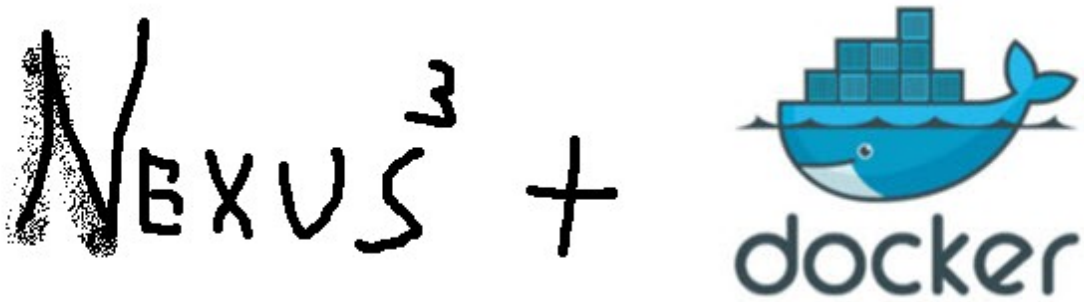
ivankrizsan.se

Create a Private Docker Registry – The Blog of Ivan Krizsan

8-10 minutos



Nexus 3 and Docker – I couldn't use the Nexus 3 logo, so I drew one myself!

In this article I will show how to set up a private Docker registry with Sonatype's Nexus Repository Manager 3.0 OSS. There may be other guides to this, but if nothing else I am writing for myself.
The main motivations for me to set up a private Docker registry are:

- Distribution of Docker images.
  This may be either small-scale between multiple computers or inside an organization.
- Restrict distribution of Docker images.
  While I prefer to make my Docker images available at DockerHub, there are cases when I feel that the Docker images are not ready for distribution to a larger audience.
  In an organization, it may undesirable or unsuitable to deploy

Docker images to a public registry.

- Use identical versions of Docker images on all my computers.
  If I build a Docker image on first one computer and then, at some later point in time, on another computer there may be differences between the images. This can be caused by updates to the software repositories used when creating the images, patches to the operating system etc. Such differences may be subtle but I prefer to eliminate as many potential sources of confusion, or even errors, that I possibly can.

  Since I am already fiddling around with Docker, I will run Nexus 3 OSS in a Docker container using the Docker image from Sonatype.

  As an example, I will make a copy of the official Tomcat Docker image to my own, private, Docker registry. To do this, we first need to pull the Tomcat image:

```
docker pull tomcat:latest
```

  In addition I will create a directory in the host to hold Nexus 3 data and logs, as described in the Nexus 3 image's documentation:

```
mkdir -p /some/dir/nexus3/data

chown -R 200 /some/dir/nexus3/data
```

  In order to simplify start of the Nexus 3 Docker container, I will use Docker Compose.

- Go to the "nexus3" directory.
  This is the parent directory of the "data" directory that we created as part of the preparations above.

- Create a file named "docker-compose.yml" with the following

content:

```
nexus:
    image: sonatype/nexus3:latest
    ports:
        - "8081:8081"
        - "8123:8123"
    volumes:
        - ./data:/nexus-data
```

Note that:

- Two ports are exposed.
  On port 8081 the web GUI of Nexus 3 will be available. The Docker registry API will be exposed on port 8123. The latter port may be changed as desired, just remember to configure Nexus 3 accordingly.
- The shared directory “nexus-data”, which is mapped to the host directory “data”.
  This directory will contain the persistent data of Nexus 3.

  In a terminal window, the Docker Quickstart terminal if you are using Docker Toolbox, with the current directory being the directory with the docker-compose.yml file we created earlier, launch the Nexus 3 Repository manager using this command:

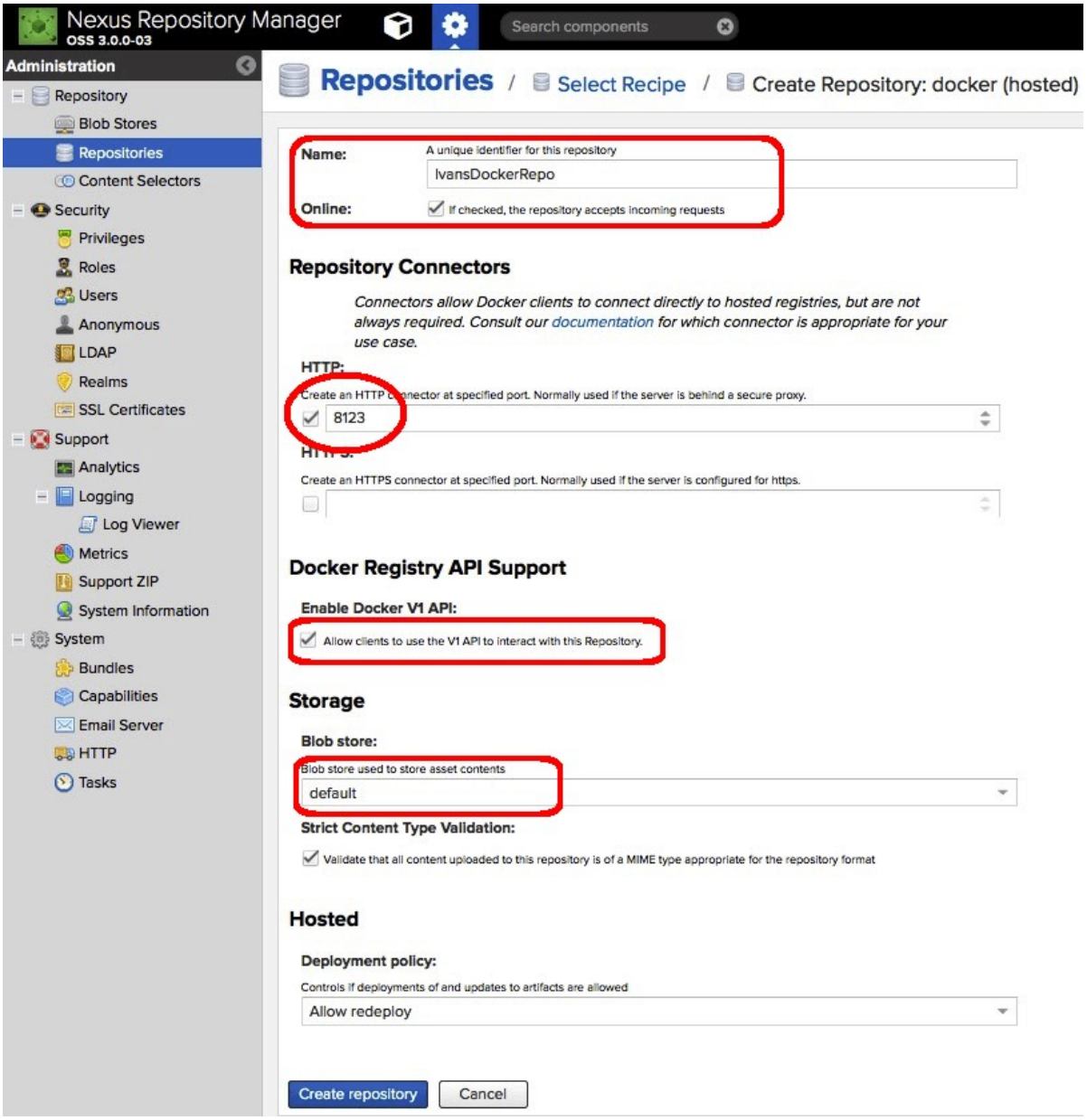
  If you are running Docker in Linux, you’ll have to add “sudo” before the above command.
  Then use the logs command to view the logs from the Nexus Docker container:

After some time, log output similar to this should appear in the Nexus log, indicating that Nexus has successfully started up:

```
nexus_1 | -------------------------------------------------------------------
nexus_1 |
nexus_1 | Started Sonatype Nexus OSS 3.0.0-03
nexus_1 |
nexus_1 | -------------------------------------------------------------------
```

With Nexus 3 up and running, we should now be able to create a Docker registry.

- Configure the Docker repository.
  Give the repository a name – in my case it is “IvansDockerRepo”.
  Make sure that the Online checkbox is checked.
  Check the HTTP checkbox under Repository Connectors and enter the port number 8123.
  Check the Enable Docker V1 API checkbox.
  Select default under Blob store.
- Click the button Create repository.



The private Docker registry is now ready to be used.

Before we can interact with the Docker registry from a Docker client, we need to log into the registry.

– Allow Unsecure Connections Windows and OS X

Since we have exposed the private Docker registry on a plain HTTP endpoint, we need to configure the Docker daemon(s) that will act as client(s) to the private Docker registry as to allow for unsecure connections. This may not be advisable in all

environments, but for the sake of this example unsecure connections are simple and will suffice.

- Locate the Docker config.json file that contains the key “InsecureRegistry”.
  On Mac OS X and assuming that the name of the Docker machine is “default”, it is located in the directory ~/.docker/machine /machines/default.
  On Windows, again assuming the name of the Docker machine is “default”, it is located in the directory C:\Users\[your user name]\.docker\machine\machines\default.
  The section of the configuration file in which the “InsecureRegistry” key can be found looks like this:

```
1   "HostOptions": {
2       "Driver": "",
3       "Memory": 0,
4       "Disk": 0,
5       "EngineOptions": {
6           "ArbitraryFlags": [],
7           "Dns": null,
8           "GraphDir": "",
9           "Env": [],
10          "Ipv6": false,
11          "InsecureRegistry": [],
12          "Labels": [],
13          "LogLevel": "",
```

```
14          "StorageDriver": "",
15          "SelinuxEnabled": false,
16          "TlsVerify": true,
17          "RegistryMirror": [],
18          "InstallURL": "https://get.docker.com"
19      },
```

- Insert the IP on which the Nexus 3 GUI is exposed along with the port number 8123 to the value of the key “InsecureRegistry”. In my case, the modified key-value will look like this:

```
"InsecureRegistry": [“192.168.1.72:8123”],
```

- If you are using the Docker for Mac or Windows Beta:
  New method (make sure you have the latest update): Insecure registries can be configured in the Docker Preferences, in the Advanced tab.
  ~~Old method: Create a file named config.json with this contents (replace the IP address as needed):~~
  ~~{“storage-driver”:”aufs”,”debug”:true,”insecure-registries”:~~
  ~~[“192.168.1.72:8123”]}~~
  ~~The set the configuration using the commando:~~
  ~~pinata set daemon @conf.json~~
- Restart the Docker machine.

– Allow Unsecure Connections Linux

To allow for unsecure connections from a Docker client running on Linux:

- Edit the file /etc/default/docker and append the following line: DOCKER_OPTS="$DOCKER_OPTS –insecure-registry=[Nexus3 Docker host IP]:8123″
- Restart the Docker service using: sudo service docker restart

– Log in to the Private Docker Registry

You should now be able to log in to the Nexus 3 registry using the following command (replace IP address as necessary) using the username “admin” and the password “admin123”, both without quotes:

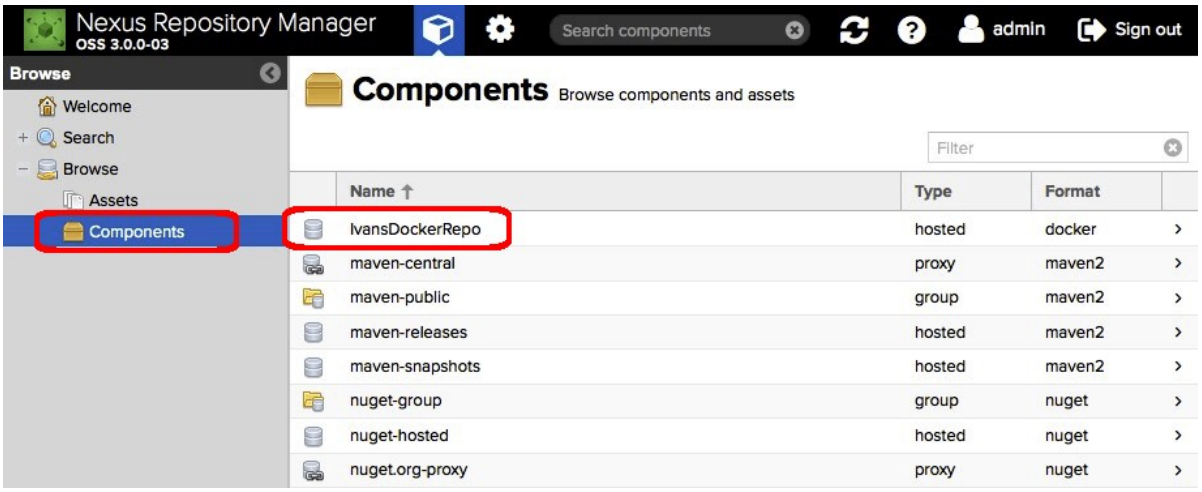
```
docker login -u admin -p admin123 192.168.1.72:8123
```

The result should be a message saying that login succeeded.

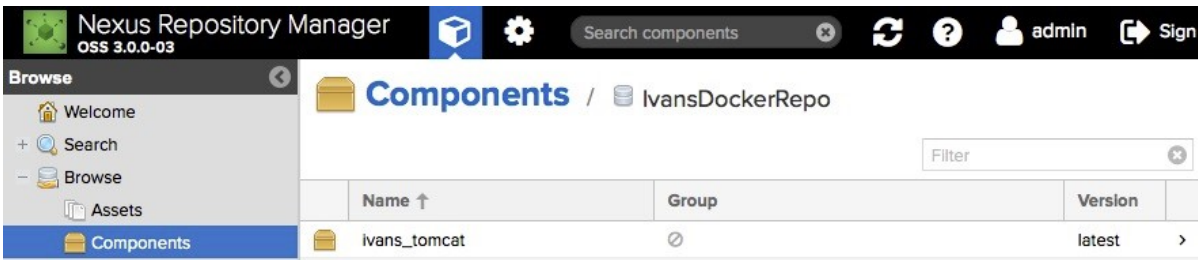
Having pulled the official Tomcat image earlier and having logged in to our private registry, we are now ready to push the Tomcat Docker image to our registry. In the instructions below, replace the IP address of the Docker registry as needed.

- Tag the image.
  docker tag tomcat:latest 192.168.1.72:8123/ivans_tomcat:latest
  This tells Docker the new name that will be used for the Docker image and the IP address of the registry to which we later will push the image.
- Push the image to our private registry.
  docker push 192.168.1.72:8123/ivans_tomcat:latest
- Open the URL http://192.168.1.72:8081 in a browser.
- Click the Components category in the Browse column on the left.
- Click the Docker registry with the name “IvansDockerRepo” in the

list of components that appear.



- In the list of components in IvansDockerRepo, there should be one single entry with the name “ivans_tomcat”.



- To pull the image, perhaps on another computer, use the following commands.

```
docker login -u admin -p admin123 192.168.1.72:8123

docker pull 192.168.1.72:8123/ivans_tomcat:latest
```

This concludes this article.

Happy coding!