JFTD-102 JFrog Platform Essentials

Hands-on Labs

Lab 1: Artifactory

1. Assigned pre-created repositories to you project
    a. `swampup17242726641.jfrog.io/` Hint: Participants JPDs are preconfigured with credentials
    b. Observation: Participant is able to login & perform operation
    c. Conclusion: Local users are provisioned
    d. Open the project, create and assign local and remote repositories to your project. (name of repositories will be provided based on seating)
2. Create a local and remote repository
    a. Log into the provided Jfrog platform with the username/password you were provided.
    b. Select your project from the project dropdown in the top right of the page
    c. Click the “administration” section of the platform to the left of the project dropdown.
    d. Under project resources select create repositories under resources on the right side of the page.
        i. Create a local DOCKER repository
            1. Fill in the repository key that will be the name of the local repository.
            2. Click the create repository button at the bottom of the page.
        ii. Create a remote DOCKER repository
            1. Fill in the repository key that will be the name of the remote repository. NOTE: Keep the URL section the same.
            2. Click the create repository button at the bottom of the page.
3. Create an Access Token on the JFrog UI
    a. Head to the administration side of the platform.
    b. Under the users section click access tokens on the right side panel.
    c. Click Generate Token on the top right corner of the page
    d. Select project Admin role and hit generate.
    e. Copy the token for use later in the lab
4. Download JFrog CLI, configure to connect to the JPD, and do a “ping”
    a. Hint: Download from web directly, also do this ahead of class
    b. Open a terminal window
        i. Windows in the search bar type CMD and run
        ii. MAC: in the search bar type in terminal
        iii. Run command jf config add
        iv. Give the configuration a name “SwampUp2024”

v. Provide URL "`swampup17242726641.jfrog.io`"
vi. Next prompt just hit enter
vii. For authentication select Access Token or username and password
viii. Continue prompts without changes by hitting enter
ix. Test with jf rt ping
    1. You should get an OK response.

5. Run Docker login command using Set Me Up functionality in Artifactory
    a. `docker login swampup17242726641.jfrog.io`
6. Download a docker image from Docker Hub using JFrog CLI (jf command)
    a. Hint: A docker image (`docker pull swampup17242726641.jfrog.io/{REMOTE-REPO-NAME}/infinitengine/swampup:1.0.0`) is identified for participants to download
    b. Observation: Participant performs operation
    c. Conclusion: Participant understands CLI operations to use remote repositories' operation to download
7. BONUS: Configure VSCode with JFrog IDE Plugin

Lab1: step 3
https://jfrog.com/getcli/
Adding and Editing Configured Servers
jf config add / edit

(Follow configuration instructions)

Ping server
jf rt ping
To specify server:
jf rt ping --server-id=SERVER_ID

Lab 2: Xray

1. Enable Xray Indexing on Local & Remote repositories
    a. Hint: Docker repositories with existing scanned docker images (in another docker repositories) are preferred
    b. Observation: Participants perform the operation
    c. Conclusion: Participants understand CVEs
2. Upload a Docker image, using docker native commands, that was downloaded in Lab1
    a. Hint: Docker images with 1st party code preferred
    b. Observation: Participants perform the operation
    c. Conclusion: Participants understand CVEs, CWEs, Secrets, etc.
3. BONUS: Scan the same docker image locally in their workstation using JFrog CLI
    a. Hint: Use the docker image from Docker Hub (interested only in CVEs)
    b. Observation: Participants perform the operation
    c. Conclusion: Participants understand CLI on-demand scanning

Lab 2: Solution

1. Administration -> Xray Settings -> Indexed Repositories -> Add a Repository
2. docker push soleng.jfrog.io/docker-virtual/infinitengine/swampup:1.0.0
3. jf docker scan swampup:1.0.0

Lab 3: Distribution

1. Create a Release Bundle, and Distribute to an Edge node
    a. Hint: Use previously uploaded Docker images in a Local repository,
    b. Observation: Participants perform the operation
    c. Conclusion: Participants understand Distribution process
2. Head to Distribution -> Release Bundles -> + New Release Bundle
    a. Name the release bundle UserNameProjectName
    b. Version the release bundle 1.0.7
    c. Create Query -> Add Query
    d. Query name should be “UserNameAQL”
    e. Add the local repository from the previous labs to “Repository Names” drop down
    f. OPTIONAL: click show AQL check box at bottom of page to see AQL format
    g. Click Next and Next. Save release bundle
    h. Click Create and Sign button
3. Distribute your signed release bundle
    a. Click versioned actions in the top right corner.
    b. Click the distribute version
    c. Select the edge node to distribute your release bundle too.
    d. Click the distribute button.

JFROG SETUP: JPD w/ Edge. Has Docker & NPM repos configured. Indexing enabled. Docker images are scanned for Basic and Advanced Security. Keep JPDs alive for a week after the session (Sep 20th). GitHub account

WORKSTATION SETUP: Git CLI, Docker Runtime, JFrog CLI, NPM Native Client, IDE (VSCode, IntelliJ), Web Browser (Chrome)

Lab1
Adding and Editing Configured Servers
jf config add / edit

Ping server
jf rt ping
To specify server:
jf rt ping --server-id=SERVER_ID

docker login
docker pull sup111epsumXX.jfrog.io/docker-hub-remote-repo/nginx:mainline-alpine3.18-slim
docker push

nginxdemos/hello:latest