**Nexus** **Repository Manager for Docker Repository**

**Document**

Version 1.0

logo

Girmiti Software Private Limited

SLV PLAZA, ARVIND AVENUE

KUNDALAHALLI,

MARATHAHALLI BANGALORE 560037

**Document History**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Submitted Date** | **Prepared by** | **Reviewed by** | **Revision History** |
| 1.0 | 23/10/2018 | Girmiti Software |  | Draft Document |

# **Nexus Repository Manager**

## **1. Install the docker service:-**

* Before install the Docker service setup Docker repository, show below

# sudo yum-config-manager ---add-repo <https://download.docker.com/linux/docker-ce.repo>

* Now install the docker by following below command

# sudo yum install docker-ce -y

* To start the Docker services

# sudo systemctl start docker

# sudo sysytemctl status docker (to check the status of the Docker

* To check the Docker version & info

# docker version (to check the version of the docker)

[root@chef-server docker]# docker version

Client:

Version: 18.09.0-beta3

API version: 1.39

Go version: go1.10.4

Git commit: 7718f80

Built: Thu Oct 4 23:53:03 2018

OS/Arch: linux/amd64

Experimental: false

Server: Docker Engine - Community

Engine:

Version: 18.09.0-beta3

API version: 1.39 (minimum version 1.12)

Go version: go1.10.4

Git commit: 7718f80

Built: Thu Oct 4 23:24:09 2018

OS/Arch: linux/amd64

Experimental: false

* #Docker info (for Docker information's)
* To stop the Docker service

# sudo systemctl stop docker

* To check the docker image list

# docker images

* To check the running docker container

# docker ps

To check all docker container list

# docker ps -a

**2. Nexus Repository Manager**

* Per-configure for Nexus Repository
  + Before run or pull the nexus images need to configure in daemon.json which is present in below path

# cd /etc/docker

# vi daemon.json

(update & add below details)

{

"insecure-registries": [

"host-name:port”,

"host-name:port”,

],

"disable-legacy-registry": true

}

For Example

{

"insecure-registries": [

"192.168.2.171:8081",

"192.168.2.171:8082",

"192.168.2.171:8083",

"192.168.2.171:8443",

"192.168.2.171:18082",

"192.168.2.171:18083"

],

"disable-legacy-registry": true

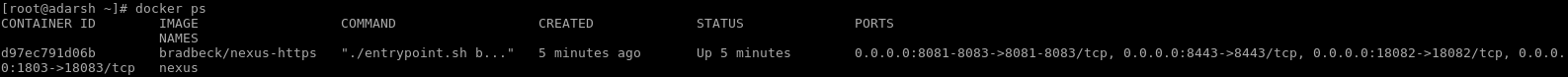
}

Save the configure & restart the dockar service

# systemctl restart docker

* To start or run the nexus image in docker

# docker run -d --rm -p 8443:8443 -p 8081:8081 -p 8082:8082 -p 8083:8083 -p 18082:18082 -p 18083:18083 --name nexus bradbeck/nexus-https

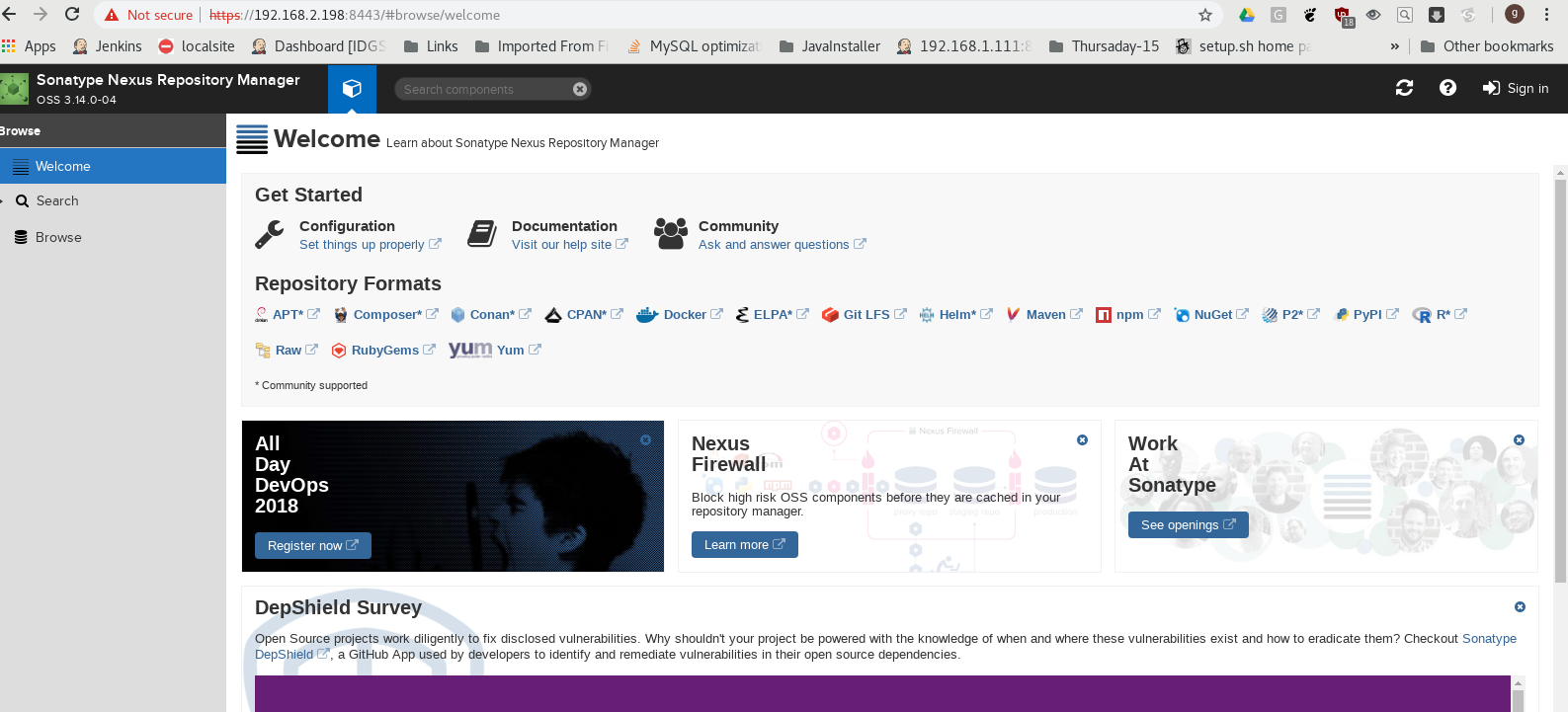


* Now go to browser type the Nexus URL(docker host ip & port number)

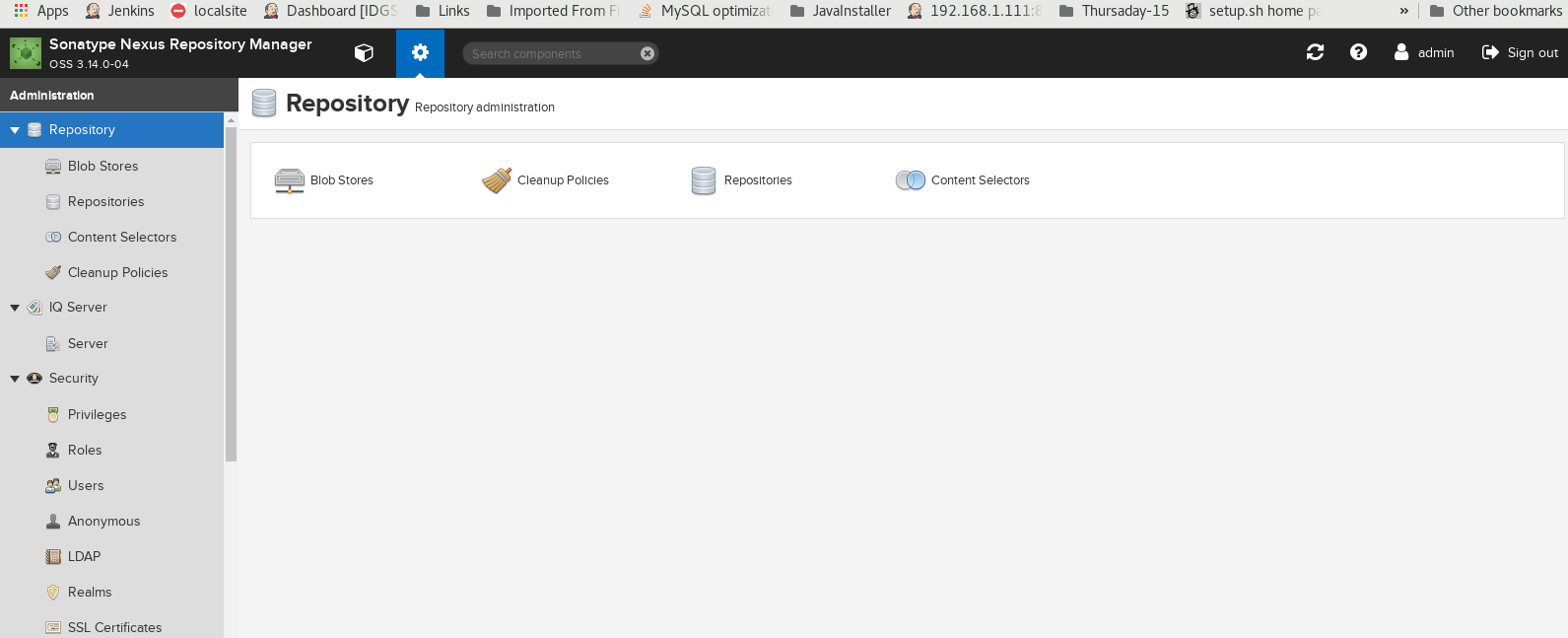
http://<docker-host ip>:8081/

Or

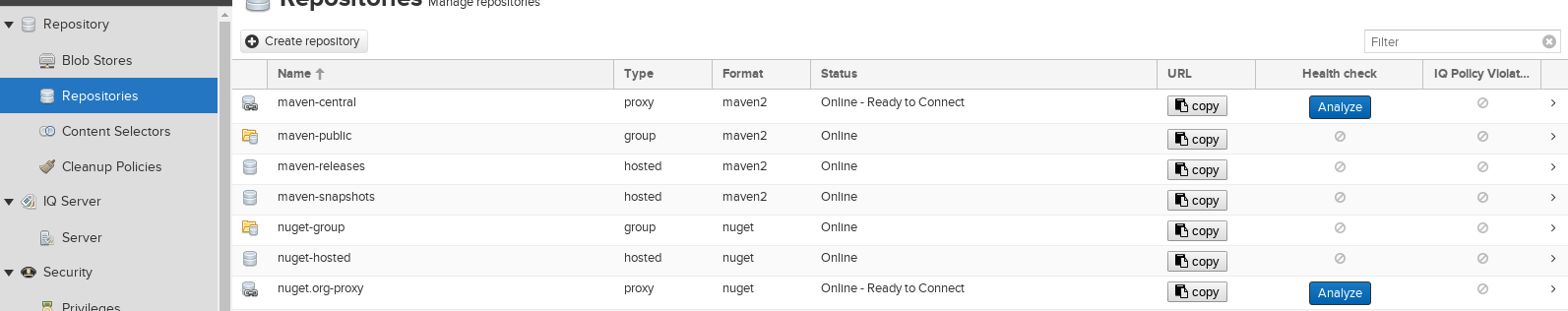
https://<docker-host ip>:8443



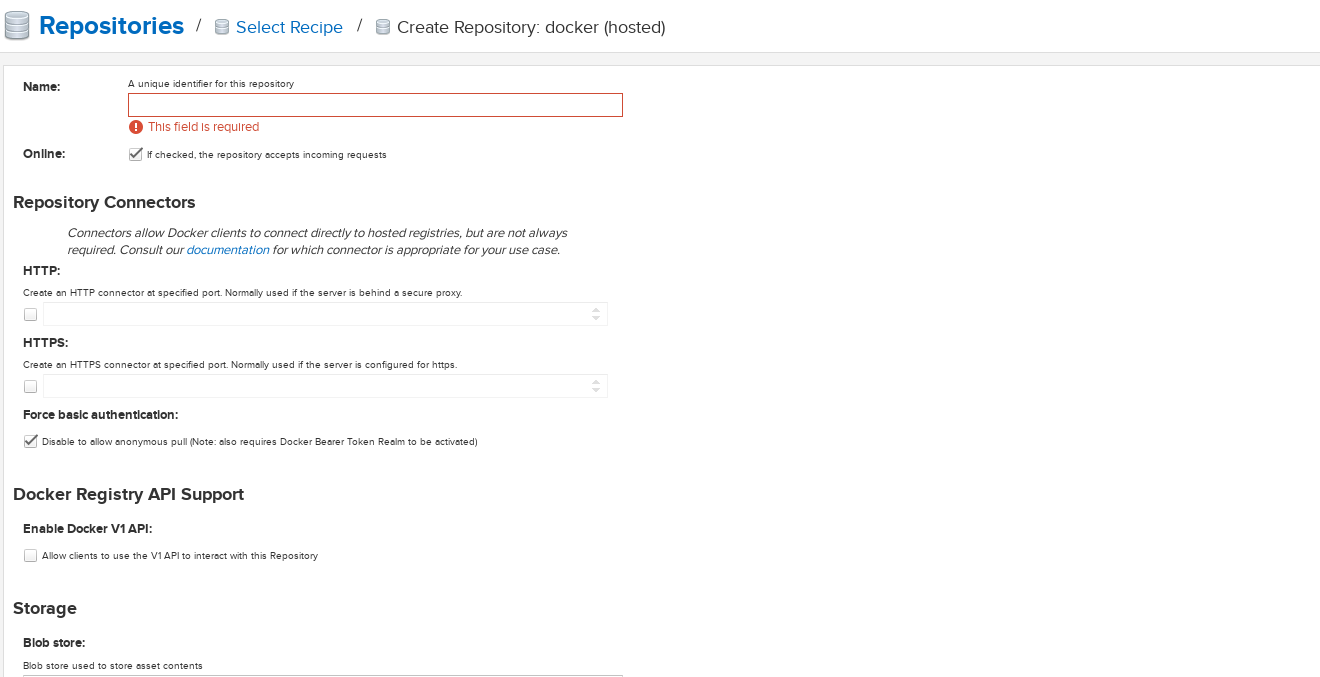
* Now log-in to Nexus
  + User Name: admin
  + Password: admin123
* After log-in to Nexus Repository Manger, now go to setting in the portal as show below



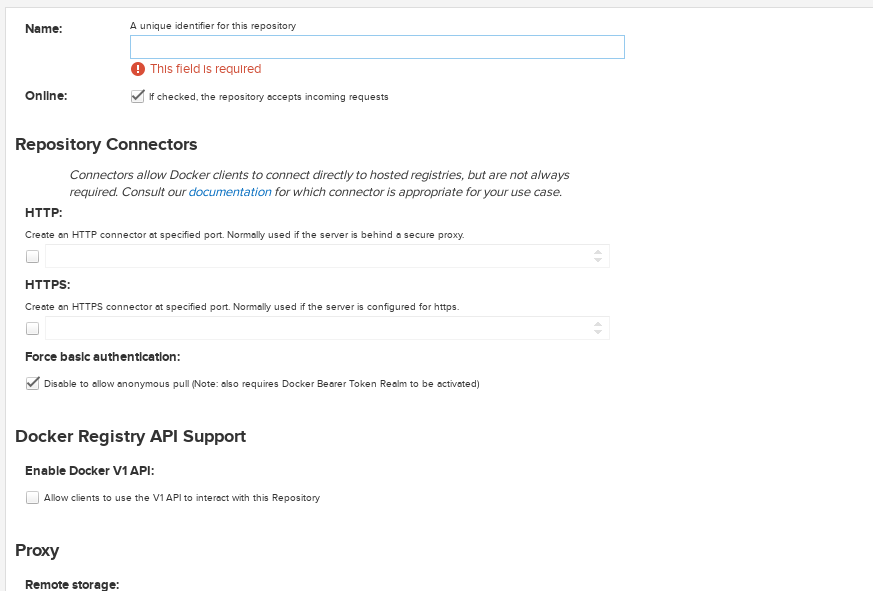
* Now go to Repository & settle ‘create repository’ button on the top



* Now select docker(hosted) recipe

Full the below details

* + - * + Name : docker-private
        + Check the HTTP & port no: 8083
        + Check the HTTPS & port no: 18083
        + Uncheck the Enable Docker V1 API Support
        + Now click on save to save the repositorie
* Now select docker(proxy) recipe



Full the details show next page

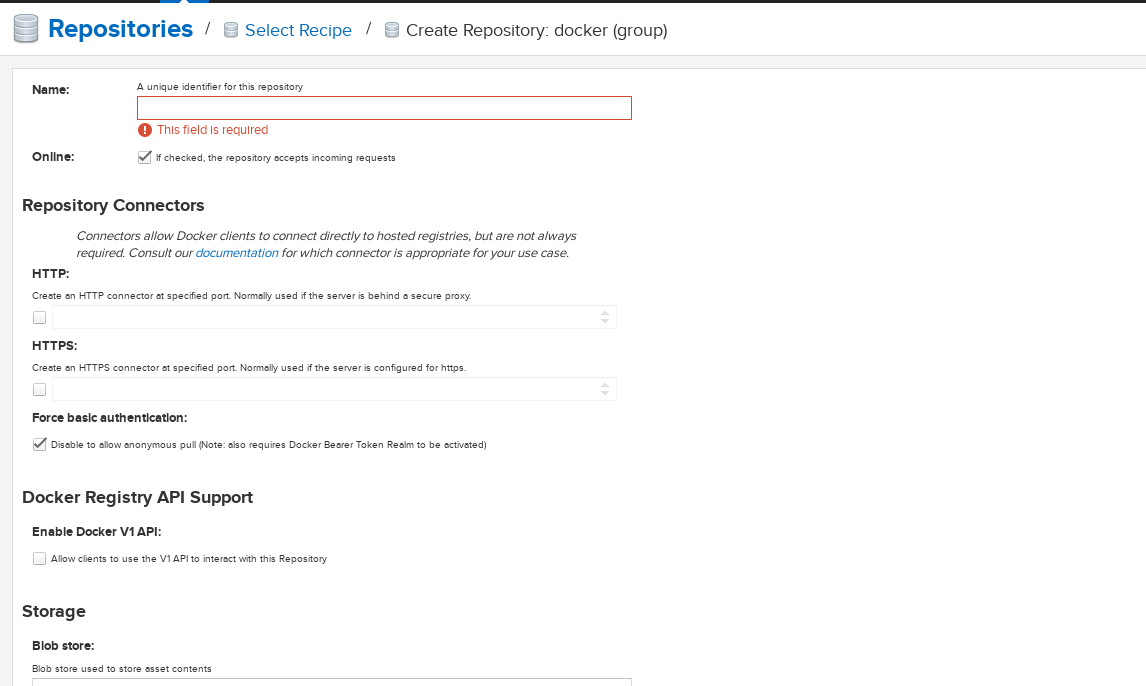
* + - * + Name : docker-hub
        + Uncheck the Enable Docker V1 API Support
        + In Proxy

Remote storage: [https://registry-1.docker.io](https://registry-1.docker.io/)

Check the ‘use the Nexus truststorage’

In docker Index select ‘Use Docker Hub”

* + - * + Now click on save to save the repositorie
* Now select docker(group) recipe



Full the below details

* + - * + Name : docker-group
        + Check the HTTP & port no: 8082
        + Check the HTTPS & port no: 18082
        + Uncheck the Enable Docker V1 API Support
        + In Group Member repositories select the both Docker repo “docker-hub & docker-private
        + Now click on save to save the repositorie

**3. Docker sever & client**

* Verify able to reach the the nexus repository server from the docker server (or) client by below docker command

# docker login -u admin -p admin123 <host-name or IP>:<port>

For example

# docker login -u admin -p admin123 192.168.2.171:8081

# docker login -u admin -p admin123 192.168.2.171:8082

# docker login -u admin -p admin123 192.168.2.171:8083

* Tag the Docker Image

# docker tag <image Id> <host-name or IP>:<port>/<image name>

* To push the Docker Image to Nexus repository

# docker push 192.168.2.171:8083/<image name>

* To Pull the Docker Image from Nexus repository to docker server or client

# docker pull 123 192.168.2.171:8082/<image name>

* To run or start the Docker container

# docker run –it --rm -p port:port --name <app name> 192.168.2.171:8082/<image name>

* To check docker container status

# docker ps (for current running container)

# docker ps -a (all container)

# docker images (to check the docker images)

* To stop & rm the docker container

# docker stop <container name> (to stop the docker container)

# docker rm <container name> (to delete the docker container)

# docker rmi <docker image ID> (to delete the docker image)

**4.Docker tomcat image to Nexus repository:-**

* Before Push the Tomcat image to Nexus Repository, first build & tag the Image

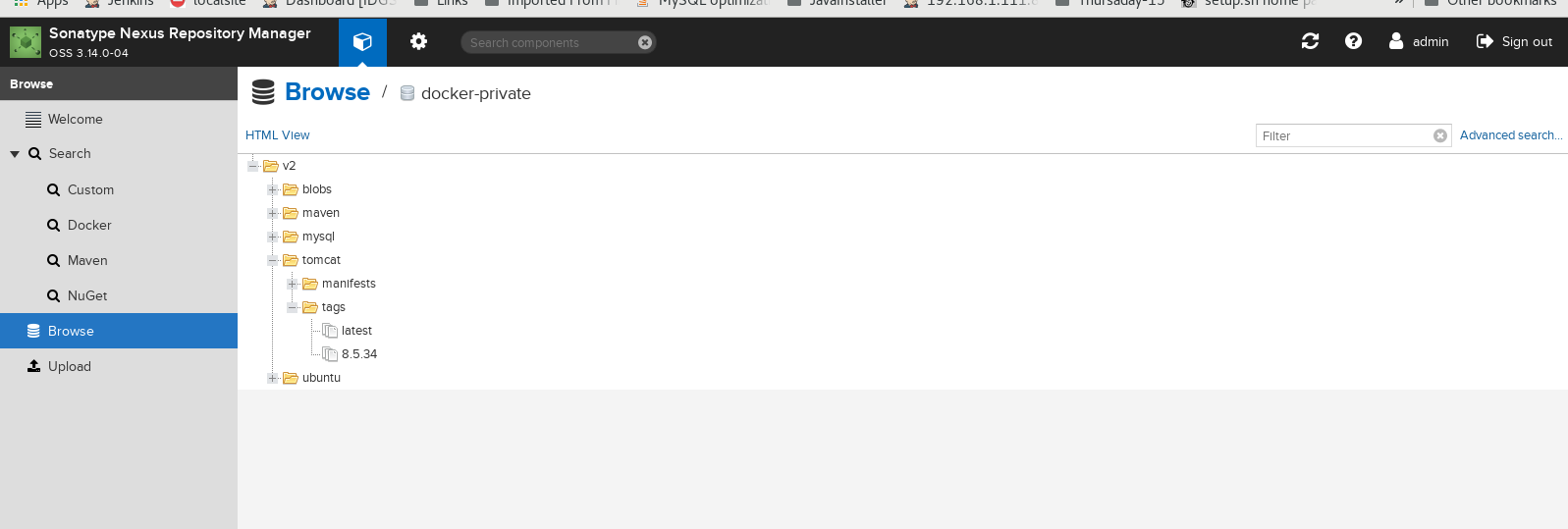
# docker pull tomcat:latest (pull the tomcat latest (8.5.34) image from the Docker Hub)

# docker tag tomcat:latest <nexus IP:8083 or Nexus URL>/tomcat

* + #docker tag tomcat:latest 192.168.2.171:8083/tomcat
* Now push the Tomcat image to Nexus Repository (when push the any image to Nexus Repository to docker-hosted repository only)

# docker push 192.168.2.171:8083/tomcat

* Now check in Nexus Repository for the Docker Image is push or not



* To pull the image on Docker server or client server, first verify able to reach the the nexus repository server

# docker login -u admin -p admin123 192.168.2.171:8082

Note:- ones the verify the authentication details in auto save in below path

# cat /root/.docker/config.json

{

"auths": {

"192.168.2.171:8082": {

"auth": "YWRtaW46YWRtaW4xMjM="

},

"192.168.2.171:8083": {

"auth": "YWRtaW46YWRtaW4xMjM="

},

"localhost:8082": {

"auth": "YWRtaW46YWRtaW4xMjM="

}

},

"HttpHeaders": {

"User-Agent": "Docker-Client/18.09.0-beta3 (linux)"

}

}

Note:- the verification of the Nexus repository access for first only when the docker server user the Nexus repository only.

* After the verification is done, now pull the the tomcat image to the docker service

# docker pull 192.168.2.171:8082/tomcat

Or

* Run the tomcat services

# docker run -d -it -rm --name <app name> 192.168.2.171:8082/tomcat

* To run the tomcat with different port

# docker run -it -rm -p <anyport>:8080 --name <app name> 192.168.2.171:8082/tomcat

* To deploy the war files to tomcat

# docker cp <part of the war file>/\*.war <tomcat con>:/usr/local/tomcat/webapps/

Note:- we can deploy the war or jar file user Dockerfile(we contain or to build the image for any change of tomcat setting as per the application & port change, with war or jar file )

(Exp of Dockerfile:-

FROM 192.168.2.171:8082/tomcat

RUN ln -sf /usr/local/tomcat/

COPY root /

ENV APP\_NAME="tomcat" \

IMAGE\_VERSION="8.5.34" \

JAVA\_OPTS="-Djava.awt.headless=true -XX:+UseG1GC -Dfile.encoding=UTF-8" \

PATH="/usr/local/java/bin:/usr/local/tomcat/bin:$PATH" \

TOMCAT\_AJP\_PORT\_NUMBER="8009" \

TOMCAT\_ALLOW\_REMOTE\_MANAGEMENT="0" \

TOMCAT\_HTTP\_PORT\_NUMBER="8090" \

TOMCAT\_PASSWORD="" \

TOMCAT\_SHUTDOWN\_PORT\_NUMBER="8005" \

TOMCAT\_USERNAME="user"

EXPOSE 8090

ADD \*.war /usr/local/tomcat/webapps/

ENTRYPOINT [ "/app-entrypoint.sh" ]

CMD [ "nami", "start", "--foreground", "tomcat" ]

)

**5.Docker Maven image to Nexus repository:-**

* Before Push the maven image to Nexus Repository, first build & tag the Image

# docker pull maven:latest (pull the maven latest version image from the Docker Hub)

# docker build maver:latest .(with Dockerfile only)

# docker tag maven:latest <nexus IP:8083 or Nexus URL>/maven

* + #docker tag tomcat:latest 192.168.2.171:8083/tomcat
* Now push the maven image to Nexus Repository (when push the any image to Nexus Repository to docker-hosted repository only)

# docker push 192.168.2.171:8083/maven

* Now pull the maven image