Lab 1 (15 min)

In this lab you will set up and configure your personal JFrog Platform environment.
Your personal Environment will be used for the other labs in the workshop.

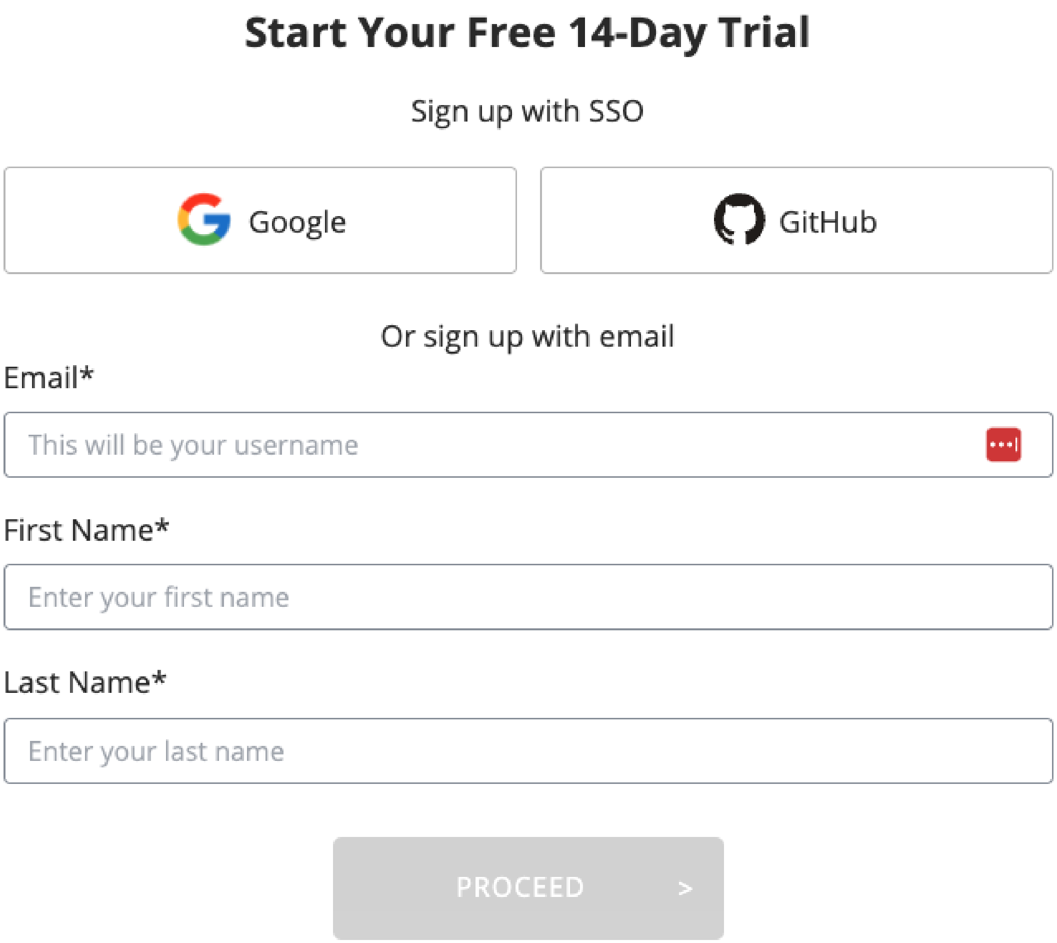
Your environment will be available for 2 weeks!

Upon successful completion of this lab you will be able to login to your personal environment with your personal credentials and observe two docker repositories configured for you. You will also be able to browse demo data and findings in the platform

Step by step instructions

1. Open your terminal & download the zip file
   ```
   curl -sLO
   https://releases.jfrog.io/artifactory/website/security/guided-trial.zip
   ```
2. Unzip it to your selected working folder
3. Browse to jfrog.com/start-free/security
4. Complete the registration process:

Phase #1:



Start Your Free 14-Day Trial

Sign up with SSO

Or sign up with email

Email*

This will be your username

First Name*

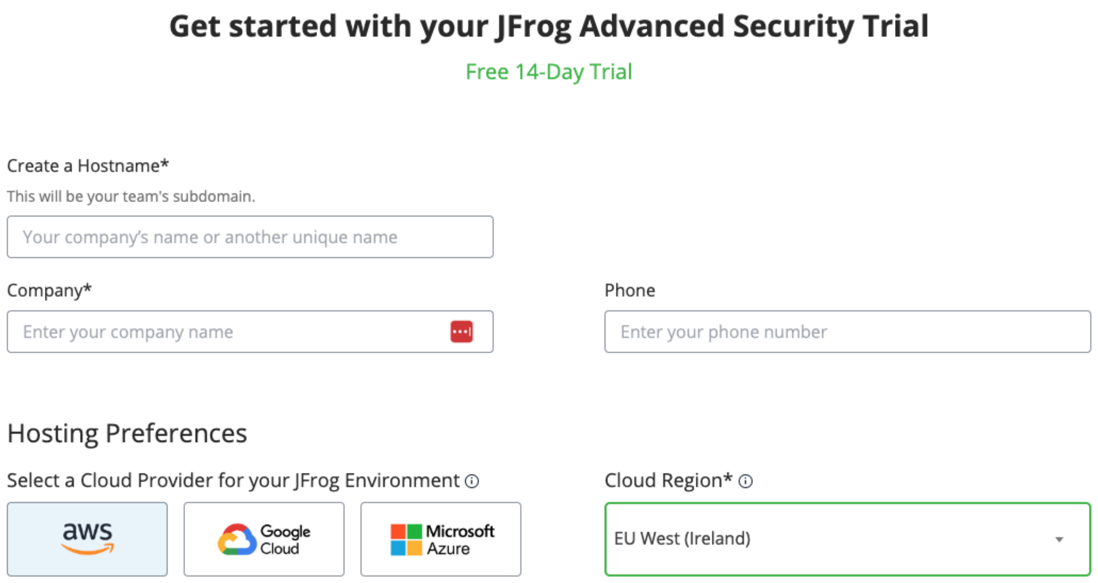
Enter your first name

Last Name*

Enter your last name

Phase #2

- Populate hostname which you should use later on. Best practice is `“firstname-lastname”` i.e. `”david-cohen”`
- Your company name
- Select AWS & “EU West” region (Ireland)



Get started with your JFrog Advanced Security Trial

Free 14-Day Trial

Create a Hostname*

This will be your team's subdomain.

Your company's name or another unique name

Company*

Enter your company name

Phone

Enter your phone number

Hosting Preferences

Select a Cloud Provider for your JFrog Environment ⓘ

Cloud Region* ⓘ

EU West (Ireland)

__Phase #3:__

Your environment is being prepared.

In the next screen, please select the password to be used (or API token in case of SSO)

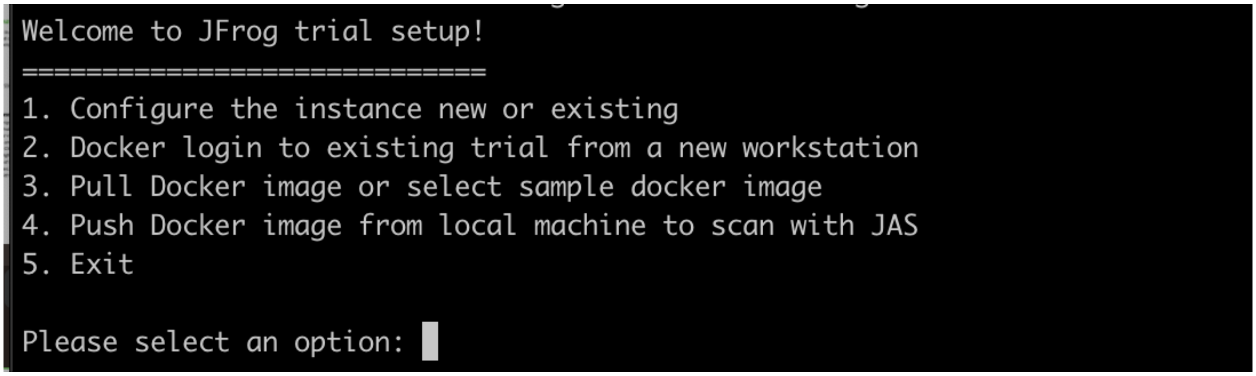
5. Return to your terminal and run

   `bash guided-trial/linux_guided_trial.sh`

   There is also a windows version in the folder if needed.

6. From the menu, select option #1:

   `Configure the instance new or existing`

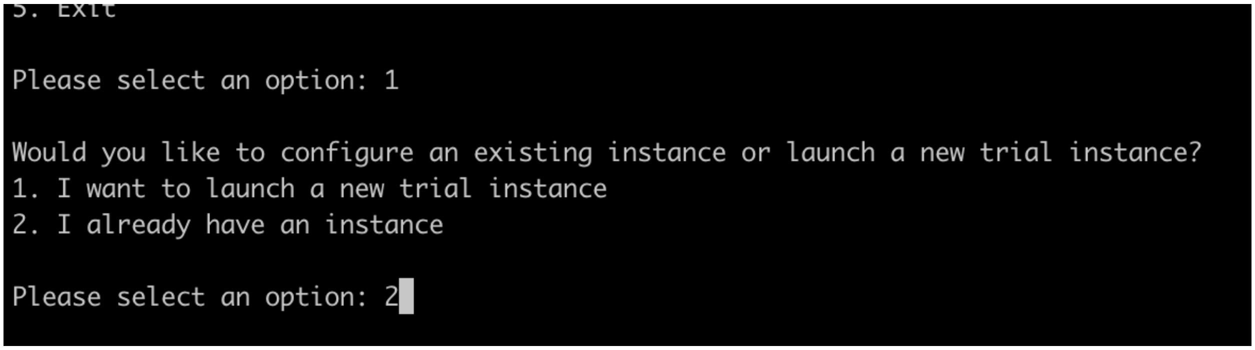


```
Welcome to JFrog trial setup!
================================
1. Configure the instance new or existing
2. Docker login to existing trial from a new workstation
3. Pull Docker image or select sample docker image
4. Push Docker image from local machine to scan with JAS
5. Exit

Please select an option: █
```

And then option #2

`I already have an instance`



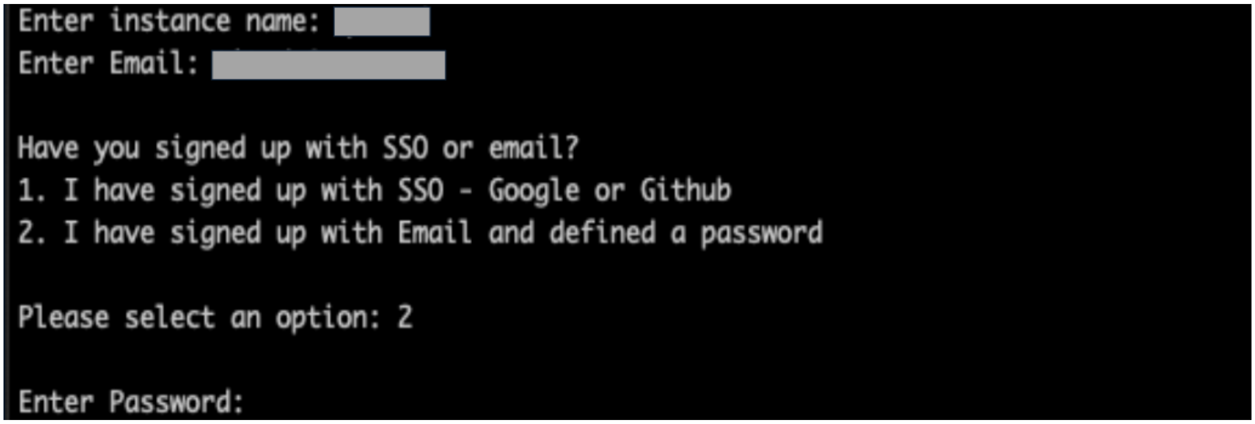
```
5. Exit

Please select an option: 1

Would you like to configure an existing instance or launch a new trial instance?
1. I want to launch a new trial instance
2. I already have an instance

Please select an option: 2
```

Now, enter your instance name, email address used & password or token as needed



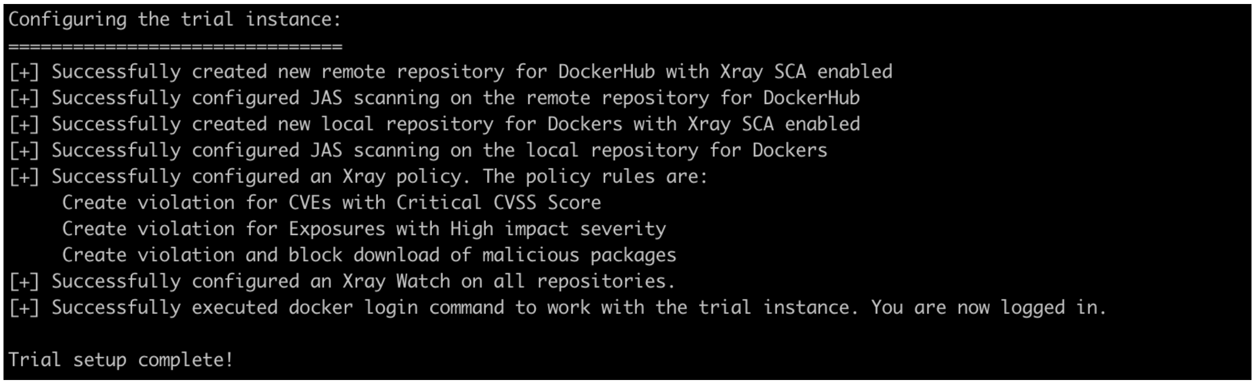
```
Enter instance name: [redacted]
Enter Email: [redacted]

Have you signed up with SSO or email?
1. I have signed up with SSO - Google or Github
2. I have signed up with Email and defined a password

Please select an option: 2

Enter Password:
```

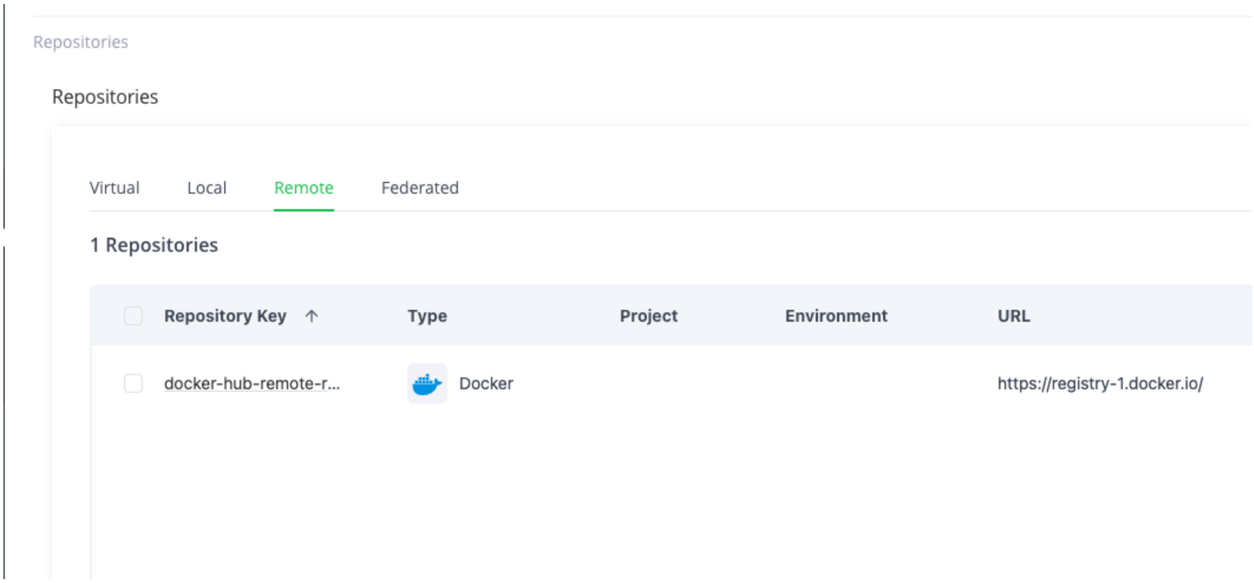
Note the script's outputs as it configures your environment:



```
Configuring the trial instance:
===============================
[+] Successfully created new remote repository for DockerHub with Xray SCA enabled
[+] Successfully configured JAS scanning on the remote repository for DockerHub
[+] Successfully created new local repository for Dockers with Xray SCA enabled
[+] Successfully configured JAS scanning on the local repository for Dockers
[+] Successfully configured an Xray policy. The policy rules are:
     Create violation for CVEs with Critical CVSS Score
     Create violation for Exposures with High impact severity
     Create violation and block download of malicious packages
[+] Successfully configured an Xray Watch on all repositories.
[+] Successfully executed docker login command to work with the trial instance. You are now logged in.

Trial setup complete!
```

7. Return to your browser and open your server at https://`<your instance name>`.jfrog.io/ui/admin/repositories/remote and see there is a `"docker-hub-remote-repo"` remote repository created:

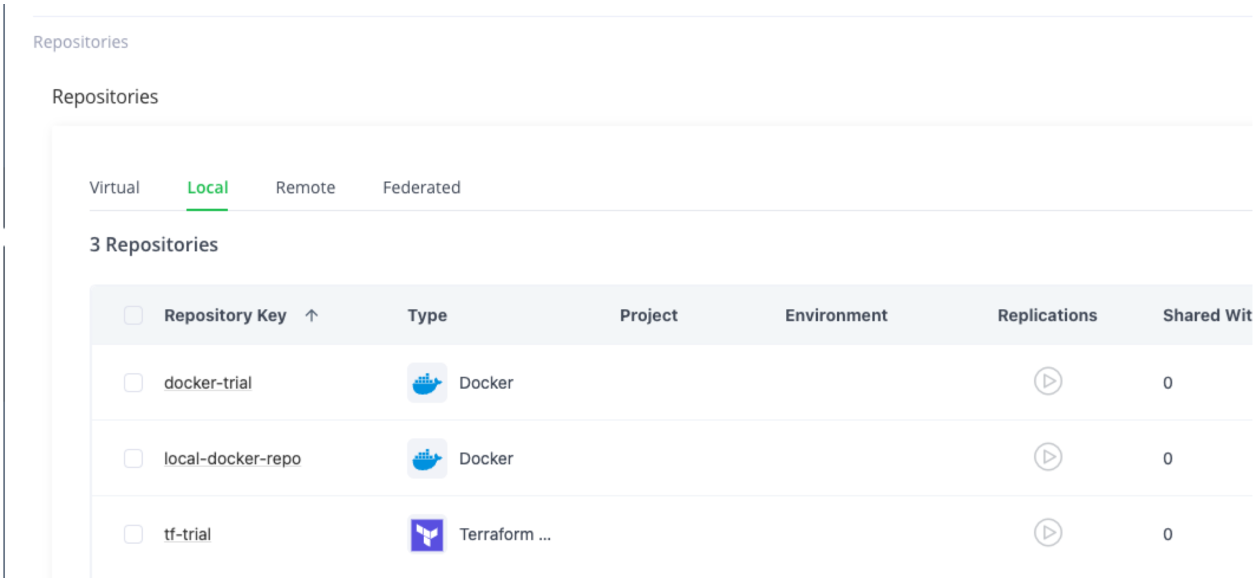


1 Repositories

Repository Key ↑	Type	Project	Environment	URL
□ docker-hub-remote-r...	Docker			https://registry-1.docker.io/

Your remote repository will be used in the next lab to pull a docker image from Docker-Hub.

8. Now, switch to the ‘local’ tab using the UI or https://`<your instance name>`.jfrog.io/ui/admin/repositories/local to see the `“local-docker-repo”` local repository created:



3 Repositories

Repository Key ↑	Type	Project	Environment	Replications	Shared Wit
□ docker-trial	Docker			▷	0
□ local-docker-repo	Docker			▷	0
□ tf-trial	Terraform ...			▷	0

Your local repository will be used in the next lab to push a docker image to the JFrog Platform.

Note the other two local repositories: `docker-trial` and `tf-trial`. Those are already pre-populated with Docker and Terraform data & can be browsed during or after the workshop.

Congratulations! You have completed Lab 1

Lab 2 (30 min)

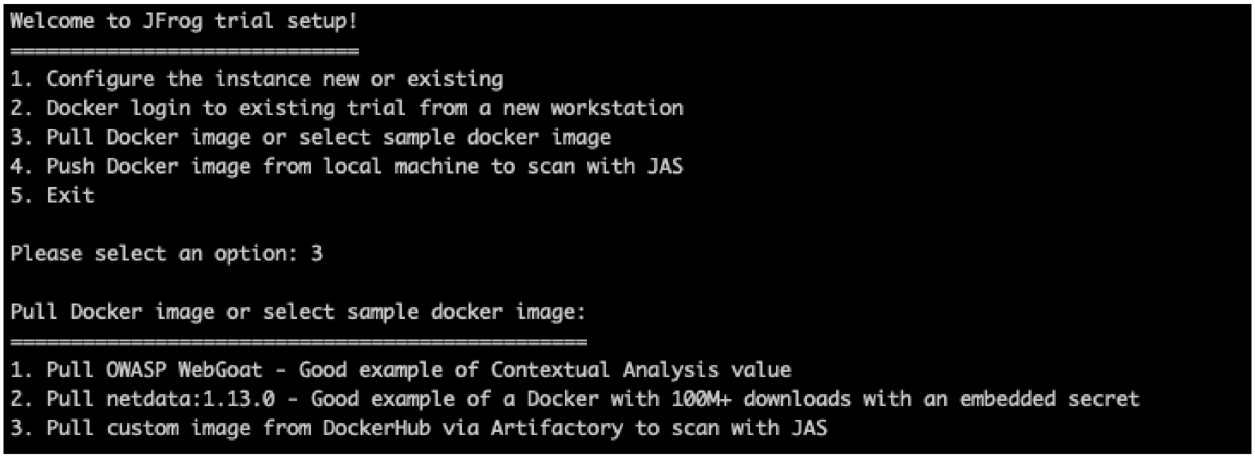
In this lab you will experience JFrog Advanced Security value with actual docker images scanning.

Upon successful completion of this lab you will gain knowledge of how to use the Security issues page and extract relevant value from it

Step by step instructions

Phase #1 - Pulling a docker image:

1. Open the terminal used in Lab 1, or, in case you've closed it, open a new one and run:
   `bash guided-trial/linux_guided_trial.sh`
2. From the menu, select option #3:
   `Pull Docker image or select sample docker image`



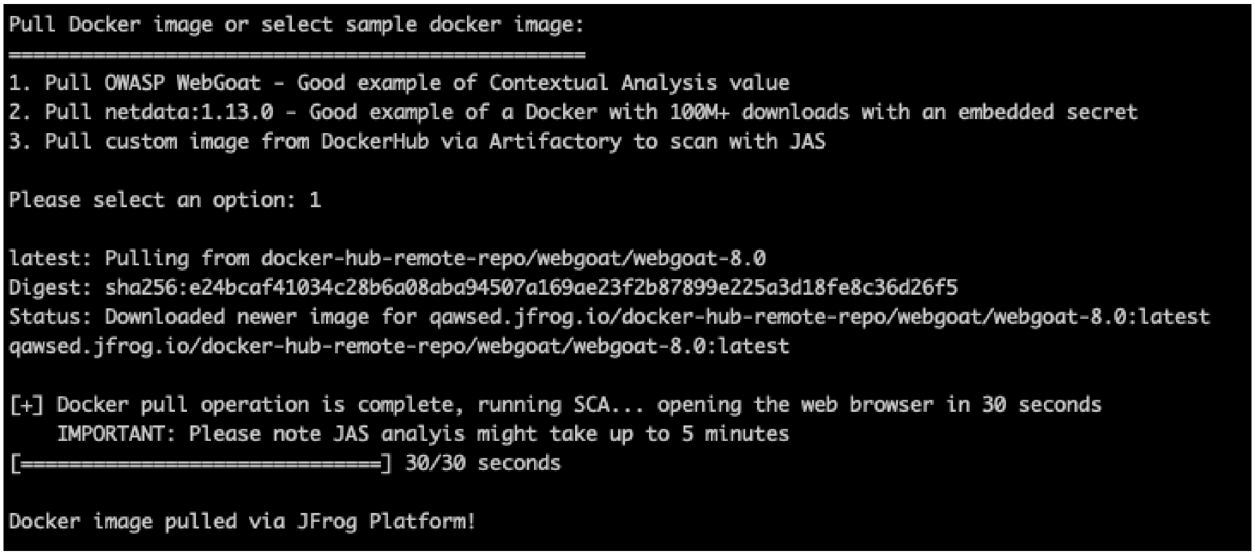
```
Welcome to JFrog trial setup!
=============================
1. Configure the instance new or existing
2. Docker login to existing trial from a new workstation
3. Pull Docker image or select sample docker image
4. Push Docker image from local machine to scan with JAS
5. Exit

Please select an option: 3

Pull Docker image or select sample docker image:
================================================
1. Pull OWASP WebGoat - Good example of Contextual Analysis value
2. Pull netdata:1.13.0 - Good example of a Docker with 100M+ downloads with an embedded secret
3. Pull custom image from DockerHub via Artifactory to scan with JAS
```

3. Now select 'WebGoat', option #1:

```
Pull OWASP WebGoat - Good example of Contextual Analysis value
```



```
Pull Docker image or select sample docker image:
================================================
1. Pull OWASP WebGoat - Good example of Contextual Analysis value
2. Pull netdata:1.13.0 - Good example of a Docker with 100M+ downloads with an embedded secret
3. Pull custom image from DockerHub via Artifactory to scan with JAS

Please select an option: 1

latest: Pulling from docker-hub-remote-repo/webgoat/webgoat-8.0
Digest: sha256:e24bcaf41034c28b6a08aba94507a169ae23f2b87899e225a3d18fe8c36d26f5
Status: Downloaded newer image for qawsed.jfrog.io/docker-hub-remote-repo/webgoat/webgoat-8.0:latest
qawsed.jfrog.io/docker-hub-remote-repo/webgoat/webgoat-8.0:latest

[+] Docker pull operation is complete, running SCA... opening the web browser in 30 seconds
    IMPORTANT: Please note JAS analyis might take up to 5 minutes
[==============================] 30/30 seconds

Docker image pulled via JFrog Platform!
```

Note how the docker image is being pulled from Docker Hub, through Artifactory to your personal laptop.

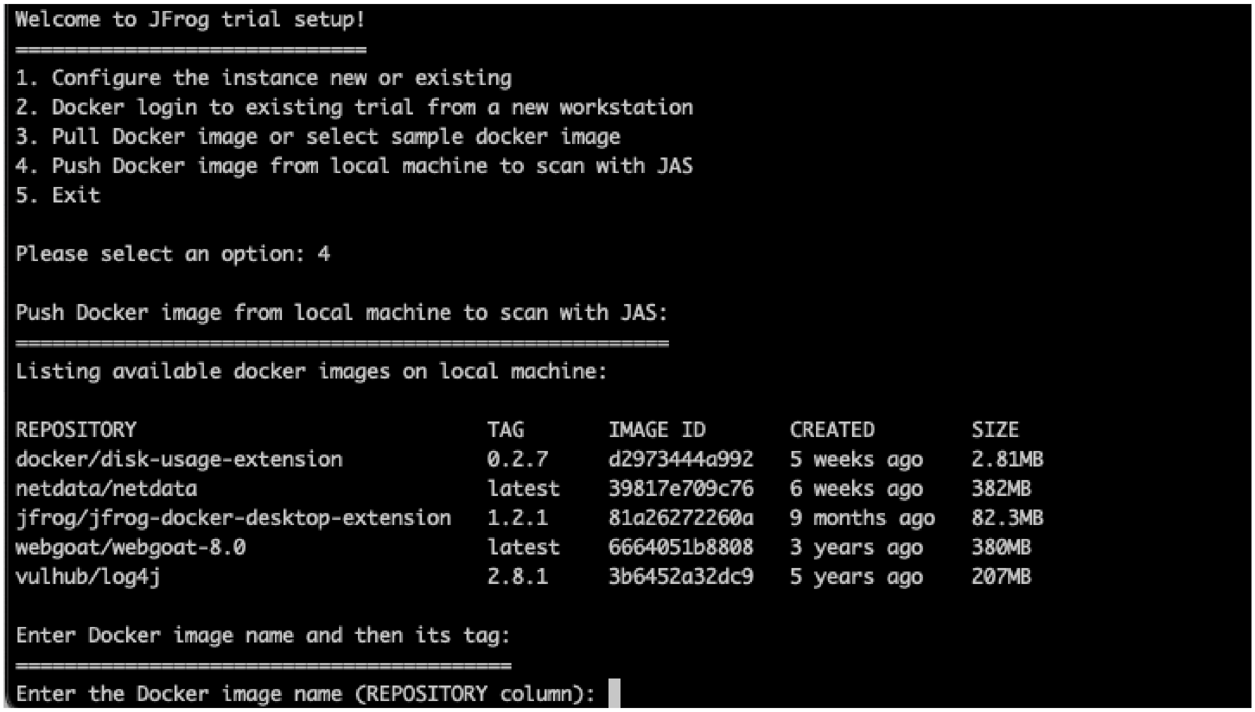
Your browser will be opened to your server’s scan results page (results may take up to 5 min to complete).

4. Look at “CVE-2022-22965”
    a. Is it applicable to this docker image?
    b. What is the risk?
    c. What is the remediation process?
5. Now look at “CVE-2023-20873”
    a. Note the CVSS score of 9.8!
    b. Why is it not applicable to this docker image?

Phase #2 - Pushing a docker image:

6. Go back to your terminal & select option #4 from the menu:

Push Docker image from local machine to scan with JAS



```
Welcome to JFrog trial setup!
=============================
1. Configure the instance new or existing
2. Docker login to existing trial from a new workstation
3. Pull Docker image or select sample docker image
4. Push Docker image from local machine to scan with JAS
5. Exit

Please select an option: 4

Push Docker image from local machine to scan with JAS:
======================================================
Listing available docker images on local machine:

REPOSITORY                             TAG       IMAGE ID       CREATED        SIZE
docker/disk-usage-extension            0.2.7     d2973444a992   5 weeks ago    2.81MB
netdata/netdata                        latest    39817e709c76   6 weeks ago    382MB
jfrog/jfrog-docker-desktop-extension   1.2.1     81a26272260a   9 months ago   82.3MB
webgoat/webgoat-8.0                    latest    6664051b8808   3 years ago    380MB
vulhub/log4j                           2.8.1     3b6452a32dc9   5 years ago    207MB

Enter Docker image name and then its tag:
=========================================
Enter the Docker image name (REPOSITORY column):
```

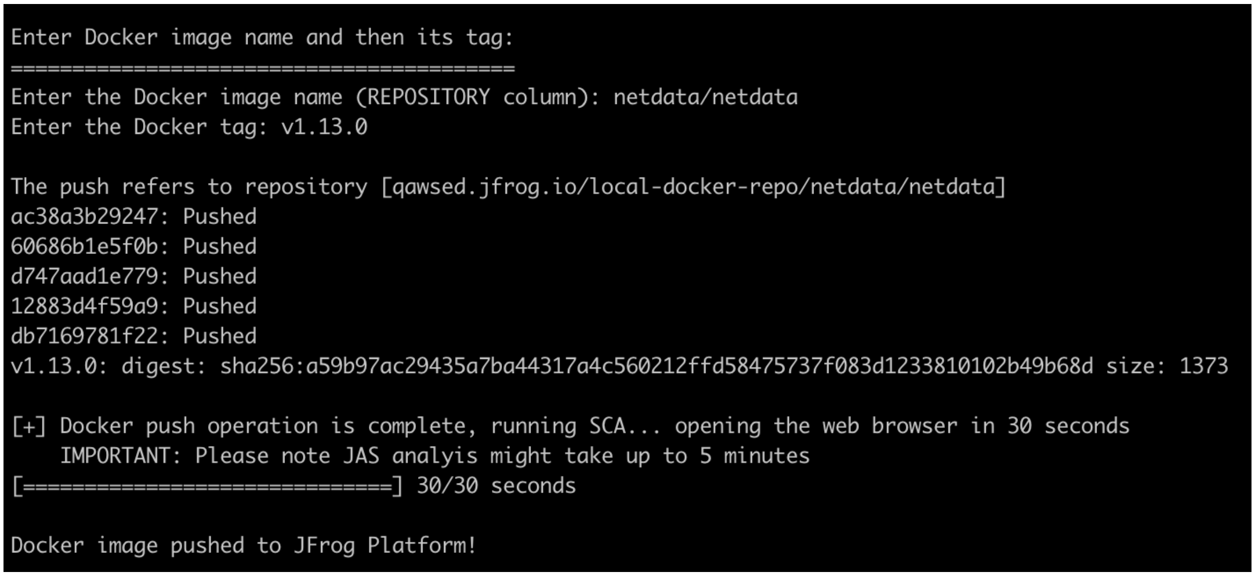
Select a docker image from the list of available images on your laptop and push it.

See how the image is uploaded to Artifactory.

Note: If you do not have one in your workstation, run in your terminal: " `docker pull netdata/netdata:v1.13.0`*"*

The examples below are using the public netdata image.

Your browser will be opened to your server's scan results page (results may take up to 5 min to complete).



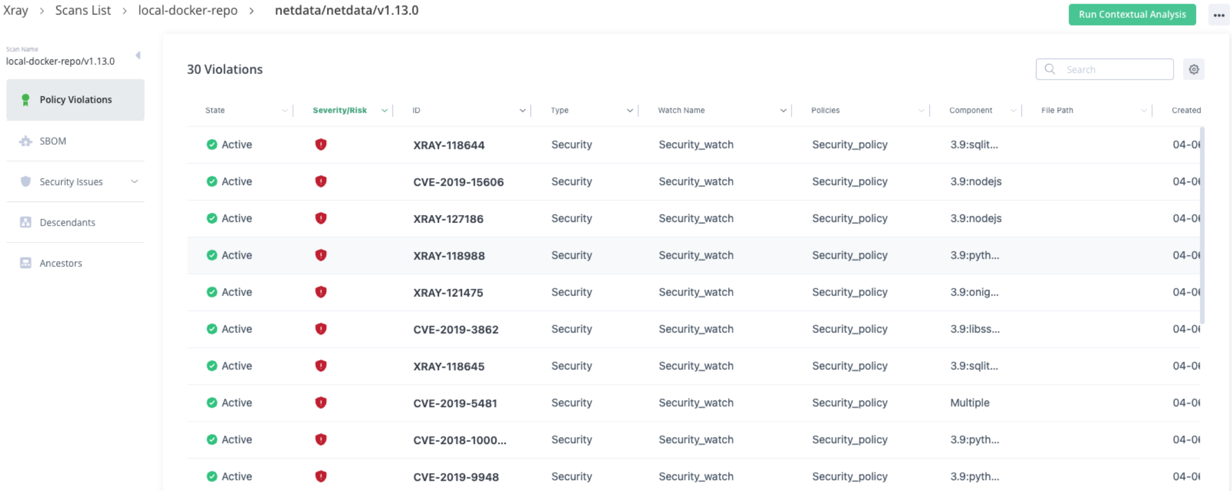
```
Enter Docker image name and then its tag:
==========================================
Enter the Docker image name (REPOSITORY column): netdata/netdata
Enter the Docker tag: v1.13.0

The push refers to repository [qawsed.jfrog.io/local-docker-repo/netdata/netdata]
ac38a3b29247: Pushed
60686b1e5f0b: Pushed
d747aad1e779: Pushed
12883d4f59a9: Pushed
db7169781f22: Pushed
v1.13.0: digest: sha256:a59b97ac29435a7ba44317a4c560212ffd58475737f083d1233810102b49b68d size: 1373

[+] Docker push operation is complete, running SCA... opening the web browser in 30 seconds
    IMPORTANT: Please note JAS analyis might take up to 5 minutes
[==============================] 30/30 seconds

Docker image pushed to JFrog Platform!
```

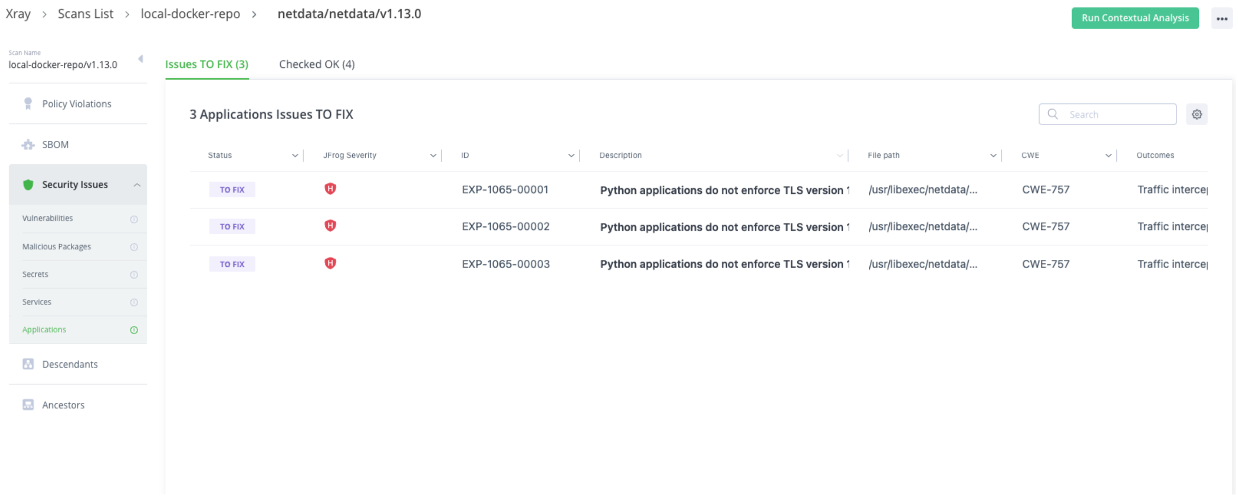
7. How many CVEs can be found in your selected docker images?
8. Do you see any High/Critical CVEs that are not applicable? Why?
9. Does your selected image have any Policy violations?



30 Violations

State	Severity/Risk	ID	Type	Watch Name	Policies	Component	File Path	Created
Active	●	XRAY-118644	Security	Security_watch	Security_policy	3.9:sqlit...		04-0
Active	●	CVE-2019-15606	Security	Security_watch	Security_policy	3.9:nodejs		04-0
Active	●	XRAY-127186	Security	Security_watch	Security_policy	3.9:nodejs		04-0
Active	●	XRAY-118988	Security	Security_watch	Security_policy	3.9:pyth...		04-0
Active	●	XRAY-121475	Security	Security_watch	Security_policy	3.9:onig...		04-0
Active	●	CVE-2019-3862	Security	Security_watch	Security_policy	3.9:libss...		04-0
Active	●	XRAY-118645	Security	Security_watch	Security_policy	3.9:sqlit...		04-0
Active	●	CVE-2019-5481	Security	Security_watch	Security_policy	Multiple		04-0
Active	●	CVE-2018-1000...	Security	Security_watch	Security_policy	3.9:pyth...		04-0
Active	●	CVE-2019-9948	Security	Security_watch	Security_policy	3.9:pyth...		04-0

10. Does your selected image have any application exposures?

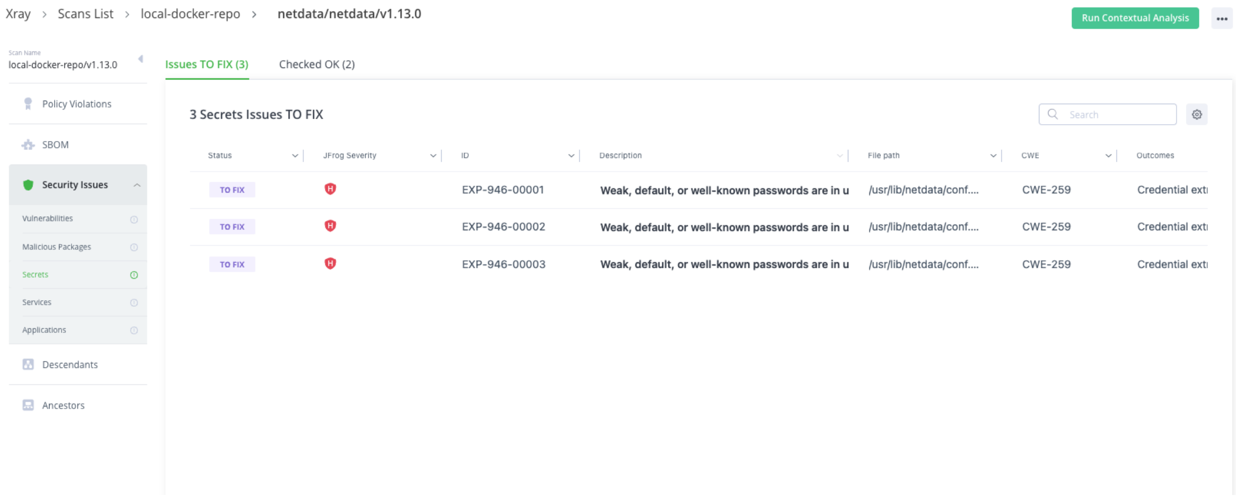


Xray > Scans List > local-docker-repo > netdata/netdata/v1.13.0

3 Applications Issues TO FIX

Status	JFrog Severity	ID	Description	File path	CWE	Outcomes
TO FIX	H	EXP-1065-00001	Python applications do not enforce TLS version 1	/usr/libexec/netdata/...	CWE-757	Traffic interce
TO FIX	H	EXP-1065-00002	Python applications do not enforce TLS version 1	/usr/libexec/netdata/...	CWE-757	Traffic interce
TO FIX	H	EXP-1065-00003	Python applications do not enforce TLS version 1	/usr/libexec/netdata/...	CWE-757	Traffic interce

11. Does your selected image have any secrets detected?



Xray > Scans List > local-docker-repo > netdata/netdata/v1.13.0

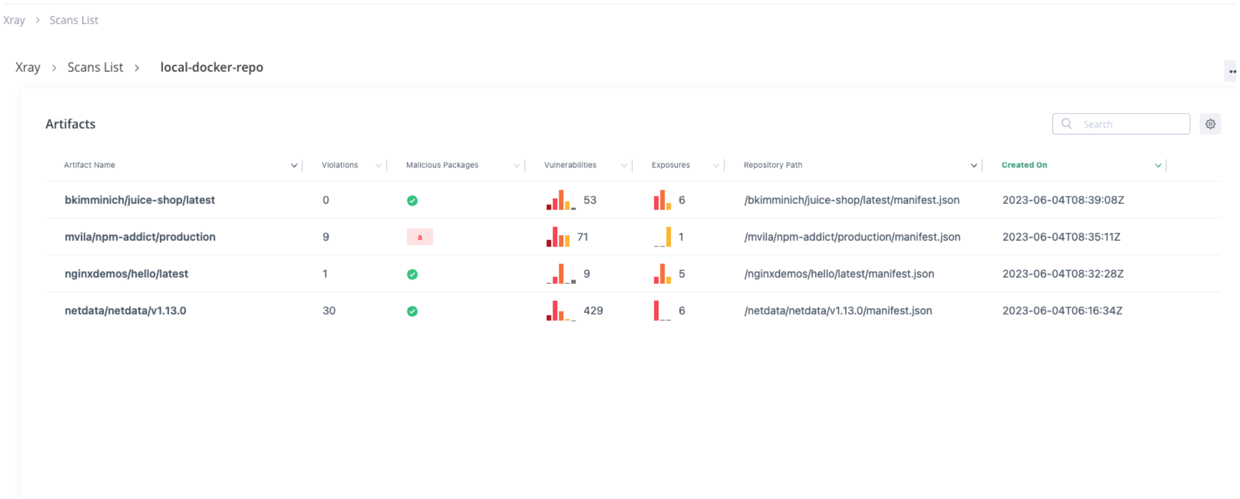
3 Secrets Issues TO FIX

Status	JFrog Severity	ID	Description	File path	CWE	Outcomes
TO FIX	H	EXP-946-00001	Weak, default, or well-known passwords are in u	/usr/lib/netdata/conf....	CWE-259	Credential ext
TO FIX	H	EXP-946-00002	Weak, default, or well-known passwords are in u	/usr/lib/netdata/conf....	CWE-259	Credential ext
TO FIX	H	EXP-946-00003	Weak, default, or well-known passwords are in u	/usr/lib/netdata/conf....	CWE-259	Credential ext

Congratulations! You have completed Lab 2

Phase #3 - Advanced

12. Browse through the PDF in your guided trial folder and read/experiment with the system other capabilities and features
13. Push additional popular docker hub images to view the results
    a. `mvila/npm-addict:production` - This image has a malicious package.
    b. `bkimminich/juice-shop` - This has Application and Secret Exposures.
    c. `nginxdemos/hello:latest` - This has Service Exposures (nginx)



Xray > Scans List > local-docker-repo

Artifacts

Artifact Name	Violations	Malicious Packages	Vulnerabilities	Exposures	Repository Path	Created On
bkimminich/juice-shop/latest	0	✓	53	6	/bkimminich/juice-shop/latest/manifest.json	2023-06-04T08:39:08Z
mvila/npm-addict/production	9	a	71	1	/mvila/npm-addict/production/manifest.json	2023-06-04T08:35:11Z
nginxdemos/hello/latest	1	✓	9	5	/nginxdemos/hello/latest/manifest.json	2023-06-04T08:32:28Z
netdata/netdata/v1.13.0	30	✓	429	6	/netdata/netdata/v1.13.0/manifest.json	2023-06-04T06:16:34Z

OLD

~~**Prerequisites** are a Mac/Windows/Linux workstation with:~~

- ~~'curl' command~~
- ~~docker client installed & running~~
- ~~Docker image(s) on your local machine~~

~~**Step 1 - Set up and configure a JAS trial instance**~~

1. ~~Launch a JAS trial instance using the script & instructions detailed in: https://jfrog.com/start-free/security/~~

~~**Step 1 - Set up and configure a JAS trial instance**~~

~~**Run the script option (*1. Launch and configure a new trial*)** and follow the prompt instructions.~~

~~Follow the prompt messages in the console:~~

1. ~~Launch a JAS trial instance with your Jfrog/Gmail email address.~~
2. ~~A local docker repository is created with JAS configured~~
3. ~~A remote docker repository is created with JAS configured~~
4. ~~A security policy is created in Xray, the policy is set to create a violation upon critical CVEs and High Exposures~~
5. ~~A watch is created in Xray, and applies to your docker repositories~~

~~**Step 2 - Scan DockerHub docker images and answer security questions**~~

~~**Run the script option (*3. Pull Docker image or select sample docker image*)**.~~

1. ~~Scan the 2 Appendix A dockers with JAS~~
2. ~~Answer the following security questions~~

a. ~~What is the Contextual Analysis scan result of "CVE-2019-20367" in the "Webgoat" docker?~~

b. ~~Give 1 example of a secret found in "netdata" Docker?~~

~~Step 3 - Scan a local docker image from your workstation~~

~~Run the script option (*4. Push Docker image from local machine to scan with JAS*)~~

1. ~~Push a docker image (of your choice) from your local workstation to the local docker repository.~~

   ~~Note: If you do not have one in your workstation, run in your terminal: "docker pull alpine"~~

2. ~~Review the JAS scans results and see if there were any JAS findings or applicable/non-applicable CVEs. If no JAS findings are found, report how many CVEs were found in the image.~~

~~Congratulations!~~

~~You have successfully completed the JFrog Security quick trial!~~

~~The Trial Environment will be kept available for you in the coming two weeks.~~

~~Feel free to further experiment with it and reach out to us for further questions and discussions with our Security and DevOps experts.~~

~~Appendix A - List of sample dockers from DockerHub:~~

- ~~webgoat/webgoat-8.0:latest~~
- ~~netdata/netdata:v1.33.1~~

~~Appendix B - Download-able Trial Zip with Script & detailed screenshots~~

- ~~https://releases.jfrog.io/artifactory/website/security/guided-trial.zip~~