ASSIGNMENT2

1)Write the Python code to send the Jar to the Jfrog repository

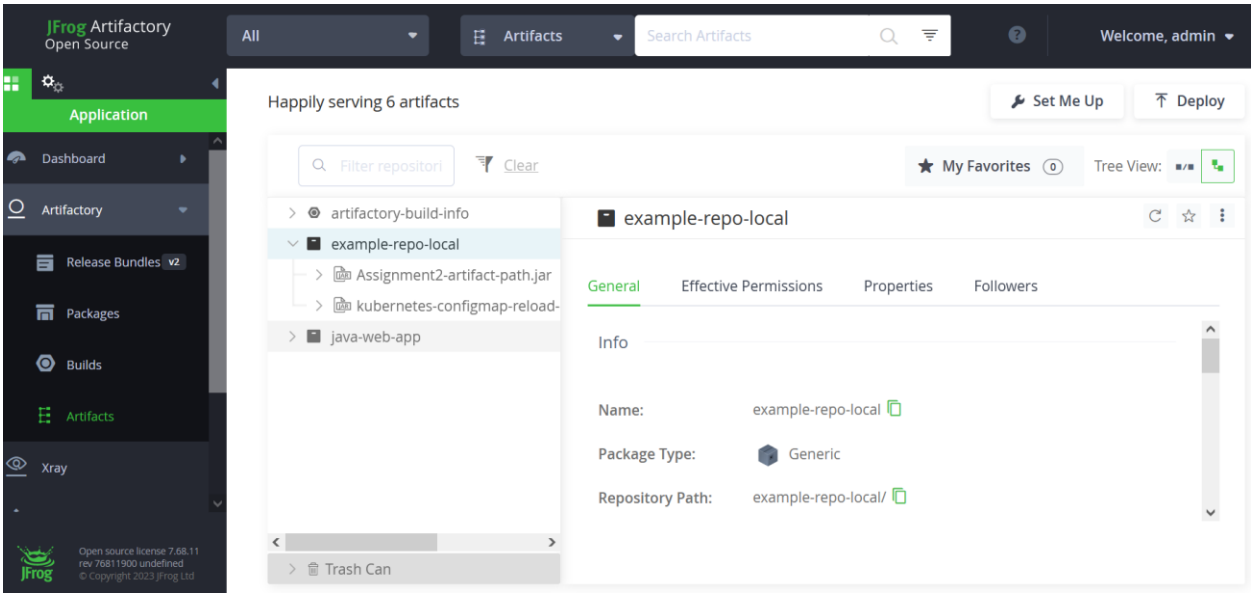
Jfrog.py code:

```
import requests

def jfrogUpload():
    url = "http://3.85.54.23:8082/artifactory/example-repo-local/"
    file_path = "/var/lib/jenkins/workspace/Demo_3.0/target/kubernetes-configmap-reload-0.0.1-SNAPSHOT.jar"
    username = "admin"
    password = "Password"

    # Construct the URL for uploading
    upload_url = f"{url}Assignment2-artifact-path.jar"  # Replace 'your-artifact-path.jar' with the desired destination in your Artifactory repository

    try:
        # Create a session and set basic authentication
        session = requests.Session()
        session.auth = (username, password)

        # Perform the file upload
        with open(file_path, 'rb') as file:
            response = session.put(upload_url, data=file)

        if response.status_code == 201:
            print(f"Upload successful: {upload_url}")
        else:
            print(f"Upload failed with status code {response.status_code}: {response.text}")
    except Exception as e:
        print(f"An error occurred: {str(e)}")

if __name__ == "__main__":
    jfrogUpload()
```

Execute Python file:

python3 Jfrog.py

Output:



Second Task given in below PAGE:

↧↧↧

2)Write the Groovy Pipeline Script stage to Integrate the Jfrog in the CI Pipeline.

First Added Three Parameters in Jenkins file for getting the Artifact Build Path, Version number and Target path from user with some default values.

```
string(name: 'artifactPath', description: "Jfrog Artifact Path", defaultValue:
'/var/lib/jenkins/workspace/Demo_3.0/target/kubernetes-configmap-reload-0.0.1-
SNAPSHOT.jar')

string(name: 'version', description: "Artifact version", defaultValue: '1')
string(name: 'targetPath', description: "Target path of Jfrog repo",
defaultValue: 'http://<Server-Ip>:8082/artifactory/java-web-app/Assignment-2-
${version}.jar')
```

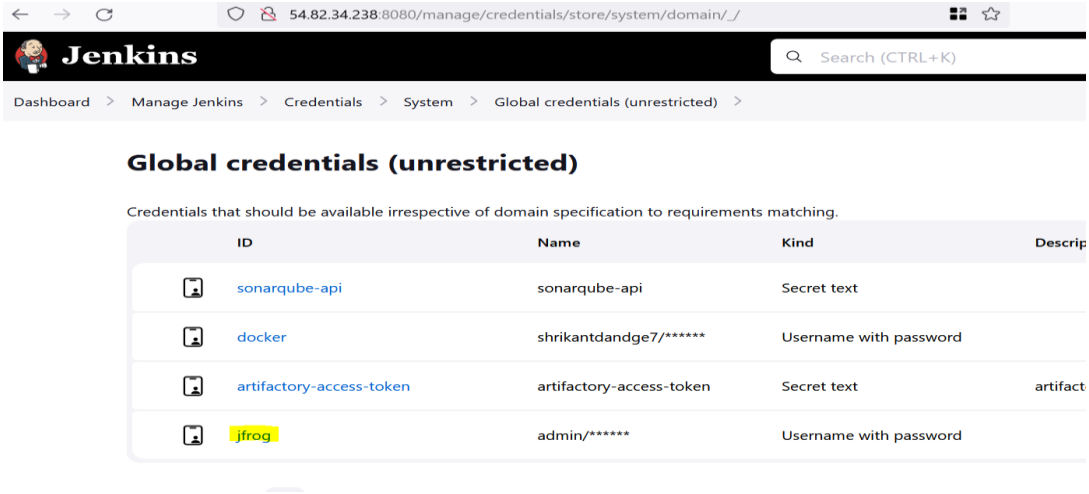
Then, Added following stage to pass the parameters to shared library function `JfrogPush()`.

```
stage('Jfrog Artifact push: Jfrog'){
         when { expression {  params.action == 'create' } }
            steps{
               script{
                   sh 'pwd'
                   JfrogPush("${params.version}", "${params.artifactPath}",
"${params.targetPath}")
                }
            }
```

Created a new groovy script under 'jenkins_shared_lib/vars/JfrogPush.groovy' in my git repo.

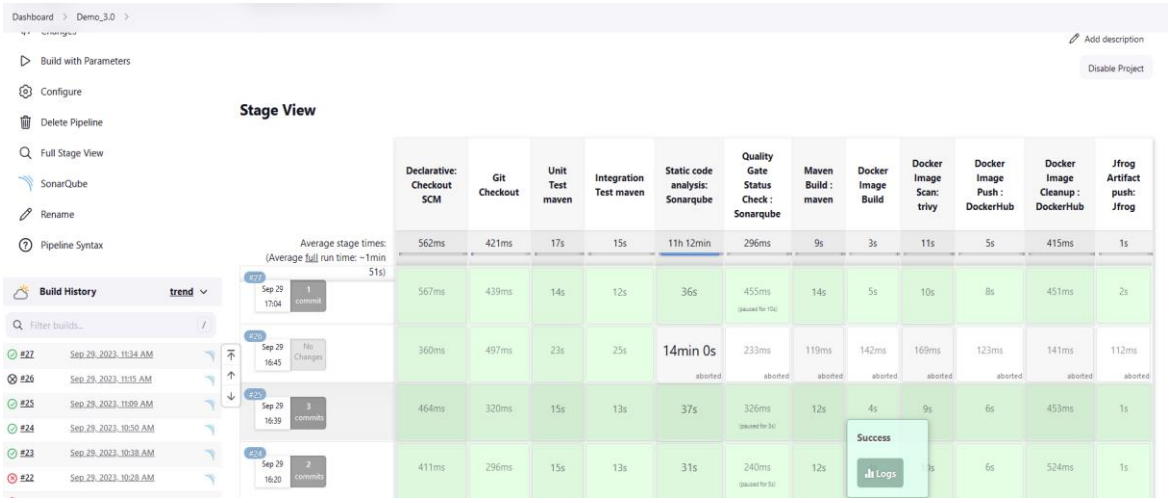
```
def call(String version, String artifactPath, String targetPath) {
        withCredentials([usernamePassword(
            credentialsId: 'jfrog',
            usernameVariable: 'USER',
            passwordVariable: 'PASS'
        )]) {
            sh "curl -u $USER:$PASS -X PUT -T ${artifactPath} ${targetPath}"
            echo "Uploaded JAR artifact assignment to Artifactory: ${targetPath}
of version ${version} "
        }
}
```

After that added the credentials of type username and password in Jenkins for ID “jfrog”.

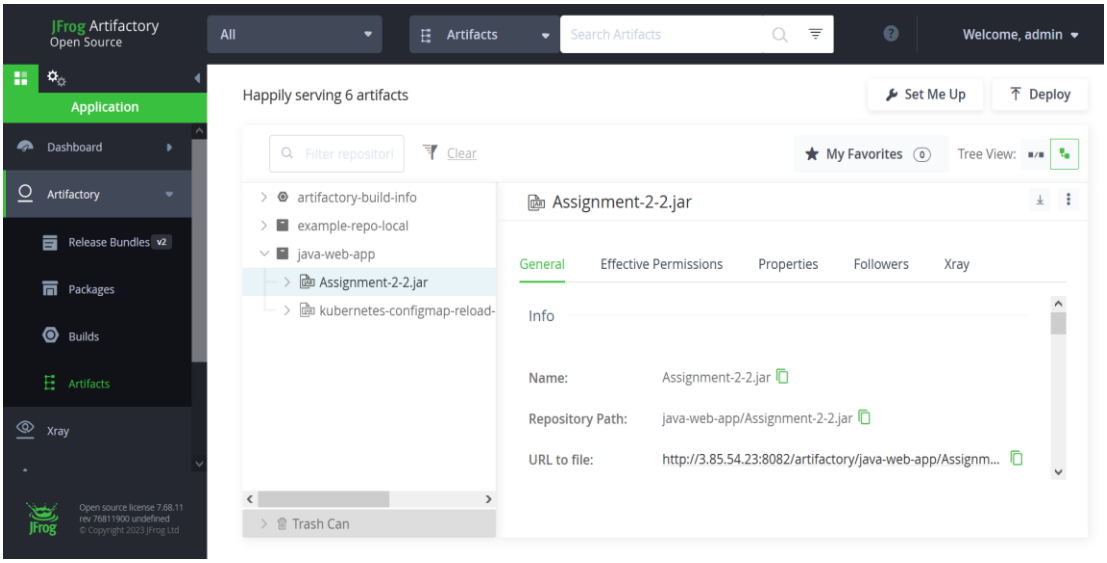


Finally, I ran the Jenkins Pipeline with default parameters and the artifact kubernetes-configmap-reload-0.0.1-SNAPSHOT.jar pushed to the Jfrog Repository path http://<Server-ip>:8082/artifactory/java-web-app/ Assignment-2-<version-number>.jar.

Jenkins Pipeline:



Jfrog Output:



END

++