



**Certified Tech
Developer**

The Ultimate Degree

Infraestructura II

Actividad obligatoria e individual

Dificultad: media

Continuous Integration con Jenkinsfile

Parte 4.1

Nexus Repository

En esta ocasión vamos a configurar una instancia de **Sonatype Nexus 3** para utilizarla como repositorio de artefactos en la próxima práctica.

Nuestro desafío es:

- Correr un contenedor con Nexus3.
- Configurar un usuario.
- Configurar un repositorio de artefactos.

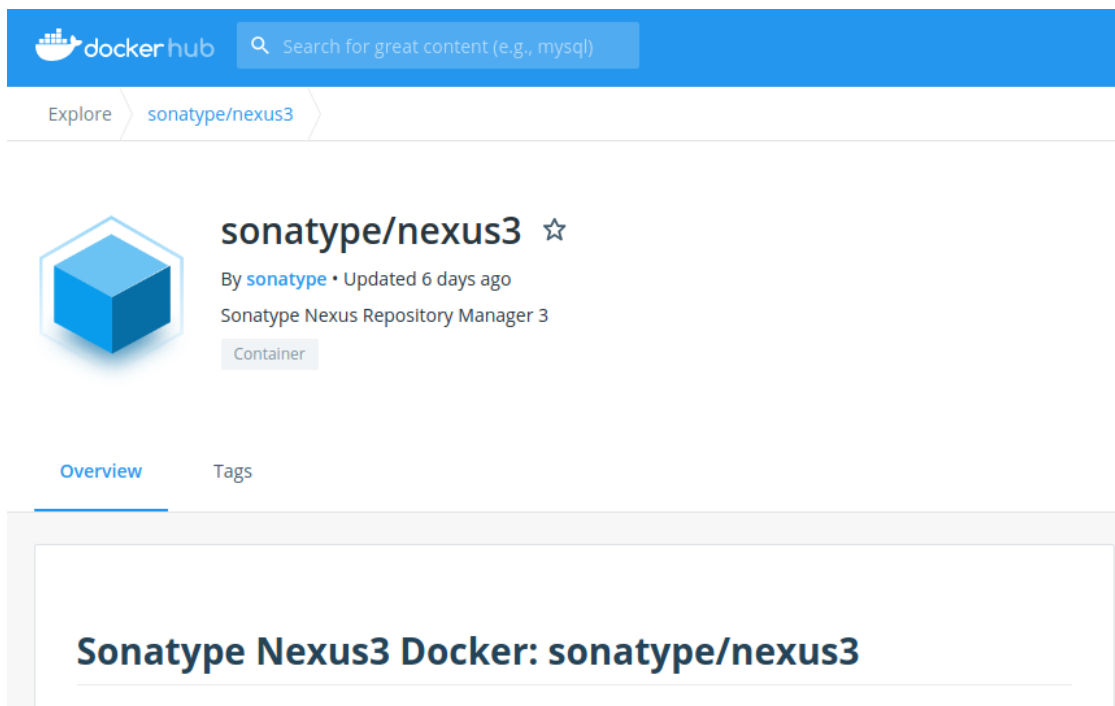
En la siguiente página se encuentra la resolución. Continúa únicamente para realizar una autoevaluación.

Resolución

- **Correr un contenedor con Nexus3**

Para levantar el contenedor, haremos uso de docker-compose. Lo primero que necesitamos es la imagen de nuestro contenedor.

Buscamos **nexus3** en **dockerhub.com** y hallamos el siguiente repositorio:



En la sección de **tags** buscamos uno reciente y escribimos lo siguiente en nuestro docker-compose:

```
version: '3'

services:

  nexus:

    container_name: nexus
```



```
image: 'docker.io/sonatype/nexus3:3.33.1'

user: 'nexus'

environment:

  BASE_URL: 'http://localhost:8081'

  NEXUS_SECURITY_RANDOMPASSWORD: 'false'

ports:

  - '8081:8081'

ulimits:

  nofile:

    soft: 65536

    hard: 65536

volumes:

  - './nexus_data:/nexus-data'
```

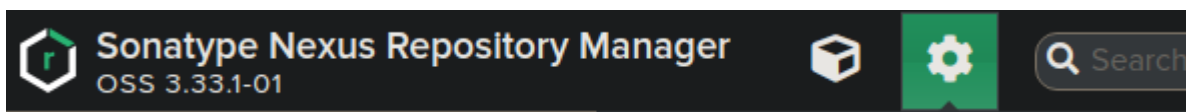
Levantamos nuestro archivo corriendo "docker-compose up"

- **Configurar un usuario**

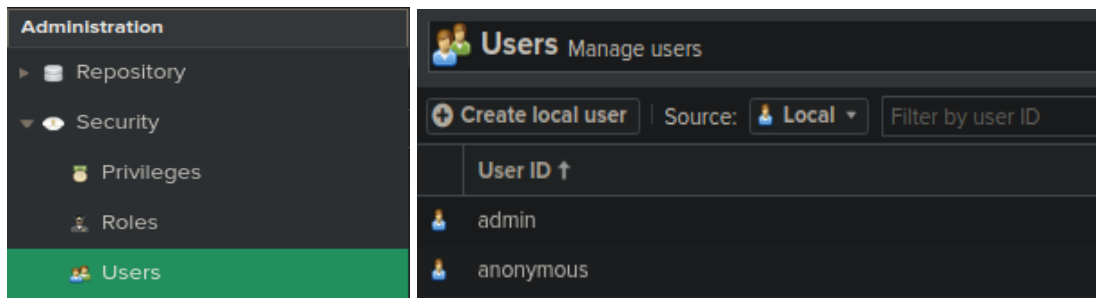
Para configurar un usuario ingresamos a Nexus (<https://localhost:8081>):

- usuario: admin
- contraseña: admin123

Accedemos al icono de **configuración**:

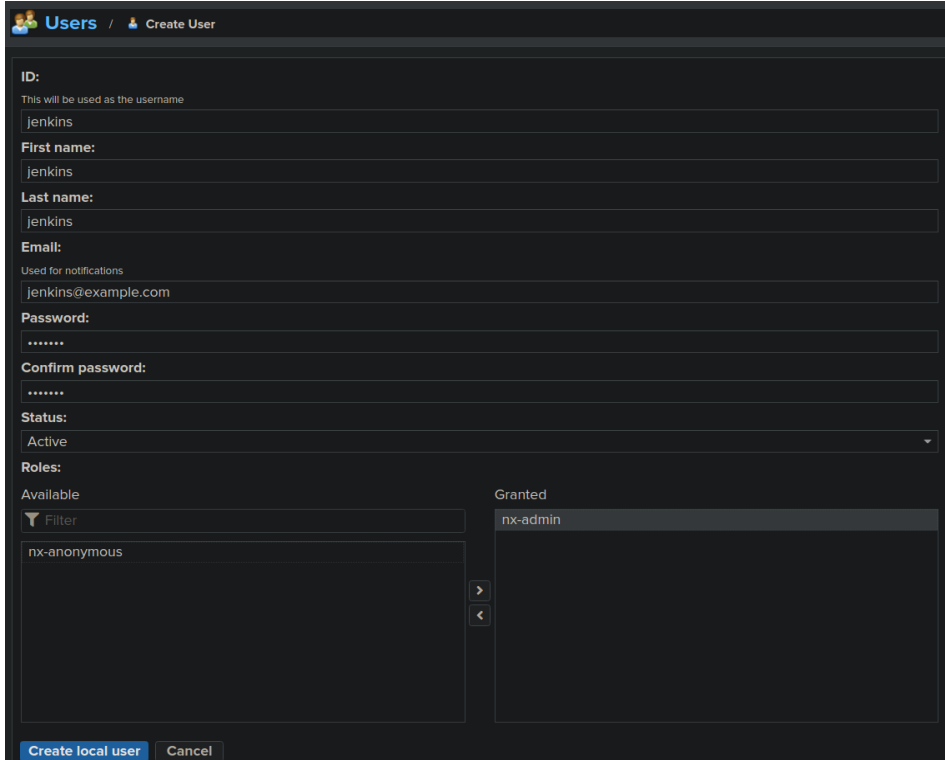


Nos dirigimos al menú. En **Security** hacemos clic en **Users** para cargar la lista de usuarios y tener la posibilidad de generar uno nuevo. Así lo veremos:



Damos clic en **Create user** y cargamos los datos de nuestro usuario. Establecemos el estado como activo y agregamos el rol **nx-admin**.

Para terminar el proceso de creación de usuario, hacemos clic en **Create local user**. Aclaración: el rol nx-admin se usa a modo de prueba, aquí deberíamos ajustar los permisos.



Users / Create User

ID:
This will be used as the username
jenkins

First name:
jenkins

Last name:
jenkins

Email:
Used for notifications
jenkins@example.com

Password:
.....

Confirm password:
.....

Status:
Active

Roles:

Available

Filter

nx-anonymous

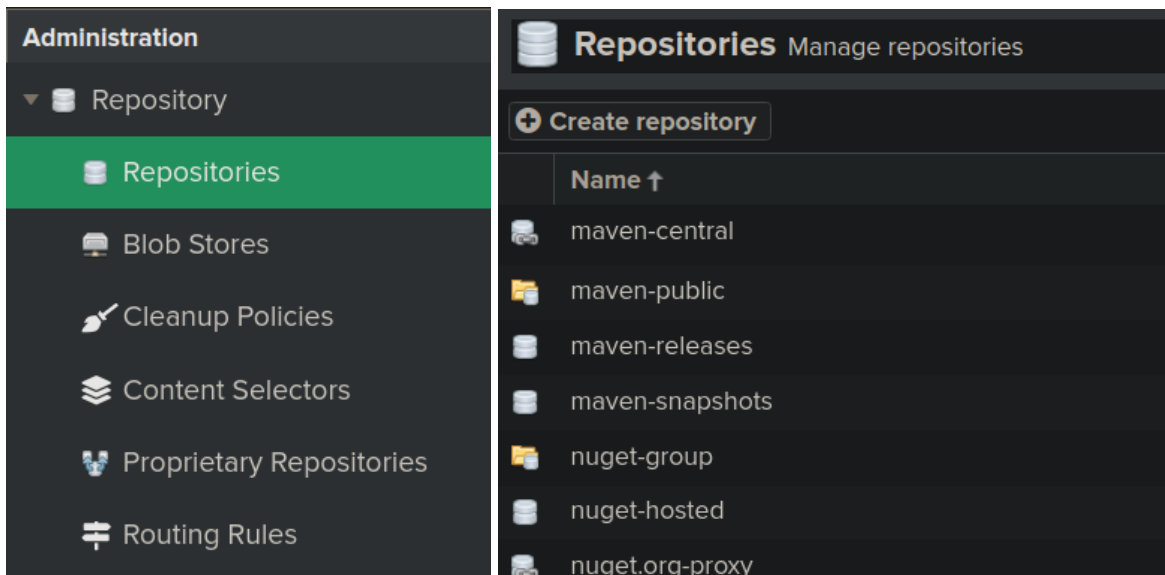
Granted

nx-admin

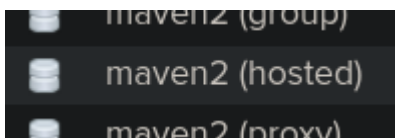
Create local user Cancel

- **Configurar un repositorio de artefactos**

Continuamos en Nexus. Esta vez vamos a la sección de **Repository** y hacemos clic en **Repositories**. Esto cargará a la derecha la lista de repositorios por defecto:



Hacemos clic en **Create repository**. Veremos una lista de “recipes” que indican tipo y formato del repositorio a crear. Vamos a elegir **maven2 (hosted)**:



Luego completamos los datos que nos solicita el formulario tal y como vemos en la siguiente imagen:



Name: A unique identifier for this repository

Online: ☒ If checked, the repository accepts incoming requests

Maven 2

Version policy:
What type of artifacts does this repository store?

Layout policy:
Validate that all paths are maven artifact or metadata paths

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

Storage

Blob store:
Blob store used to store repository contents

Strict Content Type Validation:
☒ Validate that all content uploaded to this repository is of a MIME type appropriate for the repository format

Hosted

Deployment policy:
Controls if deployments of and updates to artifacts are allowed

Proprietary Components:
☐ Components in this repository count as proprietary for namespace conflict attacks (requires Sonatype Nexus Firewall)

Conclusión

En esta práctica vimos cómo levantar en un contenedor con docker-compose un gestor de artefactos como Nexus3 buscando la imagen en el catálogo de DockerHub. Además, realizamos una configuración básica, creando un usuario y un repositorio.

¡Ya tenemos todo listo para la próxima práctica! ¡Seguimos!