Docker module practical task 2

Practical task 2

Follow the steps below to practice with using remote repositories for storing Docker images and executing security scans:

Prerequisites:

- Use images that were built during Docker practice task
- Get/check access to corporate AWS account and read basic instructions
  - FAQ for interns
  - GD AWS cloud for beginners
  - Tagging policy for Grid Dynamics AWS accounts
- Read documentation about Docker repositories in Nexus
  - Using Nexus 3 as Your Repository – Part 3: Docker Images
- Read documentation about AWS ECR
  - Getting started with Amazon ECR using the AWS Management Console - Amazon ECR
  - Using Amazon ECR with the AWS CLI - Amazon ECR

Task:

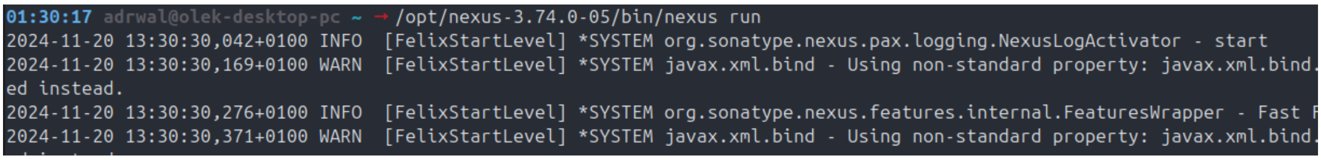
- Uploading docker images to Nexus and ECR
  - Create and configure docker repository in Nexus
  - Upload spring-petclinic image to Nexus
  - Create and configure repository in ECR
  - Upload spring-petclinic image to ECR
- Perform security scan for uploaded images in ECR
  - Get scan report
  - Analyze results

There might be (or not) security vulnerabilities that need to be addressed. Discussion about possible workarounds to get rid of them and overall analysis of reports should be held.

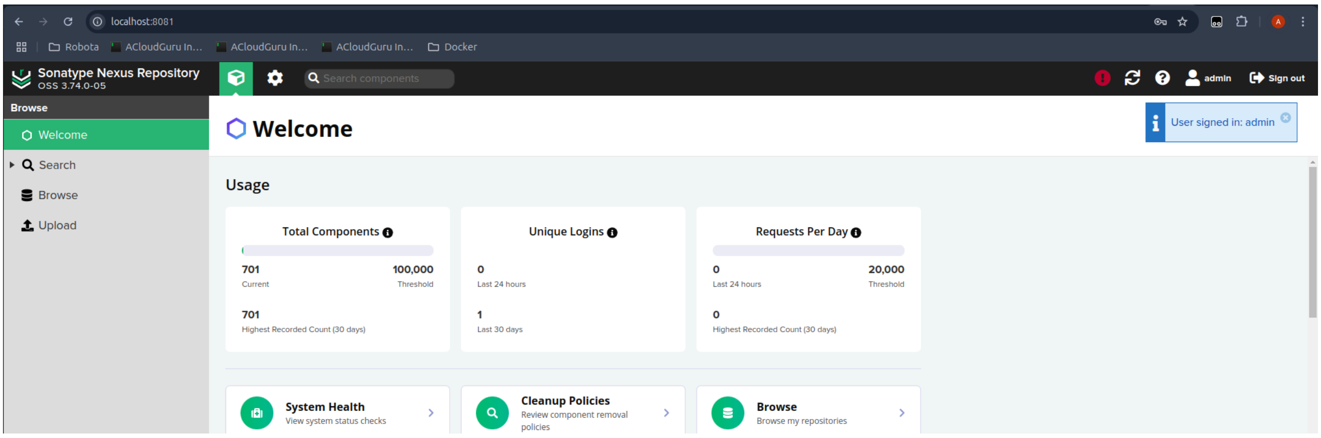
1. Uploading docker images to Nexus and ECR

a. Nexus

First we need to start our local Nexus server.



```
01:30:17 adrwal@olek-desktop-pc ~ → /opt/nexus-3.74.0-05/bin/nexus run
2024-11-20 13:30:30,042+0100 INFO  [FelixStartLevel] *SYSTEM org.sonatype.nexus.pax.logging.NexusLogActivator - start
2024-11-20 13:30:30,169+0100 WARN  [FelixStartLevel] *SYSTEM javax.xml.bind - Using non-standard property: javax.xml.bind.
ed instead.
2024-11-20 13:30:30,276+0100 INFO  [FelixStartLevel] *SYSTEM org.sonatype.nexus.features.internal.FeaturesWrapper - Fast F
2024-11-20 13:30:30,371+0100 WARN  [FelixStartLevel] *SYSTEM javax.xml.bind - Using non-standard property: javax.xml.bind.
```

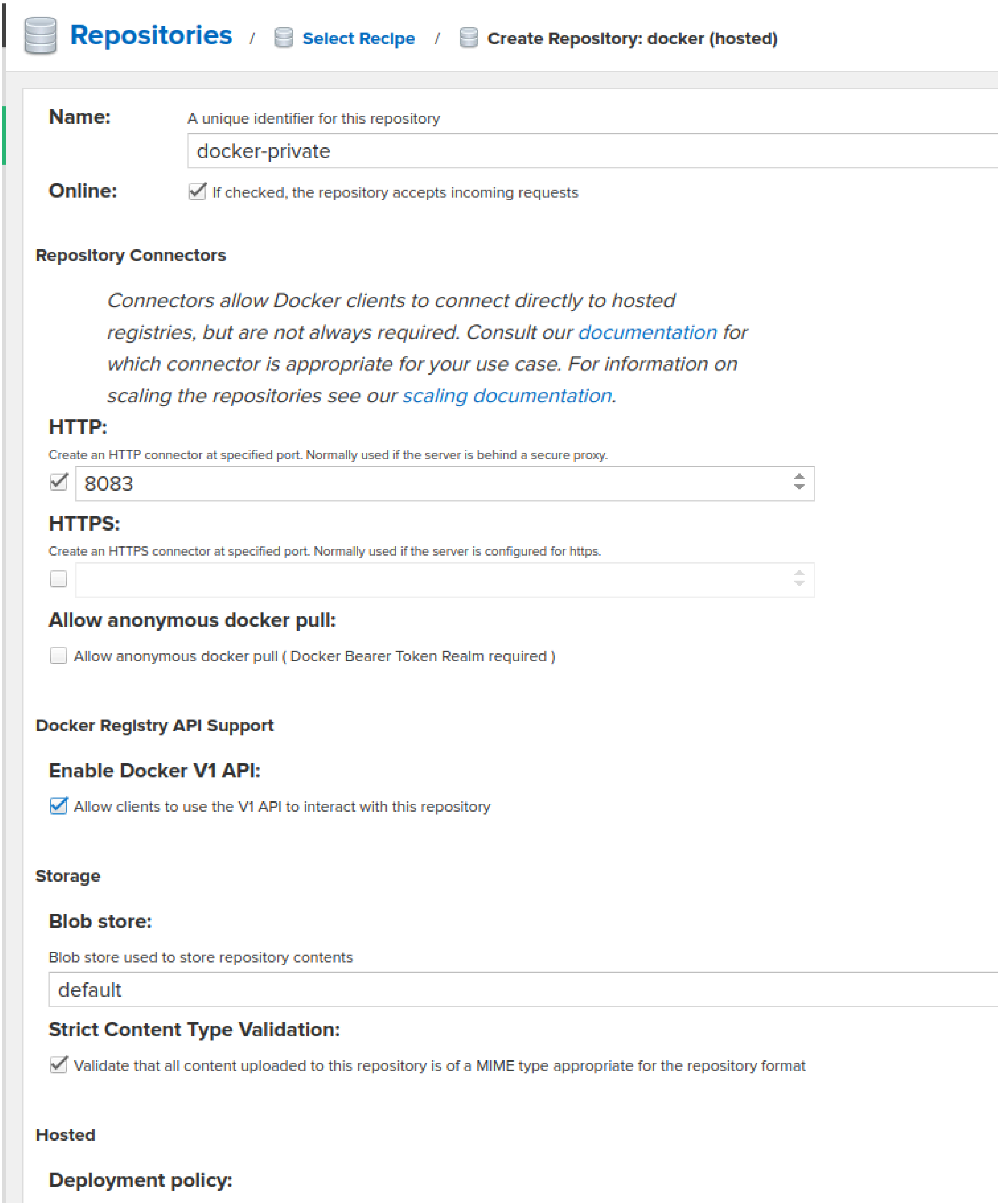


localhost:8081

User signed in: admin

Now we need to configure nexus to host docker images. For that we will need to setup private repository for our own images, add proxy for docker hub images and create group to to provide access to both, on single URL.

Private repository:



Repositories / Select Recipe / Create Repository: docker (hosted)

Name: A unique identifier for this repository
docker-private

Online: ☑ If checked, the repository accepts incoming requests

Repository Connectors

Connectors allow Docker clients to connect directly to hosted registries, but are not always required. Consult our documentation for which connector is appropriate for your use case. For information on scaling the repositories see our scaling documentation.

HTTP:
Create an HTTP connector at specified port. Normally used if the server is behind a secure proxy.
☑ 8083

HTTPS:
Create an HTTPS connector at specified port. Normally used if the server is configured for https.
☐

Allow anonymous docker pull:
☐ Allow anonymous docker pull (Docker Bearer Token Realm required)

Docker Registry API Support

Enable Docker V1 API:
☑ Allow clients to use the V1 API to interact with this repository

Storage

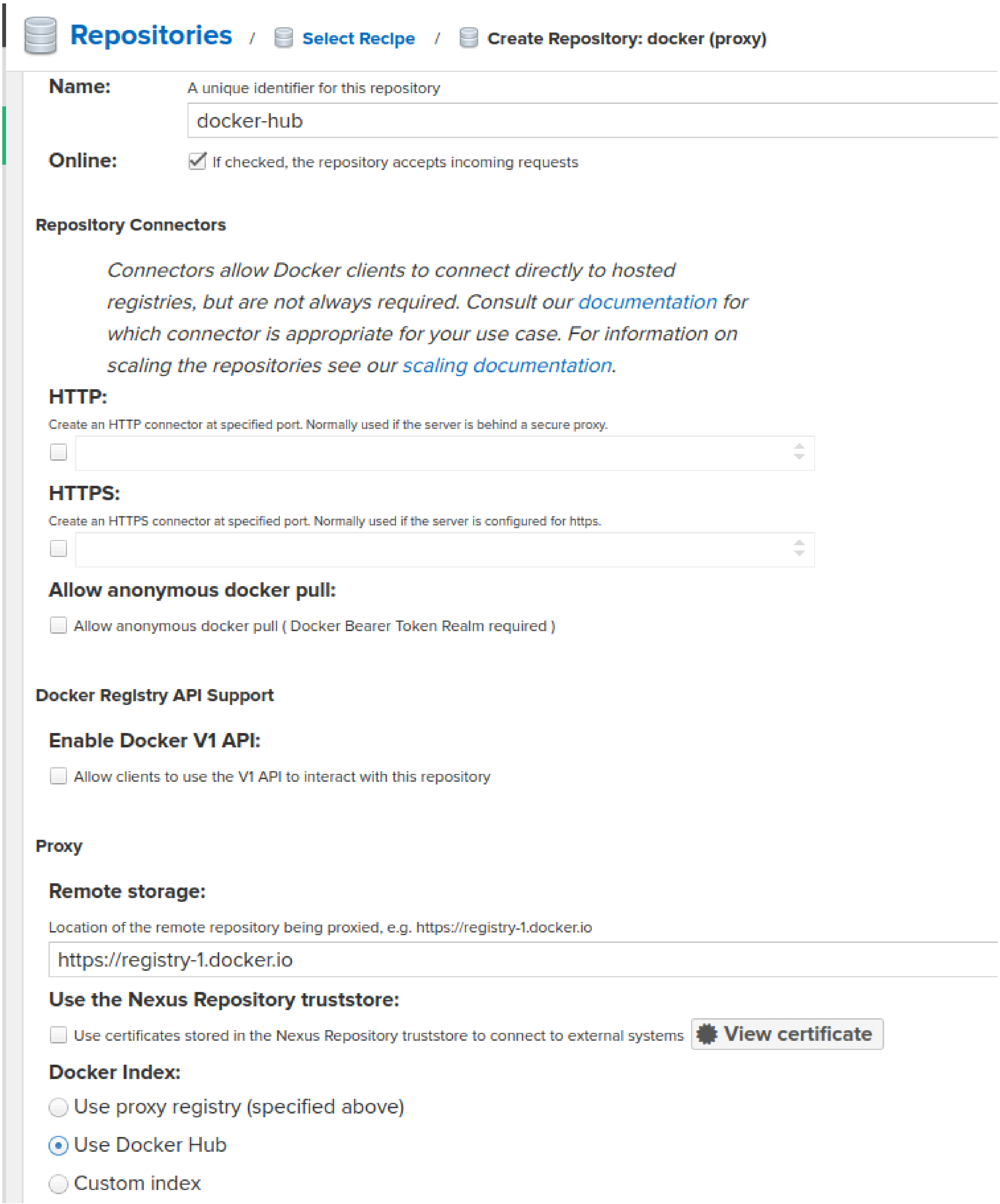
Blob store:
Blob store used to store repository contents
default

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:

Proxy repository:



Repositories / Select Recipe / Create Repository: docker (proxy)

Name: A unique identifier for this repository

docker-hub

Online: ☑ If checked, the repository accepts incoming requests

Repository Connectors

Connectors allow Docker clients to connect directly to hosted registries, but are not always required. Consult our documentation for which connector is appropriate for your use case. For information on scaling the repositories see our scaling documentation.

HTTP:

Create an HTTP connector at specified port. Normally used if the server is behind a secure proxy.

□

HTTPS:

Create an HTTPS connector at specified port. Normally used if the server is configured for https.

□

Allow anonymous docker pull:

□ Allow anonymous docker pull (Docker Bearer Token Realm required)

Docker Registry API Support

Enable Docker V1 API:

□ Allow clients to use the V1 API to interact with this repository

Proxy

Remote storage:

Location of the remote repository being proxied, e.g. https://registry-1.docker.io

https://registry-1.docker.io

Use the Nexus Repository truststore:

□ Use certificates stored in the Nexus Repository truststore to connect to external systems

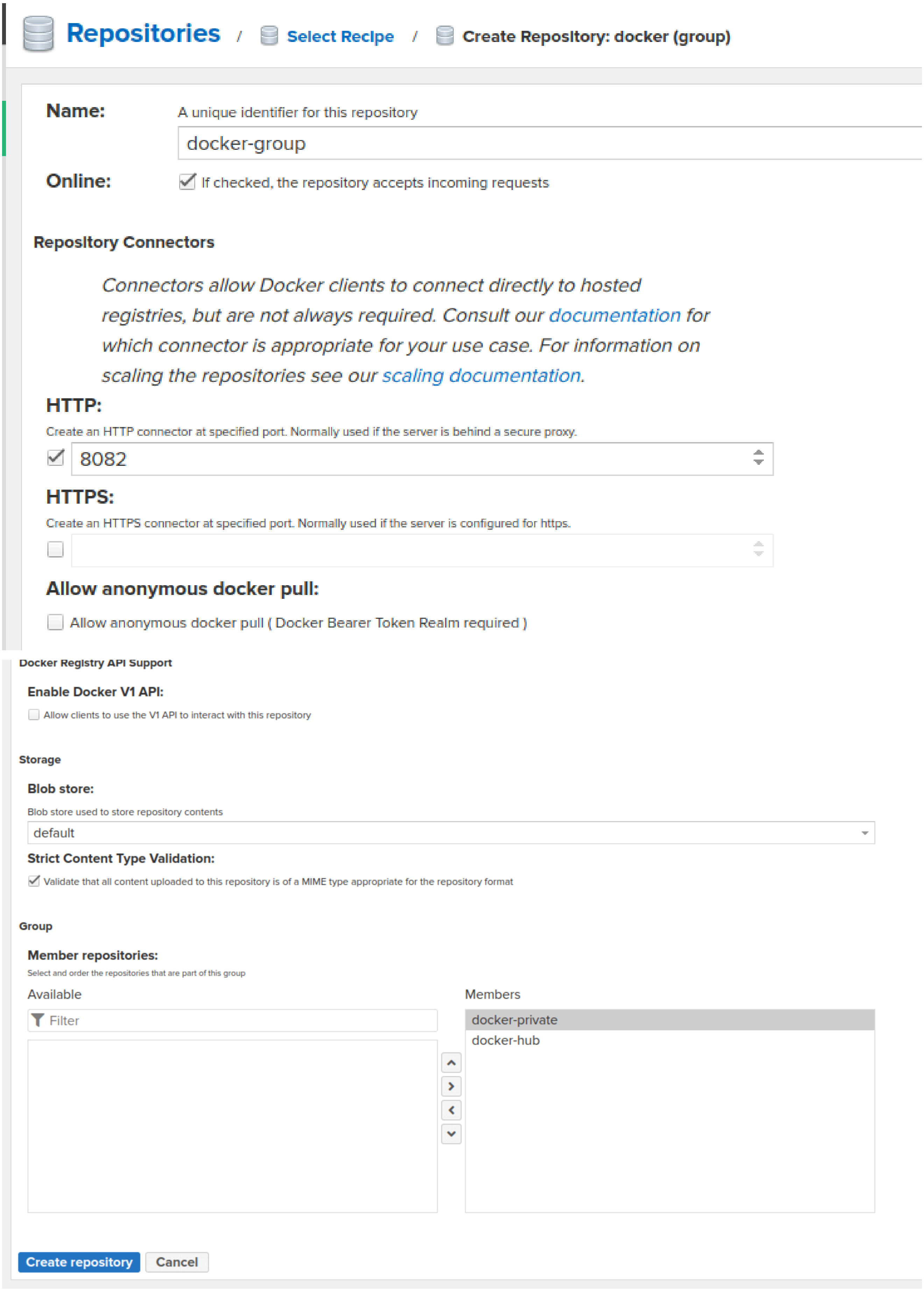
Docker Index:

○ Use proxy registry (specified above)

● Use Docker Hub

○ Custom index

Group repository:



Repositories / Select Recipe / Create Repository: docker (group)

Name: A unique identifier for this repository

docker-group

Online: ☑ If checked, the repository accepts incoming requests

Repository Connectors

Connectors allow Docker clients to connect directly to hosted registries, but are not always required. Consult our documentation for which connector is appropriate for your use case. For information on scaling the repositories see our scaling documentation.

HTTP:

Create an HTTP connector at specified port. Normally used if the server is behind a secure proxy.

☑ 8082

HTTPS:

Create an HTTPS connector at specified port. Normally used if the server is configured for https.

□

Allow anonymous docker pull:

□ Allow anonymous docker pull (Docker Bearer Token Realm required)

Docker Registry API Support

Enable Docker V1 API:

□ Allow clients to use the V1 API to interact with this repository

Storage

Blob store:

Blob store used to store repository contents

default

Strict Content Type Validation:

☑ Validate that all content uploaded to this repository is of a MIME type appropriate for the repository format

Group

Member repositories:

Select and order the repositories that are part of this group

Available

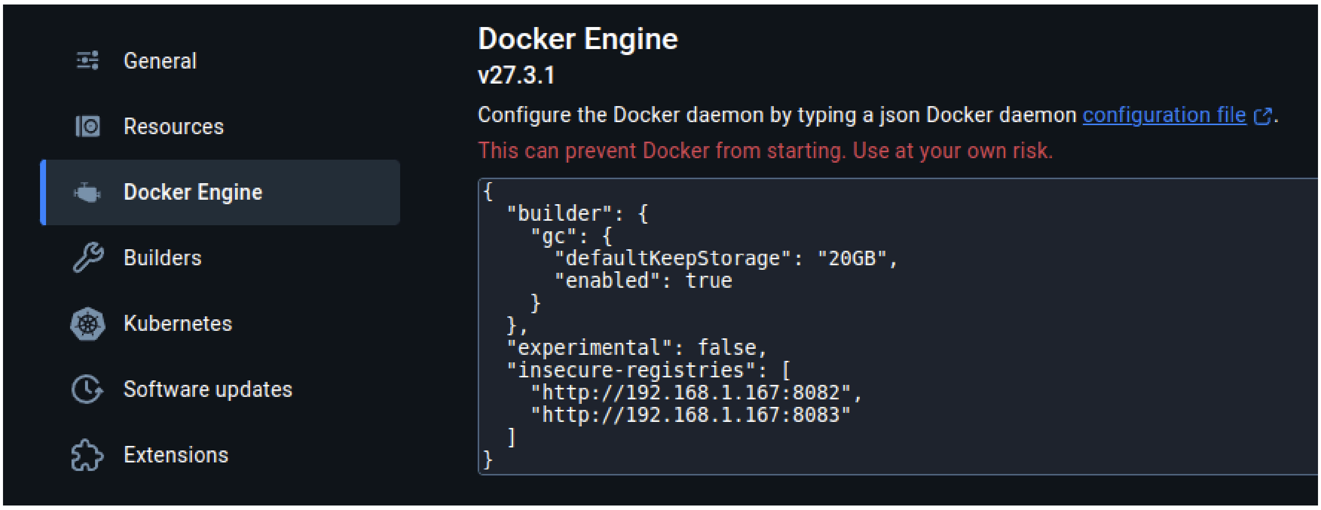
Filter

Members

docker-private

docker-hub

Now we need to authorize docker to access these repositories. However because we use them in HTTP, we first need to whitelist them since docker by default doesn't allow HTTP connections. To do that we add `insecure-registries` to docker daemon config.



Docker Engine

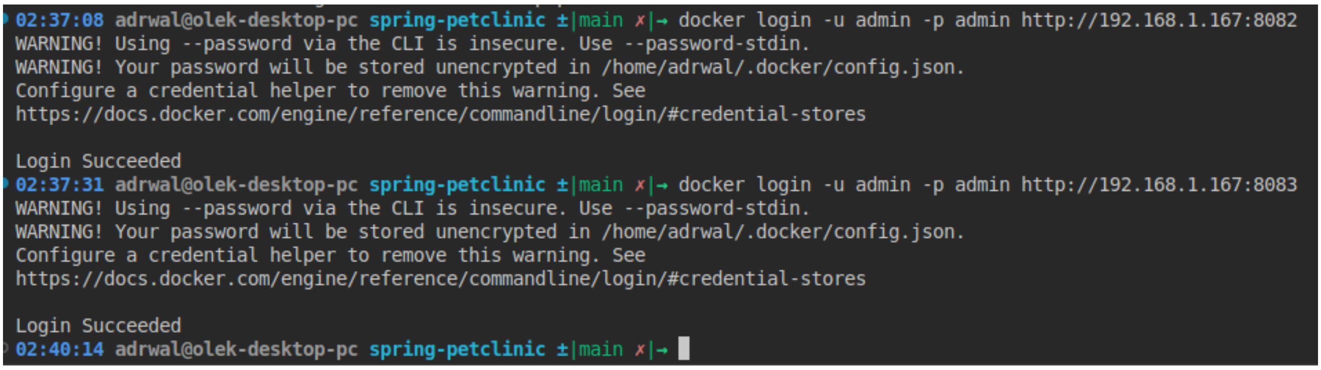
v27.3.1

Configure the Docker daemon by typing a json Docker daemon configuration file.

This can prevent Docker from starting. Use at your own risk.

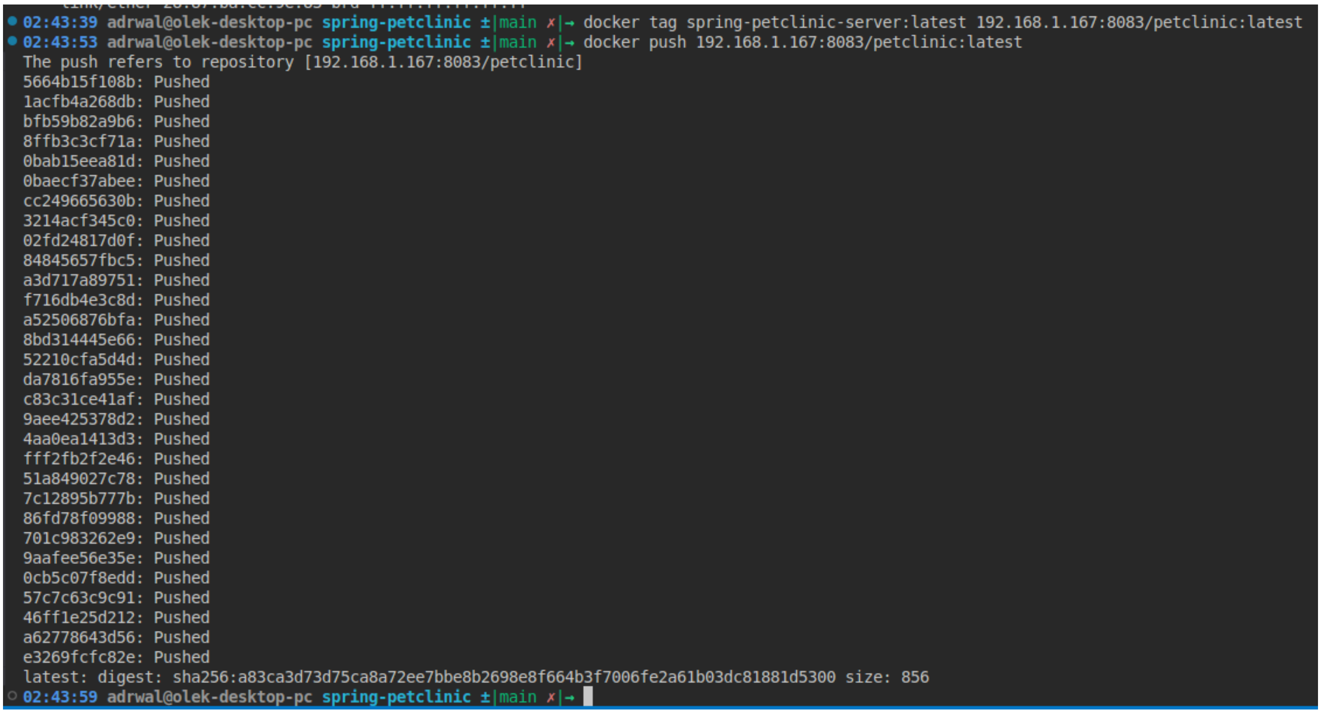
```
{
  "builder": {
    "gc": {
      "defaultKeepStorage": "20GB",
      "enabled": true
    }
  },
  "experimental": false,
  "insecure-registries": [
    "http://192.168.1.167:8082",
    "http://192.168.1.167:8083"
  ]
}
```

Now we can login to those repositories with docker.

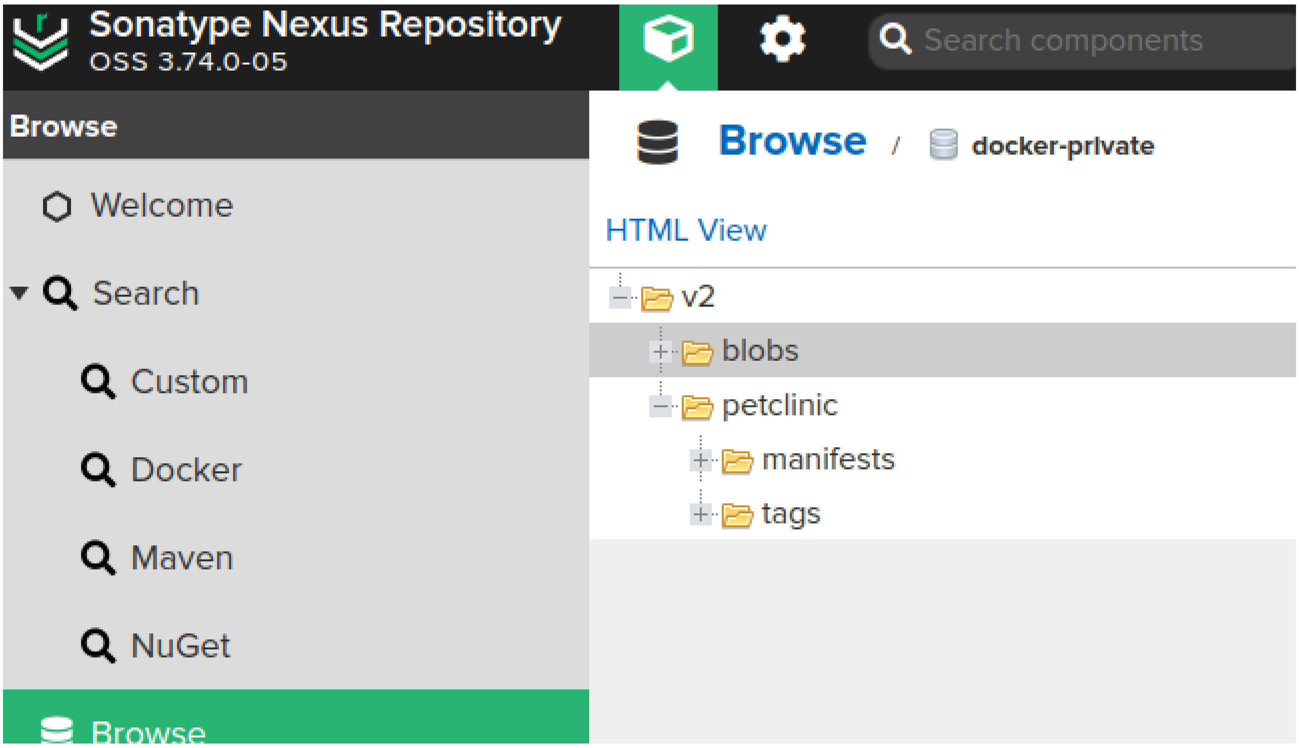


```
02:37:08 adrwal@olek-desktop-pc spring-petclinic ±|main ✗|→ docker login -u admin -p admin http://192.168.1.167:8082
WARNING! Using --password via the CLI is insecure. Use --password-stdin.
WARNING! Your password will be stored unencrypted in /home/adrwal/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credential-stores

Login Succeeded
02:37:31 adrwal@olek-desktop-pc spring-petclinic ±|main ✗|→ docker login -u admin -p admin http://192.168.1.167:8083
WARNING! Using --password via the CLI is insecure. Use --password-stdin.
WARNING! Your password will be stored unencrypted in /home/adrwal/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credential-stores

Login Succeeded
02:40:14 adrwal@olek-desktop-pc spring-petclinic ±|main ✗|→
```

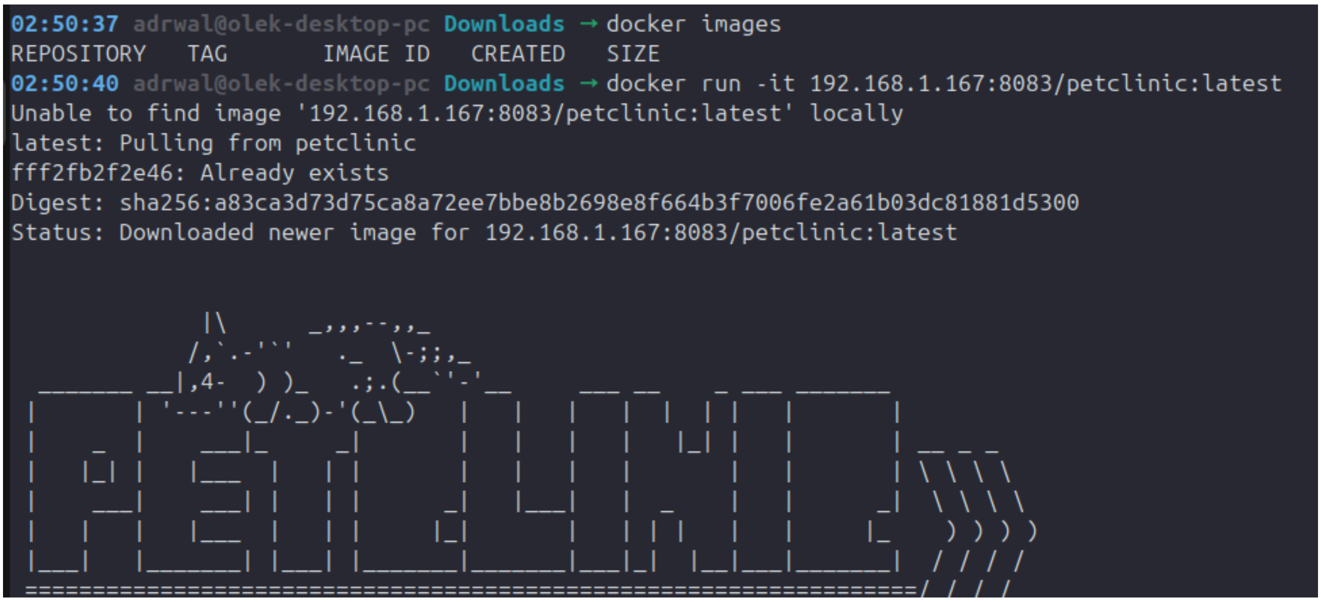
We can now push image created by earlier docker compose to nexus. Docker needs to know which repository to use to pull/push. It does it by checking the image tag, so we need to correctly tag our image.



```
● 02:43:39 adrwal@olek-desktop-pc spring-petclinic ±|main ✗|→ docker tag spring-petclinic-server:latest 192.168.1.167:8083/petclinic:latest
● 02:43:53 adrwal@olek-desktop-pc spring-petclinic ±|main ✗|→ docker push 192.168.1.167:8083/petclinic:latest
  The push refers to repository [192.168.1.167:8083/petclinic]
  5664b15f108b: Pushed
  1acfb4a268db: Pushed
  bfb59b82a9b6: Pushed
  8ffb3c3cf71a: Pushed
  0bab15eea81d: Pushed
  0baecf37abee: Pushed
  cc249665630b: Pushed
  3214acf345c0: Pushed
  02fd24817d0f: Pushed
  84845657fbc5: Pushed
  a3d717a89751: Pushed
  f716db4e3c8d: Pushed
  a52506876bfa: Pushed
  8bd314445e66: Pushed
  52210cfa5d4d: Pushed
  da7816fa955e: Pushed
  c83c31ce41af: Pushed
  9aee425378d2: Pushed
  4aa0ea1413d3: Pushed
  fff2fb2f2e46: Pushed
  51a849027c78: Pushed
  7c12895b777b: Pushed
  86fd78f09988: Pushed
  701c983262e9: Pushed
  9aafee56e35e: Pushed
  0cb5c07f8edd: Pushed
  57c7c63c9c91: Pushed
  46ff1e25d212: Pushed
  a62778643d56: Pushed
  e3269fcfc82e: Pushed
  latest: digest: sha256:a83ca3d73d75ca8a72ee7bbe8b2698e8f664b3f7006fe2a61b03dc81881d5300 size: 856
○ 02:43:59 adrwal@olek-desktop-pc spring-petclinic ±|main ✗|→
```



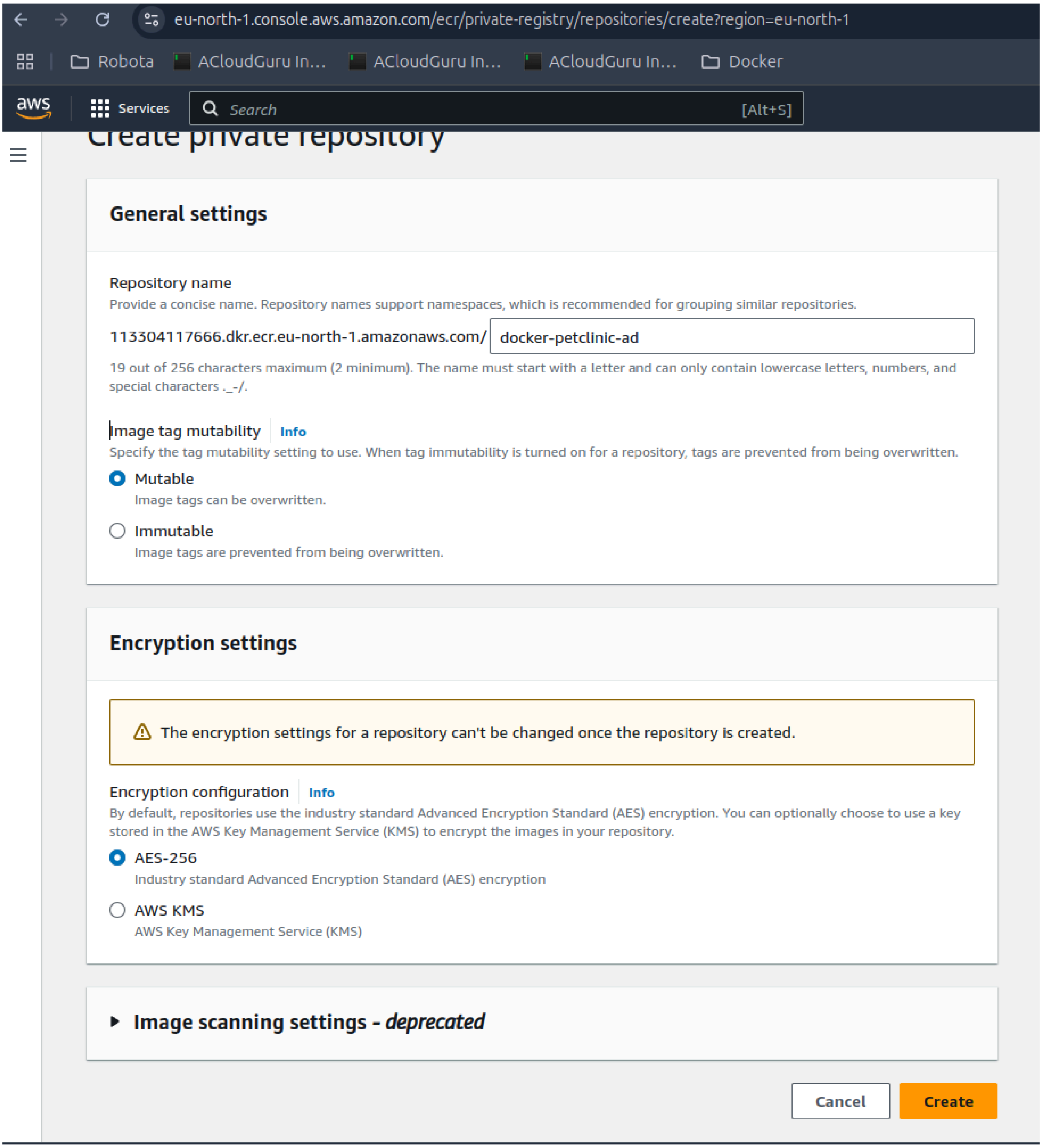
We can verify that this image can also be pulled from private repository by first removing all images and then trying to run petclinic image uploaded to our private docker repository.



```
02:50:37 adrwal@olek-desktop-pc Downloads → docker images
REPOSITORY   TAG       IMAGE ID   CREATED   SIZE
02:50:40 adrwal@olek-desktop-pc Downloads → docker run -it 192.168.1.167:8083/petclinic:latest
Unable to find image '192.168.1.167:8083/petclinic:latest' locally
latest: Pulling from petclinic
fff2fb2f2e46: Already exists
Digest: sha256:a83ca3d73d75ca8a72ee7bbe8b2698e8f664b3f7006fe2a61b03dc81881d5300
Status: Downloaded newer image for 192.168.1.167:8083/petclinic:latest


              |\      _,,,--,,_
             /,`.-'`'   ._  \-;;,_
  _______ __|,4-  ) )_   .;.(__`'-'__     ___ __    _ ___ _______
 |       | '---''(_/._)-'(_\_)   |   |   |   |  |  | |   |       |
 |    _  |    ___|_     _|       |   |   |   |   |_| |   |       | __ _ _
 |   |_| |   |___  |   | |       |   |   |   |       |   |       | \ \ \ \
 |    ___|    ___| |   | |      _|   |___|   |  _    |   |      _|  \ \ \ \
 |   |   |   |___  |   | |     |_|       |   | | |   |   |     |_    ) ) ) )
 |___|   |_______| |___| |_______|_______|___|_|  |__|___|_______|  / / / /
 ==================================================================/ / / /
```

b. ECR

First we create new private AWS ECR repository.


eu-north-1.console.aws.amazon.com/ecr/private-registry/repositories/create?region=eu-north-1
Robota
ACloudGuru In...
ACloudGuru In...
ACloudGuru In...
Docker
aws
Services
Search
[Alt+S]

Create private repository

General settings

Repository name
Provide a concise name. Repository names support namespaces, which is recommended for grouping similar repositories.
113304117666.dkr.ecr.eu-north-1.amazonaws.com/ docker-petclinic-ad
19 out of 256 characters maximum (2 minimum). The name must start with a letter and can only contain lowercase letters, numbers, and special characters ._-/.

Image tag mutability Info
Specify the tag mutability setting to use. When tag immutability is turned on for a repository, tags are prevented from being overwritten.
● Mutable
Image tags can be overwritten.
○ Immutable
Image tags are prevented from being overwritten.

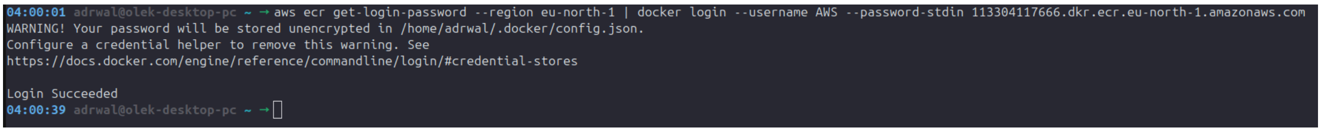
Encryption settings

⚠ The encryption settings for a repository can't be changed once the repository is created.

Encryption configuration Info
By default, repositories use the industry standard Advanced Encryption Standard (AES) encryption. You can optionally choose to use a key stored in the AWS Key Management Service (KMS) to encrypt the images in your repository.
● AES-256
Industry standard Advanced Encryption Standard (AES) encryption
○ AWS KMS
AWS Key Management Service (KMS)

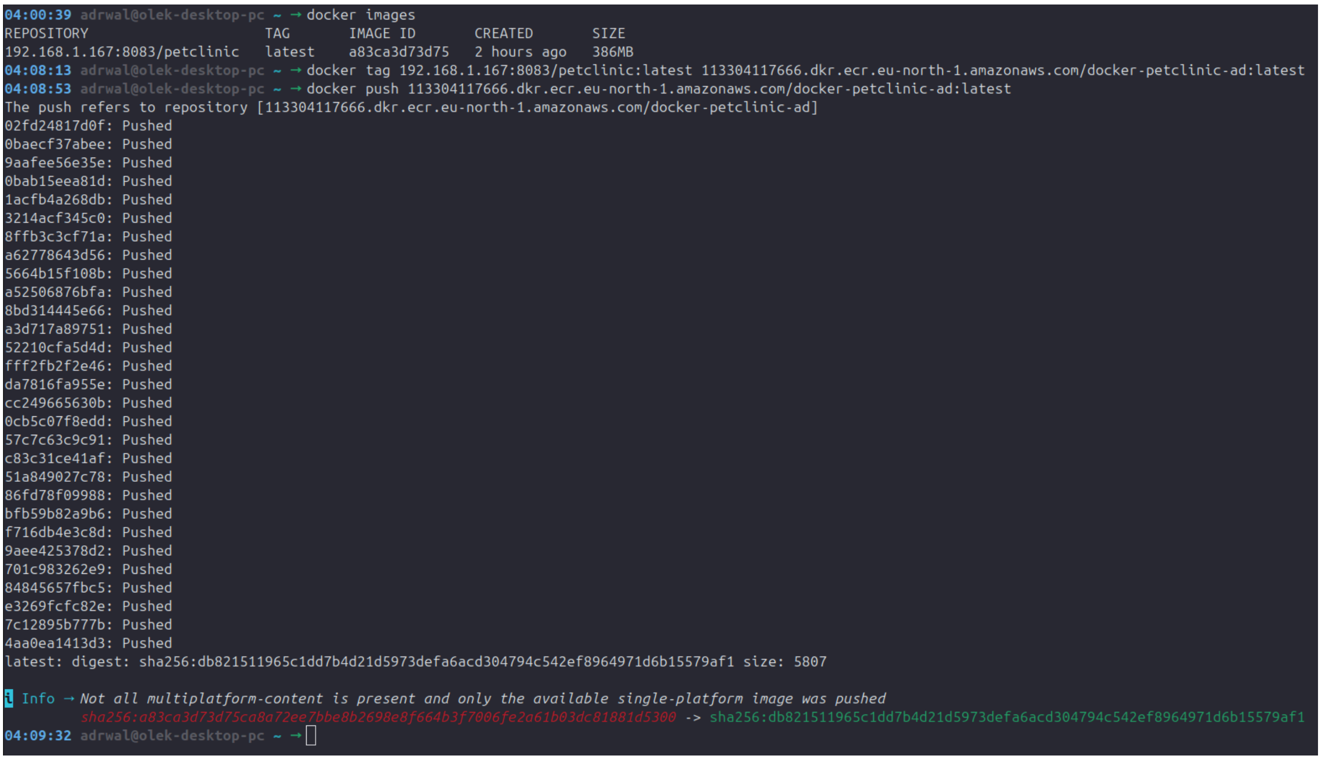
▸ Image scanning settings - *deprecated*

Now authenticate docker to use it:



```
04:00:01 adrwal@olek-desktop-pc ~ → aws ecr get-login-password --region eu-north-1 | docker login --username AWS --password-stdin 113304117666.dkr.ecr.eu-north-1.amazonaws.com
WARNING! Your password will be stored unencrypted in /home/adrwal/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credential-stores

Login Succeeded
04:00:39 adrwal@olek-desktop-pc ~ →
```

Now we need to correctly tag the image from earlier steps and push it to ECR.



```
04:00:39 adrwal@olek-desktop-pc ~ → docker images
REPOSITORY                    TAG       IMAGE ID       CREATED       SIZE
192.168.1.167:8083/petclinic  latest    a83ca3d73d75   2 hours ago   386MB
04:08:13 adrwal@olek-desktop-pc ~ → docker tag 192.168.1.167:8083/petclinic:latest 113304117666.dkr.ecr.eu-north-1.amazonaws.com/docker-petclinic-ad:latest
04:08:53 adrwal@olek-desktop-pc ~ → docker push 113304117666.dkr.ecr.eu-north-1.amazonaws.com/docker-petclinic-ad:latest
The push refers to repository [113304117666.dkr.ecr.eu-north-1.amazonaws.com/docker-petclinic-ad]
02fd24817d0f: Pushed
0baecf37abee: Pushed
9aafee56e35e: Pushed
0bab15eea81d: Pushed
1acfb4a268db: Pushed
3214acf345c0: Pushed
8ffb3c3cf71a: Pushed
a62778643d56: Pushed
5664b15f108b: Pushed
a52506876bfa: Pushed
8bd314445e66: Pushed
a3d717a89751: Pushed
52210cfa5d4d: Pushed
fff2fb2f2e46: Pushed
da7816fa955e: Pushed
cc249665630b: Pushed
0cb5c07f8edd: Pushed
57c7c63c9c91: Pushed
c83c31ce41af: Pushed
51a849027c78: Pushed
86fd78f09988: Pushed
bfb59b82a9b6: Pushed
f716db4e3c8d: Pushed
9aee425378d2: Pushed
701c983262e9: Pushed
84845657fbc5: Pushed
e3269fcfc82e: Pushed
7c12895b777b: Pushed
4aa0ea1413d3: Pushed
latest: digest: sha256:db821511965c1dd7b4d21d5973defa6acd304794c542ef8964971d6b15579af1 size: 5807

i Info → Not all multiplatform-content is present and only the available single-platform image was pushed
         sha256:a83ca3d73d75ca8a72ee7bbe8b2698e8f664b3f7006fe2a61b03dc81881d5300 -> sha256:db821511965c1dd7b4d21d5973defa6acd304794c542ef8964971d6b15579af1
04:09:32 adrwal@olek-desktop-pc ~ →
```

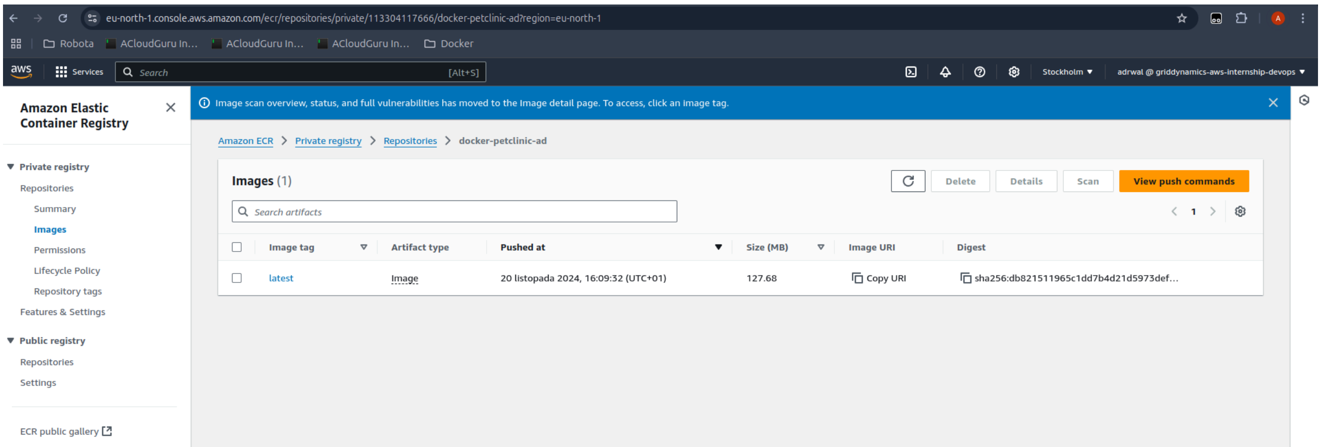


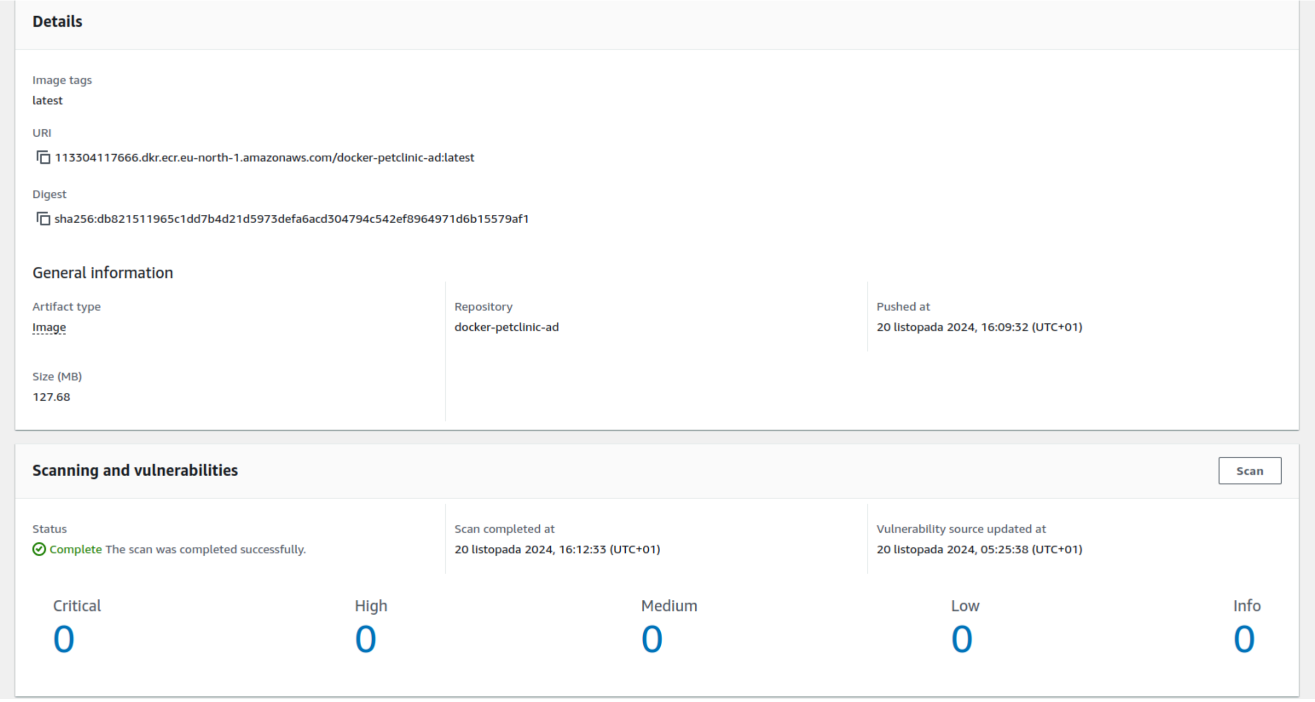
Image scan overview, status, and full vulnerabilities has moved to the Image detail page. To access, click an image tag.

Images (1)

□	Image tag	Artifact type	Pushed at	Size (MB)	Image URI	Digest
□	latest	Image	20 listopada 2024, 16:09:32 (UTC+01)	127.68	Copy URI	sha256:db821511965c1dd7b4d21d5973def...

1. Perform security scan for uploaded images in ECR

Scanning the image shows 0 vulnerabilities in every category



Details

Image tags
latest

URI
113304117666.dkr.ecr.eu-north-1.amazonaws.com/docker-petclinic-ad:latest

Digest
sha256:db821511965c1dd7b4d21d5973defa6acd304794c542ef8964971d6b15579af1

General information

Artifact type	Repository	Pushed at
Image	docker-petclinic-ad	20 listopada 2024, 16:09:32 (UTC+01)

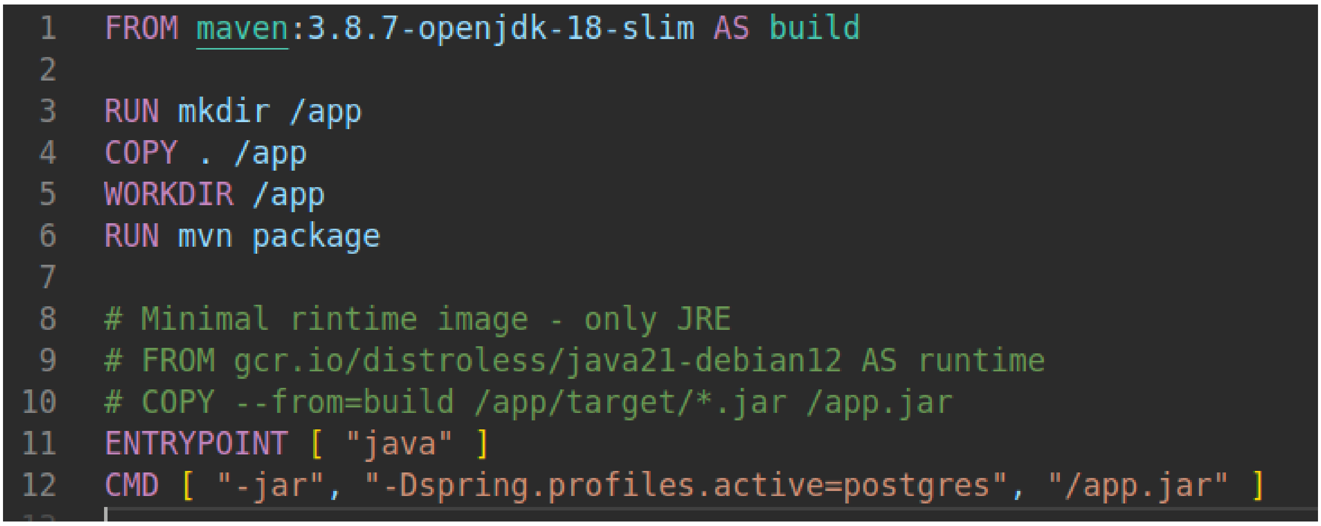
Size (MB)
127.68

Scanning and vulnerabilities

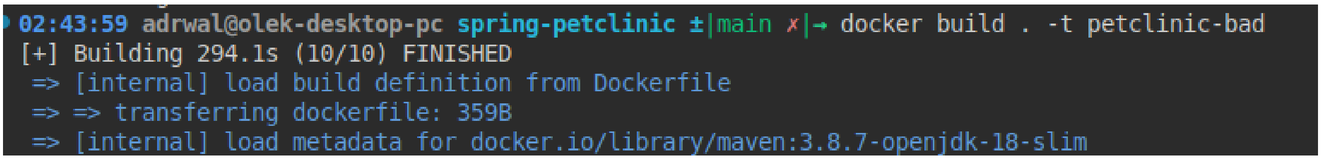
Status	Scan completed at	Vulnerability source updated at
Complete The scan was completed successfully.	20 listopada 2024, 16:12:33 (UTC+01)	20 listopada 2024, 05:25:38 (UTC+01)

Critical	High	Medium	Low	Info
0	0	0	0	0

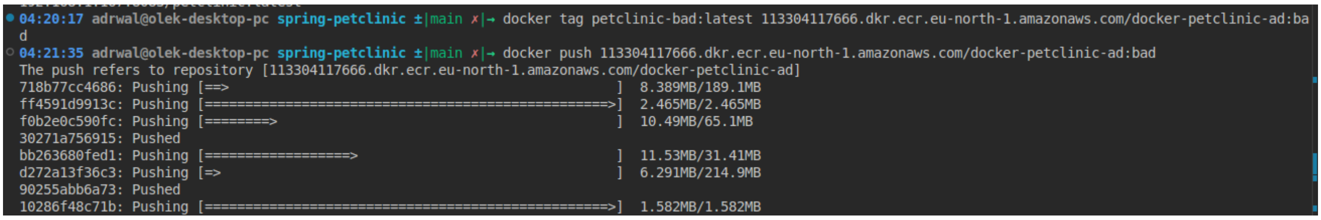
This is probably because we are using minimal container with only JRE. We can verify if that is true by commenting out lines responsible for second build stage in our Dockerfile and uploading this new image.



```
FROM maven:3.8.7-openjdk-18-slim AS build

RUN mkdir /app
COPY . /app
WORKDIR /app
RUN mvn package

# Minimal rintime image - only JRE
# FROM gcr.io/distroless/java21-debian12 AS runtime
# COPY --from=build /app/target/*.jar /app.jar
ENTRYPOINT [ "java" ]
CMD [ "-jar", "-Dspring.profiles.active=postgres", "/app.jar" ]
```



```
02:43:59 adrwal@olek-desktop-pc spring-petclinic ±|main ✗|→ docker build . -t petclinic-bad
[+] Building 294.1s (10/10) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 359B
 => [internal] load metadata for docker.io/library/maven:3.8.7-openjdk-18-slim
```



```
04:20:17 adrwal@olek-desktop-pc spring-petclinic ±|main ✗|→ docker tag petclinic-bad:latest 113304117666.dkr.ecr.eu-north-1.amazonaws.com/docker-petclinic-ad:ba
d
04:21:35 adrwal@olek-desktop-pc spring-petclinic ±|main ✗|→ docker push 113304117666.dkr.ecr.eu-north-1.amazonaws.com/docker-petclinic-ad:bad
The push refers to repository [113304117666.dkr.ecr.eu-north-1.amazonaws.com/docker-petclinic-ad]
718b77cc4686: Pushing [==>                                                ]  8.389MB/189.1MB
ff4591d9913c: Pushing [==================================================>]  2.465MB/2.465MB
f0b2e0c590fc: Pushing [========>                                          ]  10.49MB/65.1MB
30271a756915: Pushed
bb263680fed1: Pushing [==================>                                ]  11.53MB/31.41MB
d272a13f36c3: Pushing [=>                                                 ]  6.291MB/214.9MB
90255abb6a73: Pushed
10286f48c71b: Pushing [==================================================>]  1.582MB/1.582MB
```

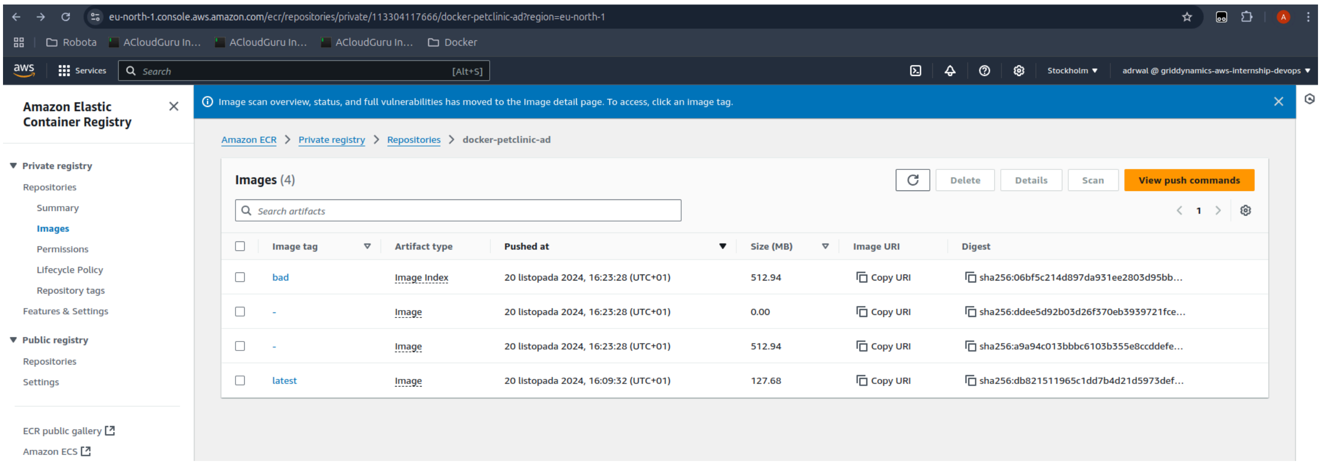
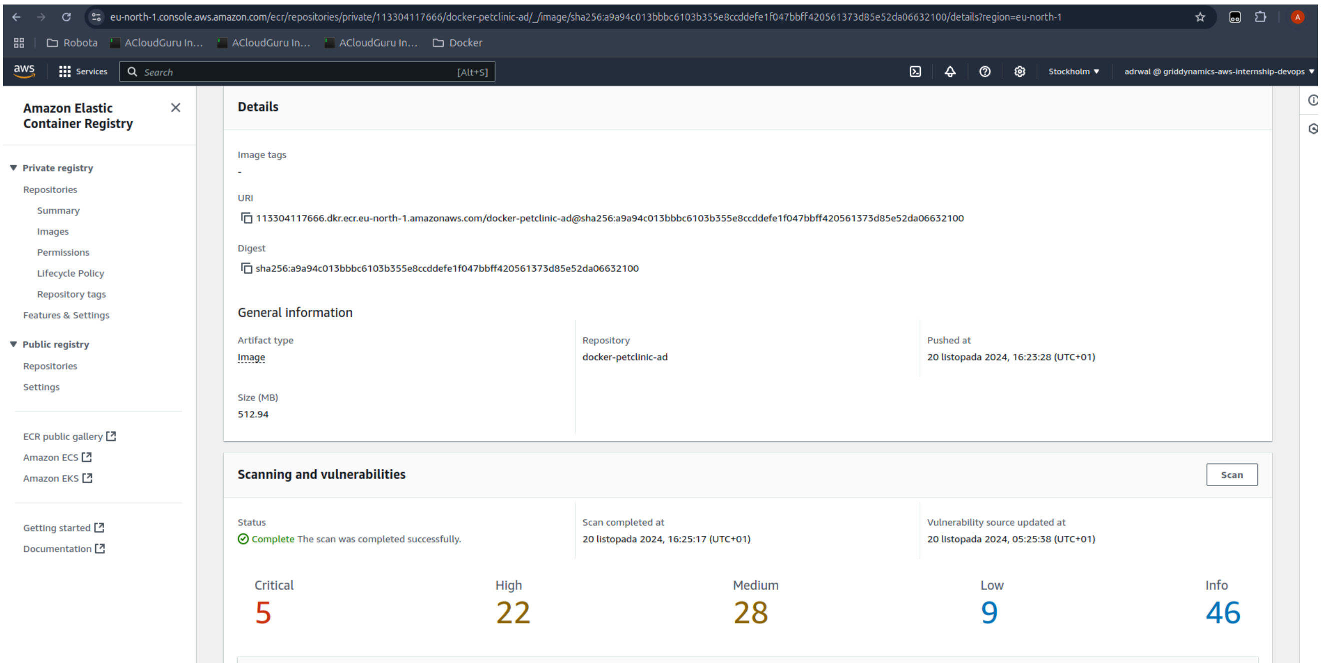


Image scan overview, status, and full vulnerabilities has moved to the image detail page. To access, click an image tag.

Images (4)

□	Image tag	Artifact type	Pushed at	Size (MB)	Image URI	Digest
□	bad	Image Index	20 listopada 2024, 16:23:28 (UTC+01)	512.94	Copy URI	sha256:06bf5c214d897da931ee2803d95bb...
□	-	Image	20 listopada 2024, 16:23:28 (UTC+01)	0.00	Copy URI	sha256:ddee5d92b03d26f370eb3939721fce...
□	-	Image	20 listopada 2024, 16:23:28 (UTC+01)	512.94	Copy URI	sha256:a9a94c013bbbc6103b355e8ccddefe...
□	latest	Image	20 listopada 2024, 16:09:32 (UTC+01)	127.68	Copy URI	sha256:db821511965c1dd7b4d21d5973def...



Details

Image tags
-

URI
113304117666.dkr.ecr.eu-north-1.amazonaws.com/docker-petclinic-ad@sha256:a9a94c013bbbc6103b355e8ccddefe1f047bbff420561373d85e52da06632100

Digest
sha256:a9a94c013bbbc6103b355e8ccddefe1f047bbff420561373d85e52da06632100

General information

Artifact type: Image
Repository: docker-petclinic-ad
Pushed at: 20 listopada 2024, 16:23:28 (UTC+01)
Size (MB): 512.94

Scanning and vulnerabilities

Status: Complete The scan was completed successfully.
Scan completed at: 20 listopada 2024, 16:25:17 (UTC+01)
Vulnerability source updated at: 20 listopada 2024, 05:25:38 (UTC+01)

Critical	High	Medium	Low	Info
5	22	28	9	46

As we expected image based on maven contains multiple vulnerabilities and is bigger in size. That is because it contains many different applications which are not needed during runtime and every application brings possible security risk. On the contrary our initial image which used `gcr.io/distroless/java21-debian12` doesn't even have shell. With all those points of failure removed our container is way more secure.