

Wszystko na podstawie przykładowego repo: <https://github.com/laskmat/jenkins-liquibase-poc>

- ```
jenkins 2023-09-01 08:22:22.822+0000 [id=25] INFO winstone.Logger#logInternal: Winstone Servlet Engine running: controlPort=disabled
jenkins 2023-09-01 08:22:24.400+0000 [id=32] INFO jenkins.InitReactorRunner$1.onAttained: Started initialization
jenkins 2023-09-01 08:22:24.512+0000 [id=52] INFO jenkins.InitReactorRunner$1.onAttained: Listed all plugins
jenkins 2023-09-01 08:22:29.181+0000 [id=43] INFO jenkins.InitReactorRunner$1.onAttained: Prepared all plugins
jenkins 2023-09-01 08:22:29.211+0000 [id=38] INFO jenkins.InitReactorRunner$1.onAttained: Started all plugins
jenkins 2023-09-01 08:22:29.243+0000 [id=52] INFO jenkins.InitReactorRunner$1.onAttained: Augmented all extensions
jenkins 2023-09-01 08:22:30.001+0000 [id=43] INFO jenkins.InitReactorRunner$1.onAttained: System config loaded
jenkins 2023-09-01 08:22:30.002+0000 [id=42] INFO jenkins.InitReactorRunner$1.onAttained: System config adapted
jenkins 2023-09-01 08:22:30.003+0000 [id=49] INFO jenkins.InitReactorRunner$1.onAttained: Loaded all jobs
jenkins 2023-09-01 08:22:30.005+0000 [id=38] INFO jenkins.InitReactorRunner$1.onAttained: Configuration for all jobs updated
jenkins 2023-09-01 08:22:30.072+0000 [id=66] INFO hudson.util.Retrier$start: Attempt #1 to do the action check updates server
jenkins WARNING: An illegal reflective access operation has occurred
jenkins WARNING: Illegal reflective access by org.codehaus.groovy.vmplugin.v7.Java$51 (file:/var/jenkins_home/war/WEB-INF/lib/groovy-all-2.4.21.jar) to constructor java.lang.invoke.MethodHandleImpl()
jenkins lang.Class, int)
jenkins WARNING: Please consider reporting this to the maintainers of org.codehaus.groovy.vmplugin.v7.Java$51
jenkins WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
jenkins WARNING: All illegal access operations will be denied in a future release
jenkins 2023-09-01 08:22:31.362+0000 [id=34] INFO jenkins.install.SetupWizard$init:
jenkins *****
jenkins *****
jenkins *****
jenkins Jenkins initial setup is required. An admin user has been created and a password generated.
jenkins Please use the following password to proceed to installation:
jenkins 354bc365fF44f5ae093ddfcb2b611e2
jenkins This may also be found at : /var/jenkins_home/secrets/initialAdminPassword
jenkins *****
jenkins *****
jenkins *****
jenkins *****
jenkins 2023-09-01 08:22:54.891+0000 [id=46] INFO jenkins.InitReactorRunner$1.onAttained: Completed initialization
jenkins 2023-09-01 08:22:54.954+0000 [id=24] INFO hudson.lifecycle.LifecycleReady: Jenkins is fully up and running
jenkins 2023-09-01 08:22:56.695+0000 [id=66] INFO h.s.DownloadService$downloadableLoad: Obtained the updated data file for hudson.tasks.Maven.MavenInstaller
jenkins 2023-09-01 08:22:56.697+0000 [id=66] INFO hudson.util.Retrier$start: Performed the action check updates server successfully at the attempt #1
```

```

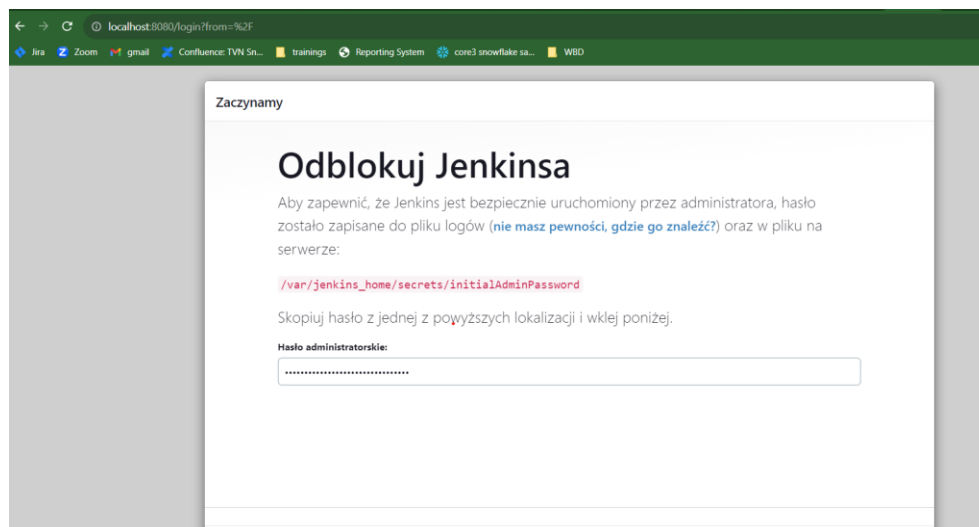
$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
6864494c5a03 jenkins-liquibase-poc-jenkins_service "/usr/bin/tini -- /u-" 7 minutes ago Up 7 minutes 0.0.0.0:8080->8080/tcp, 0.0.0.0:50000->50000/tcp jenkins

$ docker exec -it 6864494c5a03 bash
jenkins@6864494c5a03:/$ cat /var/jenkins_home/secrets/initialAdminPassword
354b2c3652f144f5ae093ddfb2b611e2
jenkins@6864494c5a03:/$ |

```

1. docker ps
2. docker exec -it <container id> bash
3. cat /var/jenkins\_home/secrets/initialAdminPassword
4. mamy hasło

3. wejść na <http://localhost:8080/> i jak się załaduje to wkleić hasło



3. wybrać standardowe wtyczki do zainstalowania  
4. dodać administratora (na przykładzie dałem credentiale jako admin/admin)

Dodawanie pierwszego użytkownika

## Stwórz pierwszego administratora

Login

Hasło

Powtórz hasło

Pełna nazwa

Adres e-mail

Jenkins 2.421

Kontynuuj jako administrator

Zapisz i zakończ

## Przygotowanie środowiska

Pluginy:

<https://plugins.jenkins.io/docker-plugin/>

<https://plugins.jenkins.io/docker-build-step/>

<https://plugins.jenkins.io/credentials-binding/>

<https://plugins.jenkins.io/envinject-api/>

<https://plugins.jenkins.io/envinject/>

<https://plugins.jenkins.io/workflow-aggregator/>

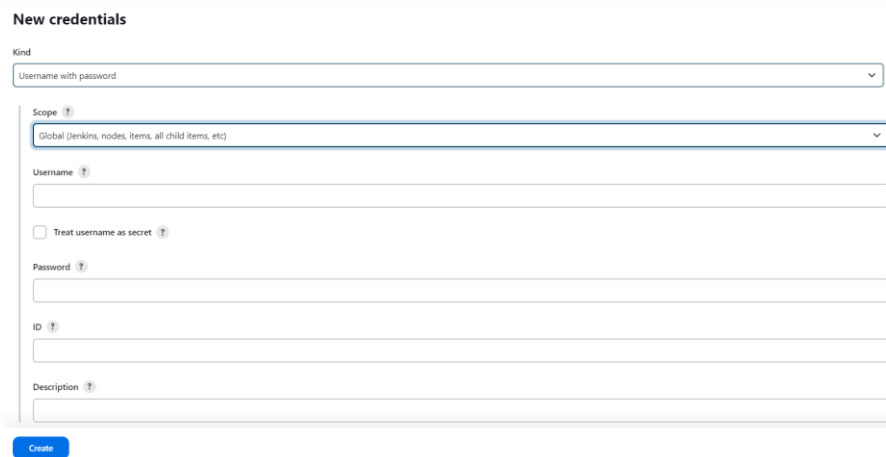
<https://plugins.jenkins.io/git/>

<https://plugins.jenkins.io/git-client/>

## Konfiguracje

### 1. Credentials – dla Snowflake

#### a. Zarządzaj Jenkinsem -> Credentials -> Global Credentials -> Add Credentials



**New credentials**

Kind: Username with password

Scope: Global (jenkins, nodes, items, all child items, etc)

Username

☐ Treat username as secret

Password

ID

Description

Create

#### b.

#### c. Create

### 2. Projekt – docelowy pipeline zawierający migracje

#### a. Nowy Projekt -> Pipeline

#### b. Potrzebne checki

- i. Do not allow concurrent builds
- ii. GitHub project - I wkleić link do repo
- iii. Prepare an environment for the run
  1. Keep Jenkins Environment Variables
  2. Keep Jenkins Build Variables

#### c. W polu: **Zawartość Właściwości** wpisać zmienne:

- i. warehouse="<nazwa warehouse>"
- ii. db="nazwa bazy"

- iii. `schema="nazwa schemy"`
- iv. `role="rola dla użytkownika który ma uprawnienia do zmiany obiektów w bazie"`
- v. `account="nazwa konta snowflake – pierwsza część z`

`.snowflakecomputing.com "`

d. Powinno to wyglądać mniej więcej tak

Prepare an environment for the run ?

- ☒ Keep Jenkins Environment Variables ?
- ☒ Keep Jenkins Build Variables ?
- ☐ Override Build Parameters

Ścieżka Pliku Właściwości ?

Zawartość Właściwości ?

```
warehouse="POC_WAREHOUSE"
db="DBT_VAULT_POC"
schema="PUBLIC"
role="POC"
account="qu13708.eu-central-1"
```

Ścieżka do Pliku Skryptu ?

Zawartość Skryptu ?

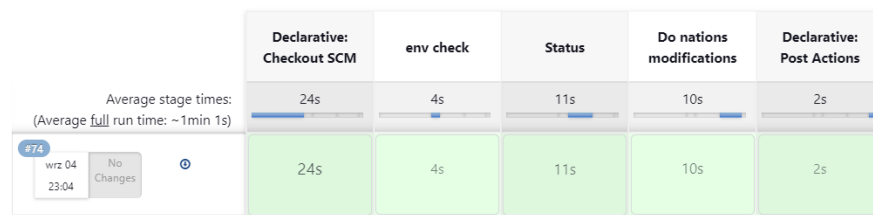
Zapisz Zastosuj

e. Pozycja **Pipeline**

- i. Definition -> Pipeline Scrip from SCM (WAŻNE!)
- ii. SCM -> GIT
  - Wypełnić danymi z repo + wymagane credentiale dostępowe (można sobie dodać klucze SSH)
  - Branches to build -> /main (lub master lub cokolwiek gdzie leży kod migracji)
  - Script Path -> tutaj trzeba wpisać nazwę pliku z definicją pipeline. Zwyczajowo: **Jenkinsfile**

f. Przykładowy pipeline:

**liquibase - Stage View**



g. Żeby bardziej to przypominało pipeline to można użyć

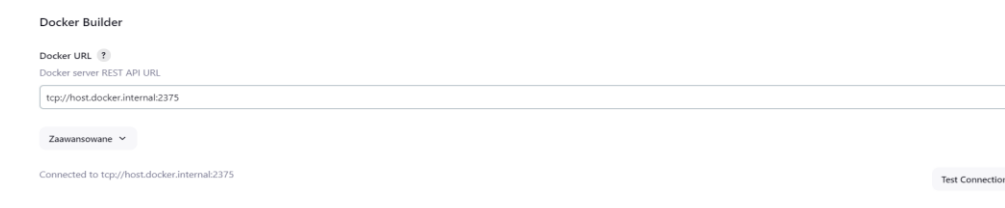
<https://www.jenkins.io/doc/book/blueocean/creating-pipelines/> ale idea będzie ta sama.

## Docker agent dla pipeline


### 1. Konfiguracja w Jenkinsie

a. Zarządzaj Jenkinsem -> System (Global) -> ctrl+f: Docker Builder

i. Docker URL: wkleić url do serwera z odblokowanym DOCKER API. Przykład (dla lokalnej instancji):



Docker Builder

Docker URL 

Docker server REST API URL

tcp://host.docker.internal:2375

Zaawansowane ▾

Connected to tcp://host.docker.internal:2375

Test Connection

Test connection ma być sukcesem.

Przykładowy test czy na danym URLu jest odblokowane API dockera:

**curl --no-buffer -XGET http://localhost:2375/version**

powinno zwrócić

```
{ "Platform": { "Name": "Docker Desktop 4.22.0 (117440)" }, "Components": [{ "Name": "Engine", "Version": "24.0.5", "Details": { "ApiVersion": "1.43", "Arch": "amd64", "BuildTime": "2023-07-21T20:35:45.000000000+00:00", "Experimental": "false", "GitCommit": "a61e2b4", "GoVersion": "go1.20.6", "KernelVersion": "5.10.16.3-microsoft-standard-WSL2", "MinAPIVersion": "1.12", "Os": "linux" } }, { "Name": "containerd", "Version": "1.6.21", "Details": { "GitCommit": "3dce8eb055cbb6872793272b4f20ed16117344f8" } }, { "Name": "runc", "Version": "1.1.7", "Details": { "GitCommit": "v1.1.7-0-g860f061" } }, { "Name": "docker-init", "Version": "0.19.0", "Details": { "GitCommit": "de40ad0" } }], "Version": "24.0.5", "ApiVersion": "1.43", "MinAPIVersion": "1.12", "GitCommit": "a61e2b4", "GoVersion": "go1.20.6", "Os": "linux", "Arch": "amd64", "KernelVersion": "5.10.16.3-microsoft-standard-WSL2", "BuildTime": "2023-07-21T20:35:45.000000000+00:00" }
```

### 2. Dedykowany Jenkins agent dla Liquibase

a. Docker-compose build liquibase\_service --no-cache

i. Opis: dockerfile zawiera w sobie oficjalny obraz liquibase w wersji 4.23.1 z doinstalowanym GITem który jest wymagany na tym runnerze

b. Po zbudowaniu – trzeba agenta wrzucić do dockerhuba. Najlepiej nie oficjalnego, ponieważ on ma ograniczenia w ściąganiu obrazów.

Komendy do uploadu na dockerhub:

**docker tag local-image:tagname new-repo:tagname**

**docker push new-repo:tagname**

## Repozytorium

Przykład: <https://github.com/laskmat/jenkins-liquibase-poc>

Budowanie migracji opiera się na plikach JENKINSFILE-X które zawierają w sobie definicje Pipelines.

<https://github.com/laskmat/jenkins-liquibase-poc/blob/main/Jenkinsfile-3>

UWAGA, część z **credentialsId** wymaga podmianki IDka credentiali dla snowflake

```
stage('Status') {
 steps {
 withCredentials([usernamePassword(credentialsId: '4f53ec09-6106-4a31-b1bc-b69cf00c4229', passwordVariable: 'password', usernameVariable: 'username')]) {
 sh 'liquibase status --url="jdbc:snowflake://https://$account.snowflakecomputing.com/?user=$username&warehouse=$warehouse&db=$db&schema=$schema&role=$role" --'
```

| T | P | Store i | Domain   | ID                                   | Name                                |
|---|---|---------|----------|--------------------------------------|-------------------------------------|
|   |   | System  | (global) | jenkins                              | jenkins                             |
|   |   | System  | (global) | 4f53ec09-6106-4a31-b1bc-b69cf00c4229 | MATEUSZ_LASKOWSKI/***** (SNOWFLAKE) |
|   |   | System  | (global) | secret_liquibase                     | liquibase.docker.properties         |
|   |   | System  | (global) | snowflake_vars                       | .env                                |

Stores scanned to Jenkins

WithCredentials – rozdziela credentials z postaci **user:password**

Pliki migracji są w postaci xml i zostały zagregowane w pliku o nazwie **dataset\_wrapper.xml**. Kolejne pliki migracji dodaje się w postaci:

```
<include file="<kolejna nazwa pliku>.xml"/>
```

a plik z zagregowanymi migracjami przekazuje się przez parametr **--changeLogFile=dataset\_wrapper.xml** do liquibase.

Wynik Liquibase status:

```
#####
##
| | () () |
| | _ _ _ _ _ | _ _ _ _ _
| | | / _ | | | | \ \ / _ | \ \
| | | (| | | | | | | (| \ \ /
\ _ / \ \ | \ _ | | _ / \ _ /
| |
| |
##
Get documentation at docs.liquibase.com
Get certified courses at learn.liquibase.com
##
#####
Starting Liquibase at 13:19:08 (version 4.23.1 #12042 built at 2023-08-10 13:48+0000)
Liquibase Version: 4.23.1
Liquibase Open Source 4.23.1 by Liquibase
6 changesets have not been applied to KACPER_KMIECIK@jdbc:snowflake://qu13708.eu-central-1.snowflakecomputing.com:443/
core-engine/tables.xml::core-engine-2023-03-23-0113::kkmecik
core-engine/tables.xml::core-engine-2023-03-23-0114::kkmecik
core-engine/tables.xml::core-engine-2023-03-20-3011::kkmecik
core-engine/tables.xml::core-engine-2023-03-20-3021::kkmecik
core-engine/tables.xml::core-engine-2023-03-20-541::kkmecik
core-engine/tables.xml::core-engine-2023-03-20-3031::kkmecik
Liquibase command 'status' was executed successfully.
```

## Wynik Liquibase update

```
#####
##
| | () () |
| | - _ _ _ _ _ | | _ _ _ _ _
| | | | / _ | | | | | ' \ / _ | / _ \
| | _ | | (| | | | | |) | (| \ _ \ _ /
\ _ _ / _ \ _ , | \ _ , _ | _ . _ / \ _ , _ | _ / \ _ |
| |
| _ |
##
##
Get documentation at docs.liquibase.com
Get certified courses at learn.liquibase.com
##
#####

Starting Liquibase at 13:19:15 (version 4.23.1 #12042 built at 2023-08-10 13:48+0000)
Liquibase Version: 4.23.1
Liquibase Open Source 4.23.1 by Liquibase
Running Changeset: core-engine/tables.xml::core-engine-2023-03-23-0113::kkmiciek
Running Changeset: core-engine/tables.xml::core-engine-2023-03-23-0114::kkmiciek
Running Changeset: core-engine/tables.xml::core-engine-2023-03-20-3011::kkmiciek
Running Changeset: core-engine/tables.xml::core-engine-2023-03-20-3021::kkmiciek
Running Changeset: core-engine/tables.xml::core-engine-2023-03-20-541::kkmiciek
Running Changeset: core-engine/tables.xml::core-engine-2023-03-20-3031::kkmiciek

UPDATE SUMMARY

Run: 6
Previously run: 0
Filtered out: 0

Total change sets: 6

Liquibase: Update has been successful. Rows affected: 6
Liquibase command 'update' was executed successfully.
```

Dokumentacja liquibase: <https://docs.liquibase.com/change-types/home.html>