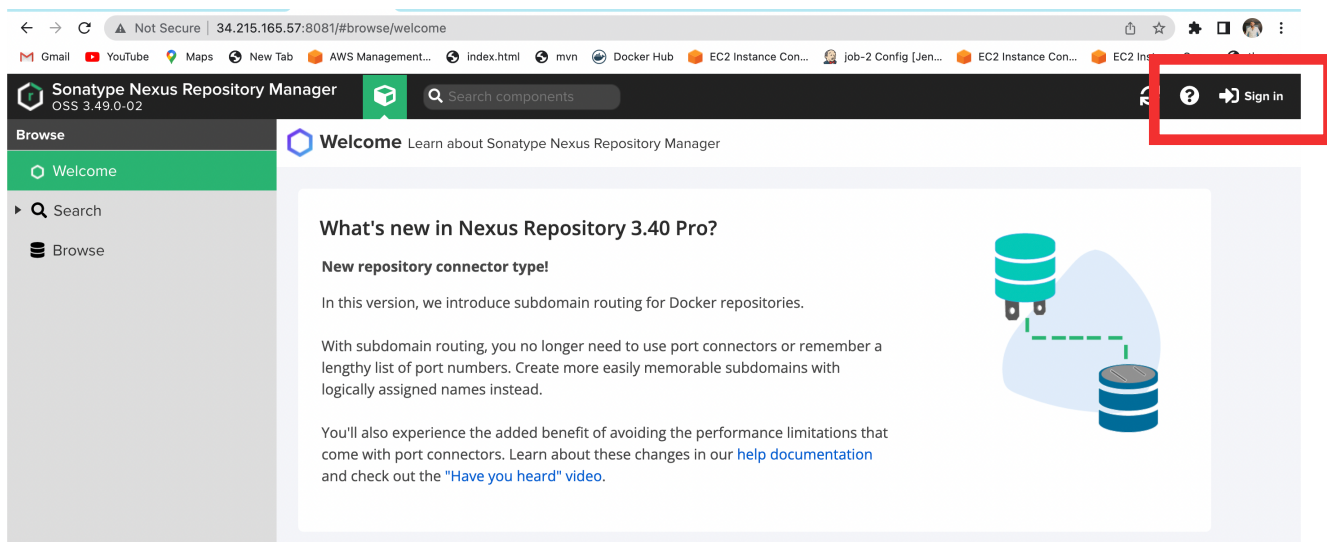
STEP-12: CHECK THE CONFIGURATION (chkconfig nexus on)

STEP-13: START THE NEXUS (systemctl start nexus)

STEP-14: CHECK THE STATUS (systemctl status nexus)

STEP-15: CONNECT TO NEXUS (public:8081)

AFTER LOGIN INTO NEXUS SIGN OUT FROM THE DASHBOARD, BECAUSE THAT IS THE DEFAULT PAGE, WE NEED TO CREATE OUR BY OWN



CLICK ON **SIGN-IN**

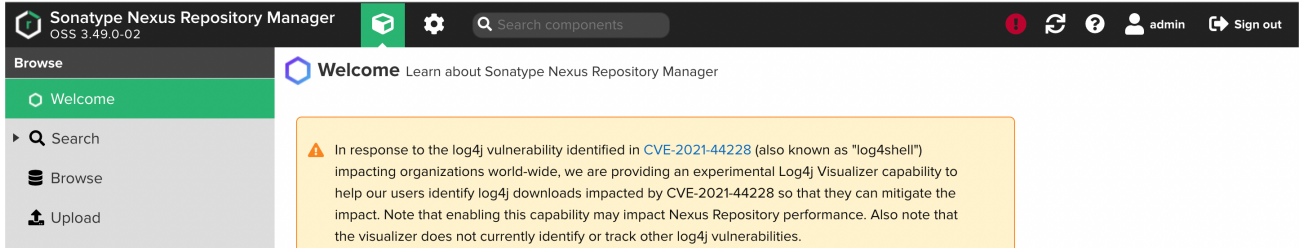
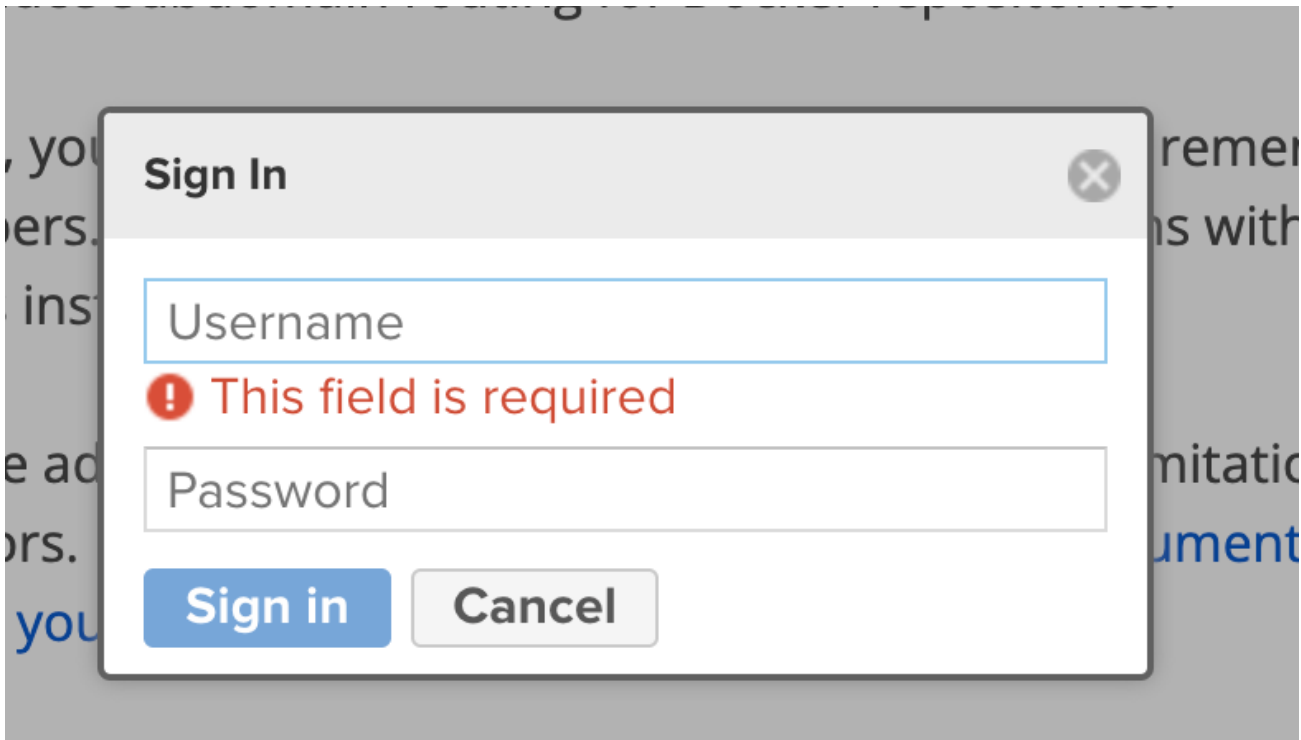
IT WILL ASKS THE USER NAME & PASSWORD

USERNAME: admin

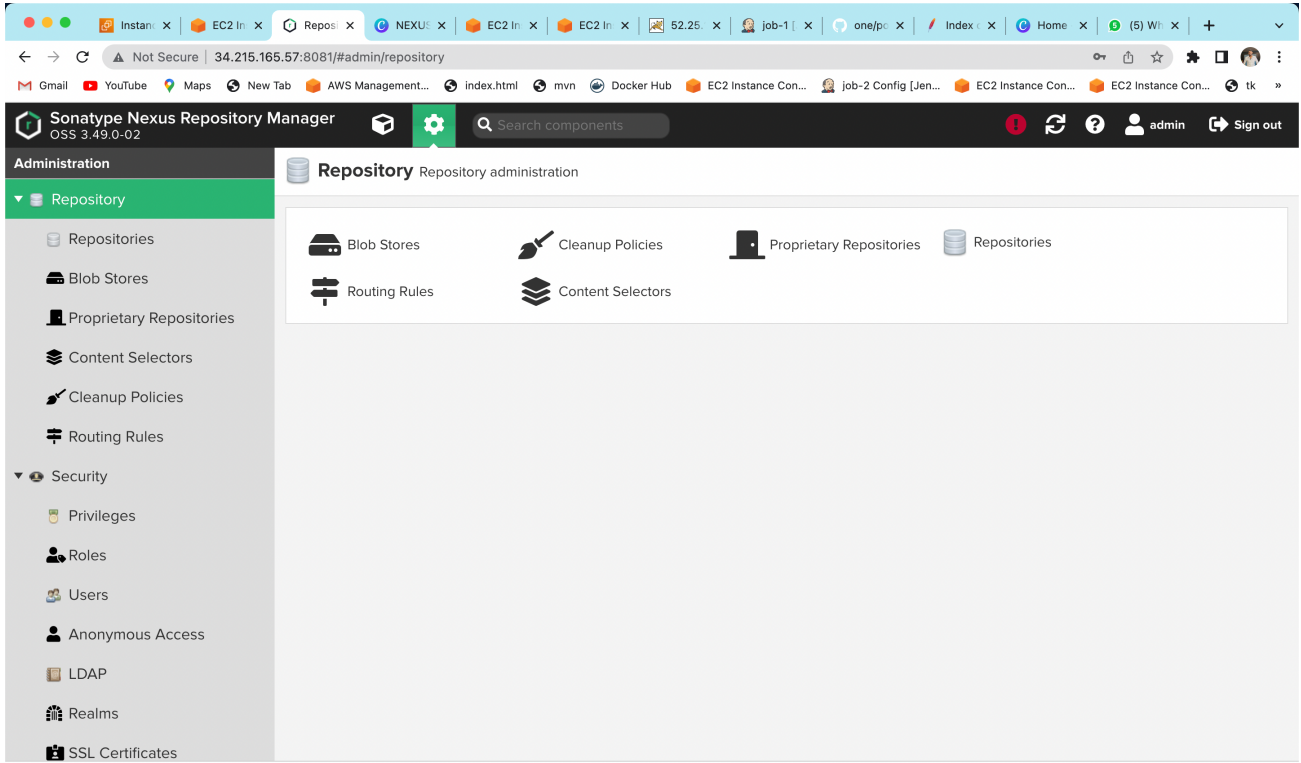
PASSWORD: cat /app/sonatype-work/nexus3/admin.password

use this command in our server to get the password and paste it into our nexus dashboard

use subdomain routing for Docker repositories.



NOW GO TO SERVER ADMINISTRATION AND CONFIGURATION



SELECT REPOSITORIES

Repositories

Manage repositories

Create repository

Filter

	Name ↑	Type	Format	Status	URL	Health check	IQ Policy Vi...	
	maven-central	proxy	maven2	Online - Ready to Connect	<div>copy</div>	Analyze		>
	maven-public	group	maven2	Online	<div>copy</div>			>
	maven-releases	hosted	maven2	Online	<div>copy</div>			>
	maven-snapshots	hosted	maven2	Online	<div>copy</div>			>
	nuget-group	group	nuget	Online	<div>copy</div>			>
	nuget-hosted	hosted	nuget	Online	<div>copy</div>			>
	nuget.org-proxy	proxy	nuget	Online - Remote Available	<div>copy</div>	Analyze		>

SELECT CREATE-REPOSITORY

Repositories

Select Recipe

Recipe ↑

apt (hosted)

>

apt (proxy)

>

bower (group)

>

bower (hosted)

>

bower (proxy)

>

cocoapods (proxy)

>

conan (proxy)

>

conda (proxy)

>

docker (group)

>

docker (hosted)

>

docker (proxy)

>

gitlfs (hosted)

>

go (group)

>

go (proxy)

>

helm (hosted)

>

helm (proxy)

>

maven2 (group)

>

maven2 (hosted)

>

maven2 (proxy)

>

npm (group)

>

npm (hosted)

>

SELECT MAVEN2(HOSTED)

Repositories

Select Recipe

Create Repository: maven2 (hosted)

Name:

A unique identifier for this repository

This field is required

Online:

☒ If checked, the repository accepts incoming requests

Maven 2

Version policy:

What type of artifacts does this repository store?

Release

Layout policy:

Validate that all paths are maven artifact or metadata paths

Strict

Content Disposition:

Add Content-Disposition header as 'Attachment' to disable some content from being inline in a browser.

Inline

GIVE REPOSITORY NAME AS myapp-releases and click on create repositories

Repositories

Manage repositories

Create repository

Filter

	Name ↑	Type	Format	Status	URL	Health check	IQ Policy Vi...	
	maven-central	proxy	maven2	Online - Ready to Connect		Analyze		>
	maven-public	group	maven2	Online				>
	maven-releases	hosted	maven2	Online				>
	maven-snapshots	hosted	maven2	Online				>
	myapp-releases	hosted	maven2	Online				>
	nuget-group	group	nuget	Online				>
	nuget-hosted	hosted	nuget	Online				>
	nuget.org-proxy	proxy	nuget	Online - Remote Available		Analyze		>

now you can see our repository created in dashboard

NOW GO TO JENKINS AND INSTALL PLUGIN "**nexus artifactory uploader**"

GO TO MANAGE JENKINS AND CONFIGURE JENKINS

GO TO SONATYPE NEXUS

Sonatype Nexus

Nexus Repository Manager Servers

Nexus Repository Manager 3.x Server

Nexus Repository Manager 3.x Server

Display Name

my-nexus

Server ID ?

nexus-repo

Server URL

http://34.215.165.57:8081

ADD CREDENTIALS :

Kind

Username with password

Scope ?

Scope ?

Global (Jenkins, nodes, items, all child items, etc) ▼

Username ?

admin

☐ Treat username as secret ?

Password ?

.....

ID ?

and add description as nexus credentials

CLICK ON TEST CONNECTION, THEN YOU WILL SEE THE MESSAGE

⚠

Nexus Repository Manager 3.x connection succeeded

Test connection

NXRM OSS 3.49.0-02 found. Some operations require Nexus Repository Manager Professional server version 3.13.0 or newer; use of an incompatible server could result in failed builds.

NOW GO AND CREATE A JOB

GIT: URL OF OUR PROJECT

Dashboard > job-1 > Configuration

Configure

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

None

Git ?

Repositories ?

Repository URL ?

https://github.com/devops0014/one.git

Credentials ?

- none - ▼

Add ▼

Advanced ▼

Add Repository

ADD BUILD STEP: invoke top-level maven targets

Build Steps

≡

Invoke top-level Maven targets ?

✖

Goals

clean package

Advanced ▾

## ADD BUILD STEP: Nexus artifact uploader

### ⌵ Nexus artifact uploader



#### Nexus Details

Nexus Version

NEXUS3



Protocol

HTTP



Nexus URL ?

34.25.165.57:8081

Credentials

admin/\*\*\*\*\* (nexus)



Add ▾

GroupId

in.javahome

Version

8.2.4

Repository ?

ashokit-releases

Artifacts

Artifact



ArtifactId

myweb

Type ?

.war



Classifier ?

File ?

target/myweb-8.3.1.war

Add

NOW GO TO POST BUILD ACTIONS: deploy war/ear to container

Configure

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

Post-build Actions

Deploy war/ear to a container

WAR/EAR files ?

target/\*.war

Context path ?

swiggy

Containers

Tomcat 9.x Remote

Credentials

tomcat/\*\*\*\*\* (tomcat credentials)

Add

Tomcat URL ?

http://52.25.184.150:8080/

NOW SAVE AND BUILD THE JOB

IF THE BUILD GETS SUCCESS:

then we can see that war files are stored in nexus repository

=====

Or or or or or or or or or or or or or or or

=====

Install nexus **platform plugin** in jenkins

Go to jenkins —> configuration system —> nexus

Display name: nexus

Server-id: nexus-repo



Server url: public ip of nexus:8081

Credentials: \*\*\*\*

Test connection

Go to job:

Configure —>

Add a build step: **Nexus Repository Manager Publisher**

Nexus instance: auto update

Nexus-repo: sample-releases

packages:

Group: search in pom.xml

Artifact id: search in pom.xml

Version: search in pom.xml

Package: war

Artifact:

File path: target/myweb-8.2.5.war

Extension: .war