

# artifact repository - nexus



artifact repository, artifact repository.

artifact repository artifact repository (Refactoring) artifact repository.

Devop pipeline artifact (artifact) artifact artifact artifact Sonatype artifact Nexus JFrog artifact jfrog artifact.

artifact artifact artifact artifact artifact artifact artifact Nexus artifact

artifact.

## Nexus

1. NeXus Repository Manager (artifact ;NXRM)
2. artifact OS : Windows / Linux / Unix / OSX
3. artifact : 3.29.2 OSS ('21.01.10 artifact )
4. License : Eclipse Public License-v 1.0
5. Pro artifact artifact : Atalssian Crowdartifact , Staging / Build artifact / artifact / artifact artifact / HA / artifact OSS artifact artifact / artifact / artifact artifact artifact / SAML / SSO artifact

## Nexus artifact artifact

1. hardware requirements

Type	Minimal	requirement
CPU	4Core	8Core
Memory	Host : 8Gb JAVA Heap : 2Gb	Host : 8Gb Java Heap : 4GB
Disk	330M	330M

- 4GB Readonly
- Java Directory ( \* 0.6 ) - java heap

## 2.

	Embbd	Blob	
XFS	O	O	
NFS V4	O	O	
NFS V3	X	X	
AWS - EBS	O	O	
AWS - EFS	X	O	Embbd
AWS - S3	X	O	
AWS- S3	X	X	S3
Samba/CIFS	X	O	
GlusterFS	X	X	Split-brain ,
Fuse	X	X	

\* Embbd : (DB / Elastic Search)

\* Blob : binary

3. 目录结构

目录	说明
bin	可执行文件 / 脚本
etc	nxrm 配置文件 / 目录
lib	nxrm 依赖库
public	nxrm 公共文件
system	系统文件 / 目录
blobs	blob 文件
cache	缓存文件
db	orientDB 数据库
elasticsearch	elasticsearch 索引
health-check	健康检查脚本
keystores	密钥库文件
log	日志文件 nexus.log → 日志文件 , request.log → http 请求日志 jvm.log → jvm stdout, stderr 日志 karaf.log → apache karaf 日志 audit → 审计日志 staks → 堆栈跟踪 tmp → 临时文件

# Nexus 配置

1. 配置项

- hosted : 本地仓库
- proxy : hosted 代理仓库 cache 缓存

- group : groupId hosted or proxy groupId groupId

## 2. groupId groupId groupId

- release : groupId groupId / groupId groupId groupId groupId (groupId groupId groupId groupId )
- snapshot : groupId groupId groupId groupId / groupId groupId groupId groupId groupId

## 3. groupId groupId

- maven2 : groupId groupId groupId groupId (group / artifact id / version groupId groupId )
- raw : groupId groupId groupId groupId groupId groupId
- yum : yum groupId groupId

## 4. groupId groupId groupId

- curl groupId groupId groupId
- test\_repo groupId img-1.png groupId

```
$ curl -u 'test:test123' http://192.168.10.10:11000/repository/test_repo/img-1.png -o img-1.png
```

- test.tar.gz groupId test\_repo groupId

```
$ curl -u 'test:test123' --upload-file
./test.tar.gz http://192.168.10.10:11000/repository/test_repo/test.tar.gz
```

- groupId groupId

```
$find ./ -exec curl -u 'test:test123' -T
./{} http://192.168.10.10:11000/repository/test_repo/test.tar.gz/{} \;
```

- test\_repo groupId test.tar.gz groupId

```
$> curl --request DELETE -u
'test:test123' http://http/192.168.10.10:11000/repository/test_repo/test.tar.gz
```

# Nexus groupId (Docker groupId )

## 1. container pull ☐☐

```
$> docker pull sonatype/nexus3:3.29.2 #2.x ☐☐ ☐☐ sonatype/nexus ☐☐☐☐ ☐
```

## 2. nexus ☐☐☐ ☐☐☐ ☐☐☐☐

```
$> mkdir /data/nexus-repo  
$> chown -R 200 /data/nexus-repo/
```

## 3. container ☐☐☐☐

```
$ docker run -d -p 88081:8081 \  
-e TZ=Asia/Seoul \  
-v /etc/localtime:/etc/localtime:ro \  
-v /data/nexus-repo:/nexus-data \  
-name nexus-repo sonatype/nexus3:3.29.2
```

## 4. Container log ☐☐ (started sonatype ☐☐☐☐ ☐☐☐☐ ☐☐☐☐ ☐☐ )

```
$ docker logs -f nexus-repo  
...  
2021-01-19 23:32:02,793+0900 INFO [jetty-main-  
1] *SYSTEM org.sonatype.nexus.repository.httpbridge.internal.ViewServlet - Initialized  
2021-01-19 23:32:02,870+0900 INFO [jetty-main-  
1] *SYSTEM org.eclipse.jetty.server.handler.ContextHandler -  
Started o.e.j.w.WebAppContext@637c45d9{Sonatype Nexus,/file:~/~/opt/sonatype/nexus/public/,A  
VAILABLE}  
2021-01-19 23:32:02,928+0900 INFO [jetty-main-  
1] *SYSTEM org.eclipse.jetty.server.AbstractConnector -  
Started ServerConnector@45bb642d{HTTP/1.1, (http/1.1)}{0.0.0.0:8081}  
2021-01-19 23:32:02,929+0900 INFO [jetty-main-1] *SYSTEM org.eclipse.jetty.server.Server -  
Started @41677ms  
2021-01-19 23:32:02,930+0900 INFO [jetty-main-
```

```
1] *SYSTEM org.sonatype.nexus.bootstrap.jetty.JettyServer -
```

```
~~~~~  
~~~~~  
\\Started Sonatype Nexus OSS 3.29.2-02  
\\~~~~~  
~~~~~
```

5. web-ui  – http://{ IP}:88081/

6.

7.

```
$docker exec -it nexus cat /nexus-data/admin.password
```

```
1123-aad-xxxx-xxxx-
```

8.   ( admin)

9. admin   (    Next)

10.

## Repository Cleanup

1. cleanup policies

- → Repository → Cleanup Policies → Create Cleanup Policy
- Cleanup policy
- (  90     )

Field	Value	Value	Value
Name	Repository name	proxy_cleanup_{{ format }}d	Repository name
Format	Repository format	raw	Repository format all formats
notes	Repository notes		
Component Age	Repository age	90	Repository age
Component Usage	Repository usage	90	Repository usage
preview repository	Repository preview		

Repository

1. Hosted

- Repository → Create Repository → RAW (hosted)
- repository
- 

Field	Value	Value	Value
Name	Repository	repo	Repository name ( -, _ )
Online	repository		Repository online
Content Disposition	Repository content disposition	attachment	Attachment / inline 1, Attachment
Blob store	Repository blob store	default	Repository blob store default

Item	Item	Item	Item
Deployment Policy	Repository / Repository	Allow redeploy	Allow redeploy : Deny redeploy : read only : ,
Cleanup policies	Repository Repository		Cleanup Repository

2. Proxy Repository

- Repository → Create Repository → RAW (Proxy)
- repository
- 

Item	Item	Item	Item
Name	Repository		, ( -, _ ) Repository
Online	repository		Repository
Remote storage	hosted		
Blocked	Proxy		
Auto blocking enabled	Remote Storage Proxy		
Maximum component age	cache ( )	60	



Property	Default	Value	Notes
Maximum metadata age	cache metadata (seconds)	60	
Blob store	default	default	Storage provider
Strict Content Type Validation	strict	strict	
Not found cache enabled	enabled	enabled	
Not found cache TTL	60	60	
Cleanup policies	Repository		Cleanup
HTTP authentication	host	host	

3. Proxy Repo cache

- Repository → Proxy Repository → Invalidate cache

Proxy

1. Proxy

Property	Proxy
-1	
0	
1	Proxy repo
2	Proxy Repo

## 2. Proxy Traffic

- `curl -x 192.168.10.10:8081 -c /tmp/cookies.txt https://www.google.com/`  
, caching flush `curl -x 192.168.10.10:8081 --flush https://www.google.com/` (flush `curl -x 192.168.10.10:8081 --flush https://www.google.com/` )

- `tcpdump -i eth0 -s 0 -A host 192.168.10.10 or host 192.168.100.10` (Hosted → Proxy), 192.168.10.10 (Proxy), 192.168.100.10 (Hosted)

```
15:42:31.554851 IP 192.168.10.10.42224 > 192.168.100.10.88081: Flags [S], seq 1521618240, win 29200, options [mss 1460,nop,nop,sackOK], length 0
```

```
15:42:31.555029 IP 192.168.100.10.88081 > 192.168.10.10.42224: Flags [S.], seq 2385004980, ack 1521618241, win 29200, options [mss 1460,nop,nop,sackOK], length 0
```

```
15:42:31.555223 IP 192.168.10.10.42224 > 192.168.100.10.88081: Flags [.], ack 1, win 29200, length 0
```

```
15:42:31.555234 IP 192.168.10.10.42224 > 192.168.100.10.88081: Flags [P.], seq 1:262, ack 1,
```

---

Revision #6

Created 7 June 2022 01:25:55 by artop0420

Updated 1 September 2024 00:19:05 by artop0420