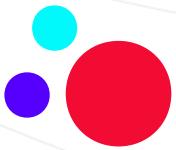




Jenkins CI/CD



HELLO

Wiktor Rosiński



<https://medium.com/@rosinski.wiktor>



<https://www.youtube.com/@wrosoftacademy>

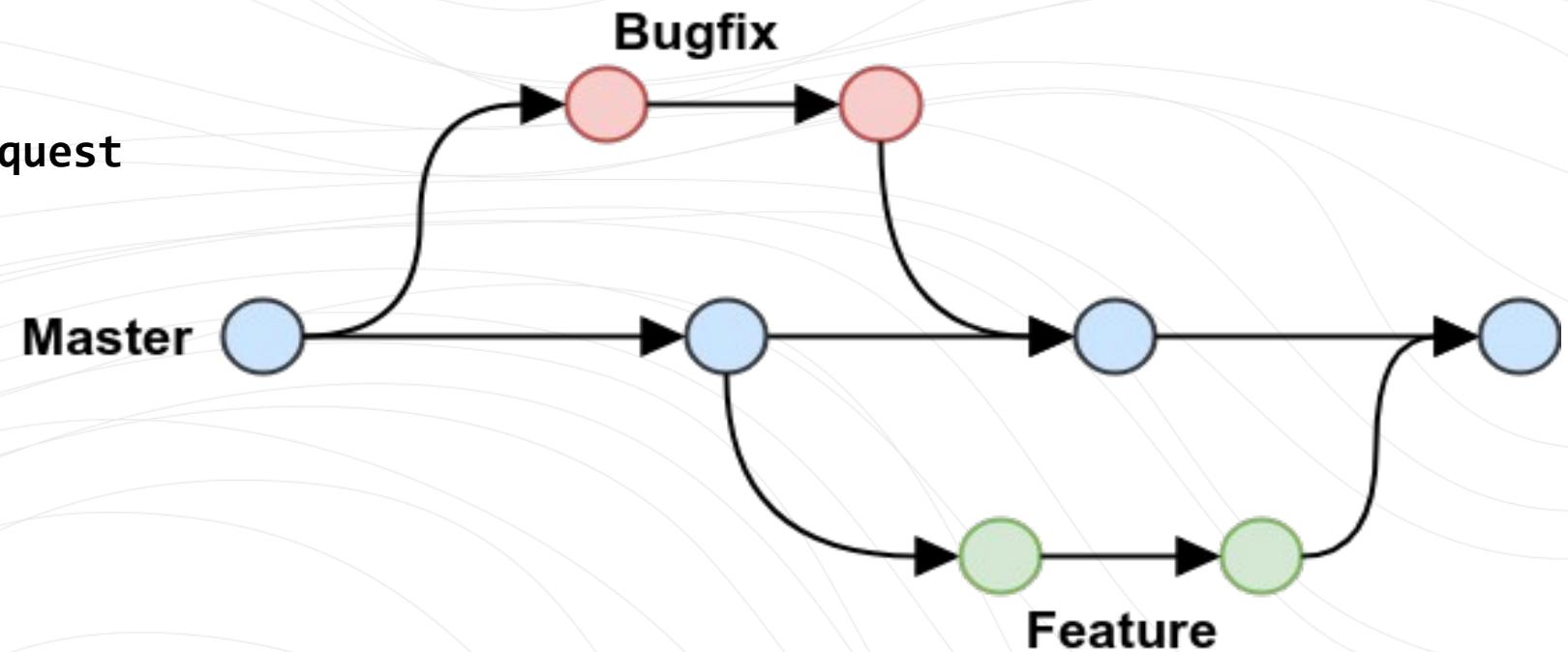


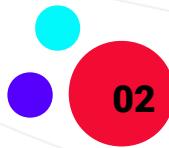
<https://www.linkedin.com/in/wiktor-rosinski/>



Flow pracy

1. Nowy task
2. Nowy branch (feature branch)
3. Implementacja
4. Merge Request / Pull Request
5. Code Review
6. Merge





02 CI & CD

Continuous Integration

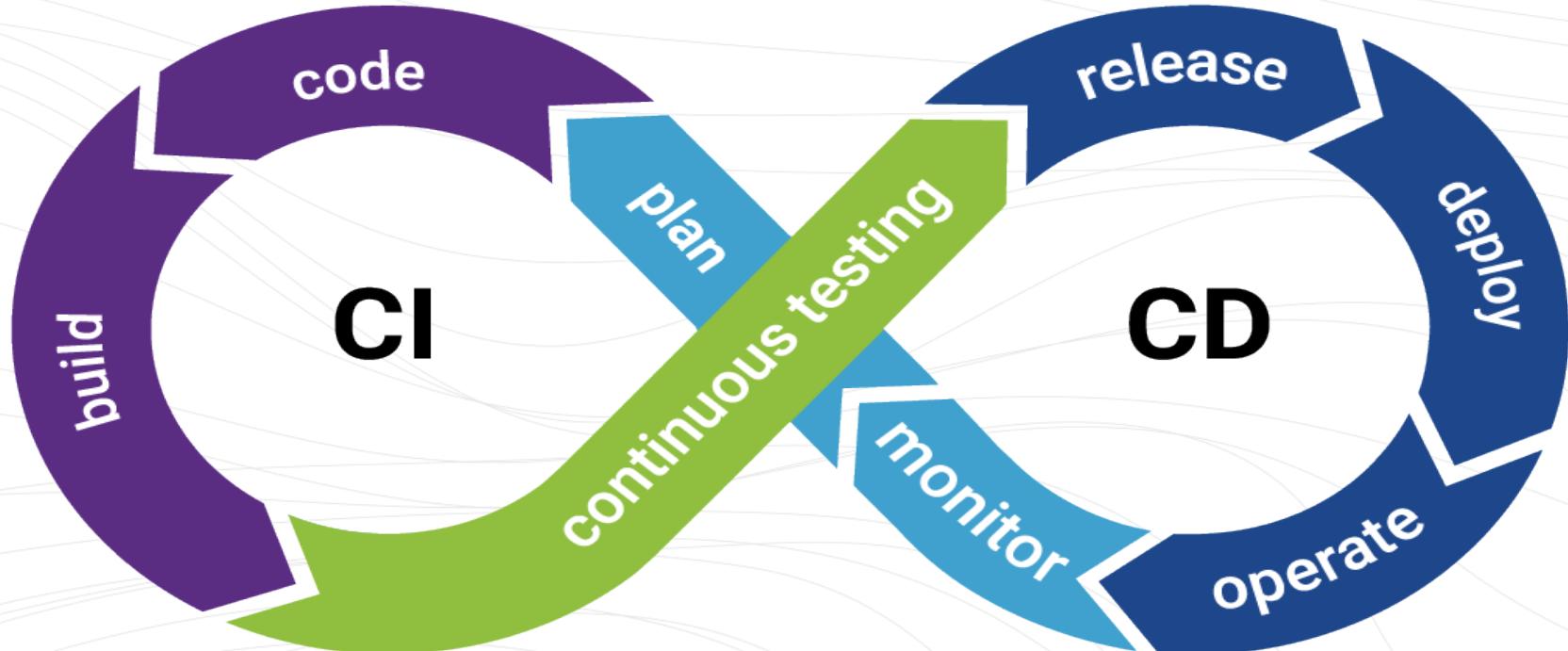
Zbiór praktyk oraz narzędzi umożliwiających regularne, powtarzalne wdrażanie zmian w kodzie do głównego repozytorium oraz ciągłym testowaniu zmian jakie wystąpiły w trakcie prac.

Continuous Delivery

Zbiór praktyk oraz narzędzi umożliwiających regularne, powtarzalne i zautomatyzowane wdrażanie zmian na serwery. Tworzenie nowej, działającej wersji programu. Niezawodne wydawanie oprogramowania.

Continuous Deployment

Pełna automatyzacja w dostarczaniu zmian na środowisko produkcyjne. Brak ingerencji człowieka. Proces sam przechodzi przez instalację na środowiskach testowych, uruchomienie testów a następnie instalacje na środowisku produkcyjnym.



- Oszczędność czasu !!!
- Łatwy rollback
- Zapewnia jakość produktu
- Mniej błędów na testach i produkcji *
- Nie trzeba ręcznie instalować na serwerach



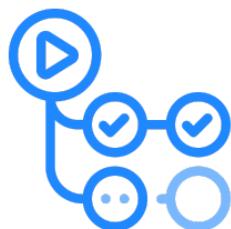
Jenkins



GitLab



Bamboo



GitHub Actions





Pobranie kodu
z
repozytorium



Zbudowanie
projektu



Testy
jednostkowe



Testy
integracyjne



Statyczna
analiza kodu



CI - efekt



Pobranie kodu
z
repozytorium



Zbudowanie
projektu



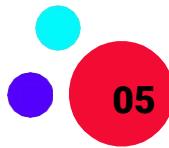
Wdrożenie na
środowisko
preprodukcyjne



Testy
akceptacyjne



Wdrożenie na
produkcję

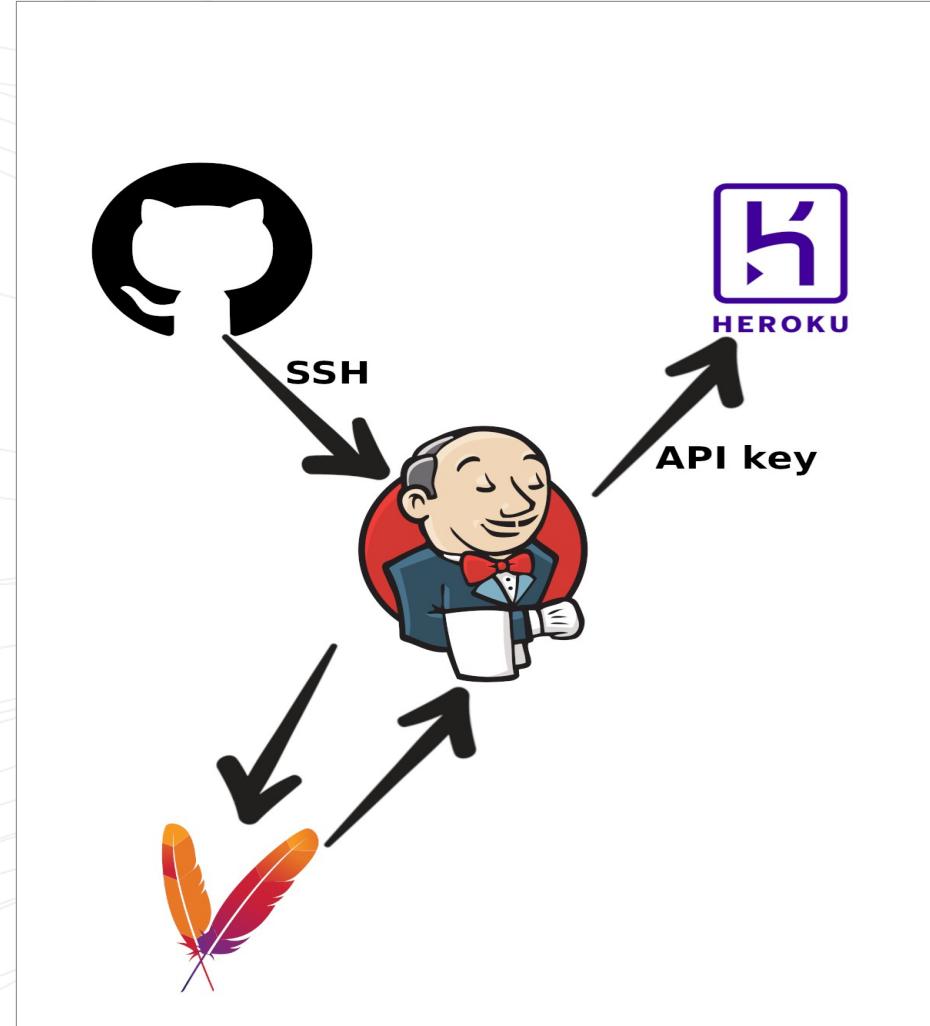


05 Jenkins

Zapraszamy Panie



Jenkins

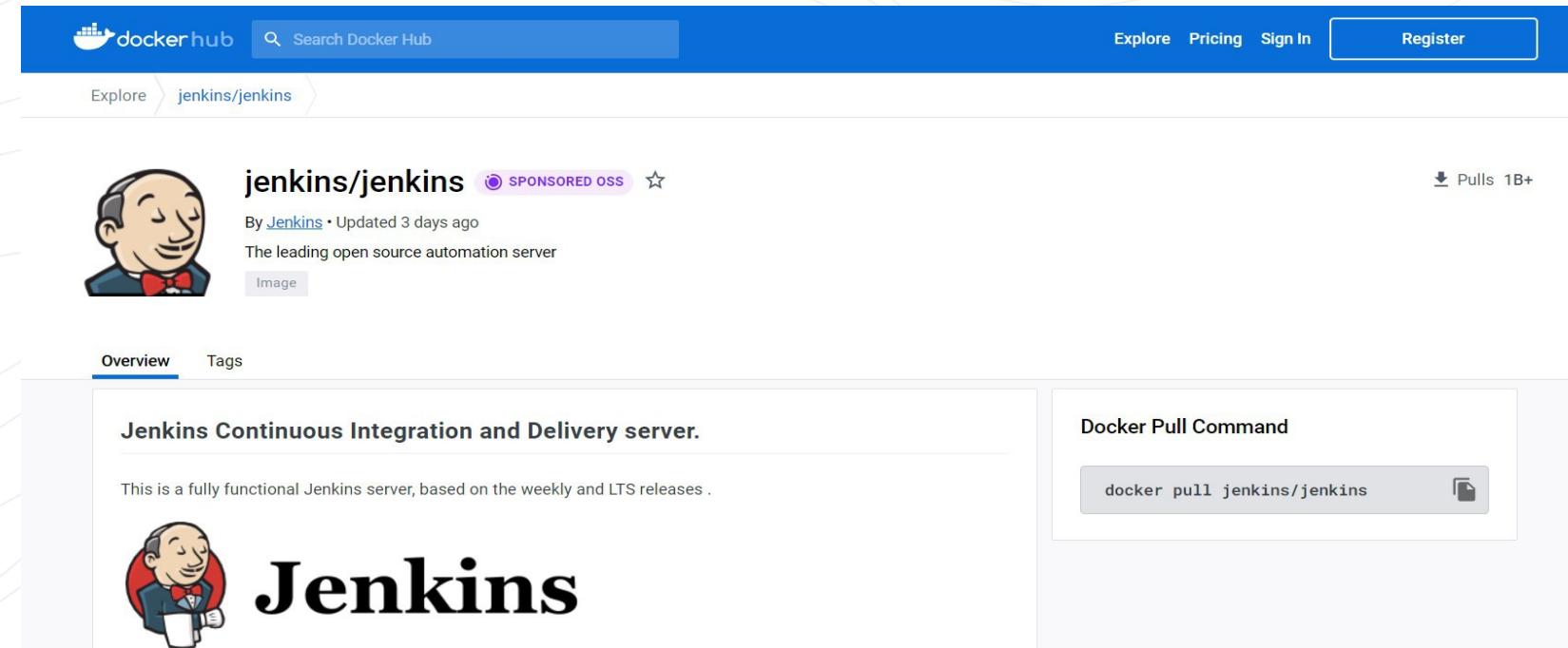


Wyszukujemy obraz Jenkinsa na DockerHub (https://hub.docker.com/_/jenkins/)

Pobieramy obraz Jenkinsa: docker pull jenkins/jenkins

Uruchamiamy kontener:

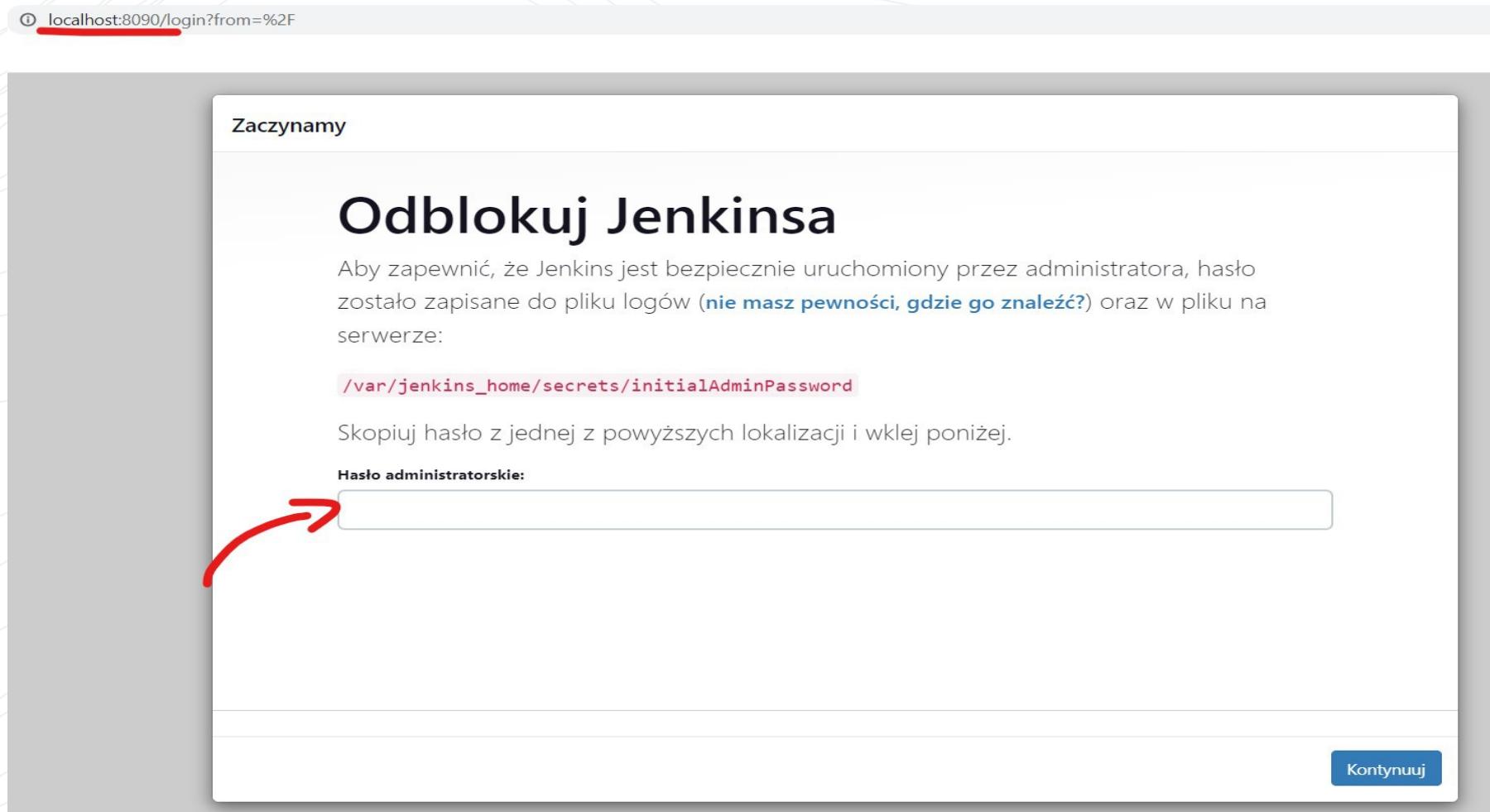
```
docker run --name jenkins -p8090:8080 -p50000:50000 -v /var/jenkins_home  
jenkins/jenkins
```



Kopiujemy klucz - potrzeby do pierwszego uruchomienia

```
MINGW64:/c/Users/admin
$ docker run --name jenkins -p8090:8080 -p50000:50000 -v /var/jenkins_home jenkins/jenkins
Running from: /usr/share/jenkins/jenkins.war
webroot: /var/jenkins_home/war
2023-06-16 19:01:00.082+0000 [id=1]     INFO  winstone.Logger#logInternal: Beginning extraction from war file
2023-06-16 19:01:04.640+0000 [id=1]     WARNING o.e.j.s.handler.ContextHandler#setContextPath: Empty contextPath
2023-06-16 19:01:08.897+0000 [id=1]     INFO  org.eclipse.jetty.server.Server#doStart: jetty-10.0.15; built: 2023-04-11T17:25:14.480Z; git: 68017dbd00236bb7e187330d7585a059610f661d; jvm 11.0.19+7
2023-06-16 19:01:05.742+0000 [id=1]     INFO  o.e.j.w.StandardDescriptorProcessor#visitServlet: NO JSP Support for /, did not find org.eclipse.jetty.jsp.JettyJspServlet
2023-06-16 19:01:05.945+0000 [id=1]     INFO  o.e.j.s.s.DefaultSessionIdManager#doStart: Session workerName=node0
2023-06-16 19:01:07.397+0000 [id=1]     INFO  hudson.WebAppMain#contextInitialized: Jenkins home directory: /var/jenkins_home found at: EnvVars.masterEnvVars.get("JENKINS_HOME")
2023-06-16 19:01:08.036+0000 [id=1]     INFO  o.e.j.s.handler.ContextHandler#doStart: Started w.44b3fe06e{Jenkins v2.410,/,file:///var/jenkins_home/war/,AVAILABLE}{/var/jenkins_home/war}
2023-06-16 19:01:08.076+0000 [id=1]     INFO  o.e.j.server.AbstractConnector#doStart: Started ServerConnector@180da663[HTTP/1.1, (http/1.1){0.0.0.0:8080}]
2023-06-16 19:01:08.144+0000 [id=1]     INFO  org.eclipse.jetty.server.Server#doStart: Started Server@3fb6cf60[STARTING][10.0.15,sto=0] @10844ms
2023-06-16 19:01:08.150+0000 [id=25]    INFO  winstone.Logger#logInternal: Winstone Servlet Engine running: controlPort=disabled
2023-06-16 19:01:08.715+0000 [id=32]    INFO  jenkins.InitReactorRunner$1#onAttained: Started initialization
2023-06-16 19:01:08.745+0000 [id=37]    INFO  jenkins.InitReactorRunner$1#onAttained: Listed all plugins
2023-06-16 19:01:10.501+0000 [id=39]    INFO  jenkins.InitReactorRunner$1#onAttained: Prepared all plugins
2023-06-16 19:01:10.511+0000 [id=33]    INFO  jenkins.InitReactorRunner$1#onAttained: Started all plugins
2023-06-16 19:01:10.527+0000 [id=38]    INFO  jenkins.InitReactorRunner$1#onAttained: Augmented all extensions
2023-06-16 19:01:10.820+0000 [id=31]    INFO  jenkins.InitReactorRunner$1#onAttained: System config loaded
2023-06-16 19:01:10.821+0000 [id=43]    INFO  jenkins.InitReactorRunner$1#onAttained: System config adapted
2023-06-16 19:01:10.823+0000 [id=32]    INFO  jenkins.InitReactorRunner$1#onAttained: Loaded all jobs
2023-06-16 19:01:10.826+0000 [id=30]    INFO  jenkins.InitReactorRunner$1#onAttained: Configuration for all jobs updated
2023-06-16 19:01:10.865+0000 [id=58]    INFO  hudson.util.Retrier#start: Attempt #1 to do the action check updates server
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.codehaus.groovy.vmplugin.v7.Java7$1 (file:/var/jenkins_home/war/WEB-INF/lib/groovy-all-2.4.21.jar) to constructor java.lang.invoke.MethodHandles$Lookup(java.lang.Class, int)
WARNING: Please consider reporting this to the maintainers of org.codehaus.groovy.vmplugin.v7.Java7$1
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
2023-06-16 19:01:11.690+0000 [id=33]    INFO  jenkins.install.SetupWizard#init
*****
*****
*****
Jenkins initial setup is required. An admin user has been created and a password generated.
Please use the following password to proceed to installation:
6e8e6f447aa84a15af9ac7708bf11db8
This may be copied to clipboard: /jenkins_home/secrets/initialAdminPassword
*****
*****
*****
2023-06-16 19:01:43.269+0000 [id=33]    INFO  jenkins.InitReactorRunner$1#onAttained: Completed initialization
2023-06-16 19:01:43.291+0000 [id=24]    INFO  hudson.lifecycle.Lifecycle#onReady: Jenkins is fully up and running
2023-06-16 19:01:44.078+0000 [id=58]    INFO  h.m.DownloadService$Downloadable#load: Obtained the updated data file for hudson.tasks.Maven.MavenInstaller
2023-06-16 19:01:44.079+0000 [id=58]    INFO  hudson.util.Retrier#start: Performed the action check updates server successfully at the attempt #1
```

- Wchodzimy do Jenkinsa. W przeglądarce pod url: `localhost:8090`
- Wklejamy skopiowany klucz



- *Instalujemy sugerowane wtyczki*

Dostosuj Jenkinsa

Wtyczki rozszerzające Jenkinsa o dodatkowe funkcjonalności, które zaspokajają wiele potrzeb.

Zainstaluj sugerowane wtyczki

Zainstaluj wtyczki, które społeczność Jenkinsa uznała za najbardziej przydatne.

Wybierz wtyczki do instalacji

Wybierz i zainstaluj wtyczki najbardziej dopasowane do Twoich potrzeb.

Instalacja...

Instalacja wtyczek

✓ Folders	OWASP Markup Formatter	Build Timeout	Credentials Binding
Timestamper	Workspace Cleanup	Ant	Gradle
Pipeline	GitHub Branch Source	Pipeline: GitHub Groovy Libraries	Pipeline: Stage View
Git	SSH Build Agents	Matrix Authorization Strategy	PAM Authentication
LDAP	Email Extension	Mailer	

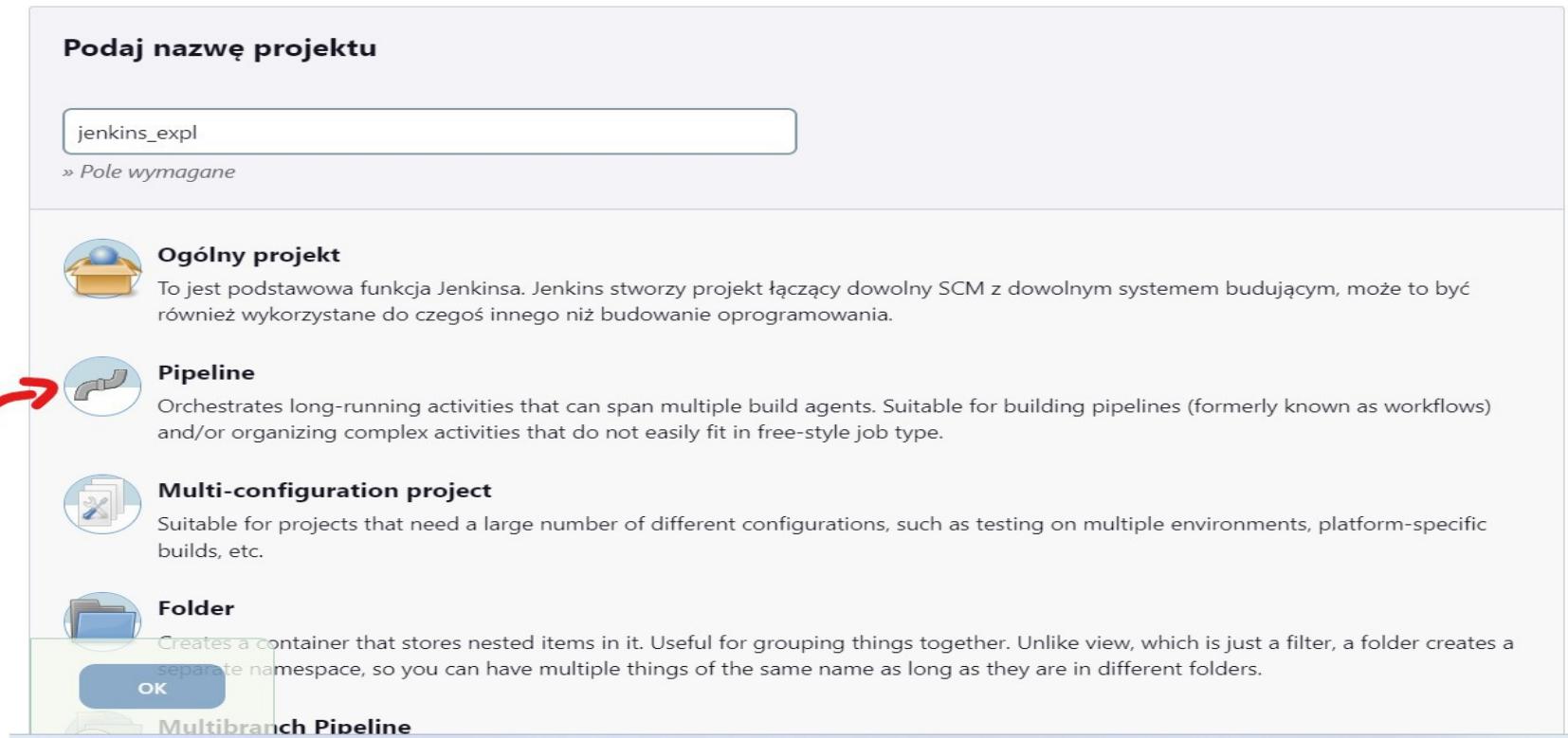
** Ionicons API
Folders
** bouncycastle API
** Instance Identity
** JavaBeans Activation Framework (JAF) API
** JavaMail API

** - zależności wymagane

Jenkins 2.410

The screenshot shows the Jenkins dashboard. At the top, there is a navigation bar with the Jenkins logo, a search bar containing "Szukaj... (⌘+K)", and user information for "admin". Below the navigation bar, the main content area has a title "Witamy w Jenkinsie!" and a subtitle "This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project." There are several call-to-action buttons: "Create a job", "Set up a distributed build", "Set up an agent", "Configure a cloud", and "Learn more about distributed builds". On the left side, there is a sidebar with links for "Nowy Projekt", "Użytkownicy", "Historia zadań", "Zarządzaj Jenkinsem", and "Moje widoki". Under "Zarządzaj Jenkinsem", there are dropdown menus for "Kolejka zadań" (showing "Nie ma zadań w kolejce") and "Status wykonawców zadań" (showing "1 Bezczytny" and "2 Bezczytny"). At the bottom right, there are links for "REST API" and "Jenkins 2.375.3".

- **Przycisk Nowy Projekt**
- **Nazwa projektu**
- **Wybieramy Pipeline**
- **Klikamy [OK]**





05

Jenkins – nowy projekt

- W tym miejscu będziemy konfigurować pipeline. Dla ułatwienia możemy przełączyć się na Definition - Pipeline script i wybrać sample GitHub + maven, a następnie skopiować do notatnika. Przyda się przy pisaniu swojego

Pipeline

Definition

Pipeline script

Script

```
1 pipeline {  
2   agent any  
3  
4   tools {  
5     // Install the Maven version configured as "M3" and add it to the path.  
6     maven "M3"  
7   }  
8  
9   stages {  
10    stage('Build') {  
11      steps {  
12        // Get some code from a GitHub repository  
13        git 'https://github.com/jglick/simple-maven-project-with-tests.git'  
14  
15        // Run Maven on a Unix agent.  
16        sh "mvn -Dmaven.test.failure.ignore=true clean package"  
17    }
```

GitHub + Maven

Ctrl + C

 Use Groovy Sandbox ?

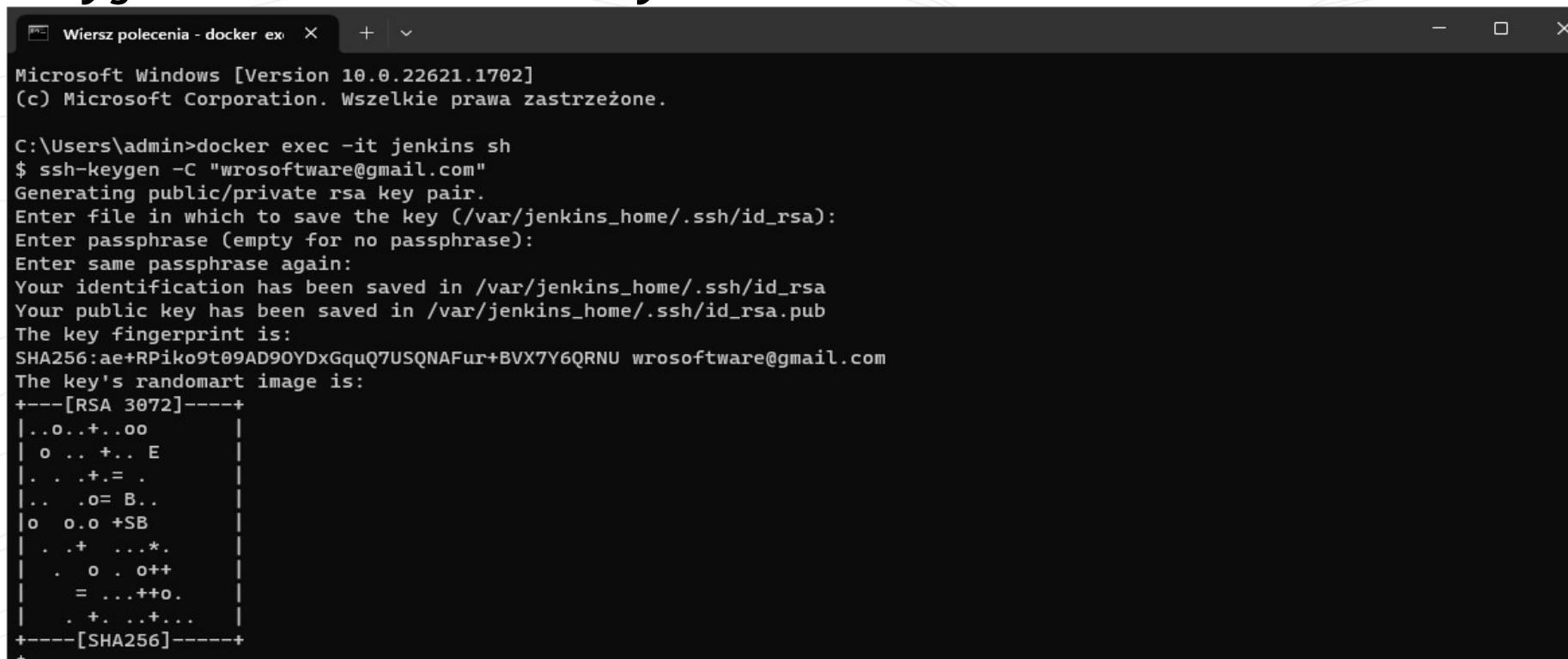
Pipeline Syntax

Zapisz

Zastosuj

Do komunikacji po SSH potrzebna będzie para kluczy. Musimy je założyć wewnątrz kontenera Jenkinsa.

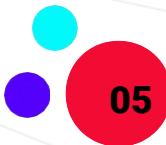
- Wchodzimy do środka kontenera: docker exec -it jenkins sh
- Tworzymy parę kluczy za pomocą ssh-keygen (taka aplikacja): ssh-keygen -t ed25519 -C "\${moj_mail@mail.com}"
- Dalej dajemy same entery (domyślna nazwa pliku, nie chcemy passphrase)
- Po wygenerowaniu dostaniemy obrazek



The screenshot shows a terminal window titled "Wiersz polecenia - docker exec". The window displays the following command and its output:

```
Microsoft Windows [Version 10.0.22621.1702]
(c) Microsoft Corporation. Wszelkie prawa zastrzeżone.

C:\Users\admin>docker exec -it jenkins sh
$ ssh-keygen -C "wrosoftw@r@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/var/jenkins_home/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /var/jenkins_home/.ssh/id_rsa
Your public key has been saved in /var/jenkins_home/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:ae+RPiko9t09AD90YDxGquQ7USQNAFur+BVX7Y6QRNU wrosoftw@r@gmail.com
The key's randomart image is:
+---[RSA 3072]---+
| ..o..+..oo
| o .. +.. E
| . . .+= .
| .. .o= B..
| o o.o +SB
| . .+ ...*.
| . o . o++
| = ...++o.
| . +. .+...
+---[SHA256]---+
$
```



05

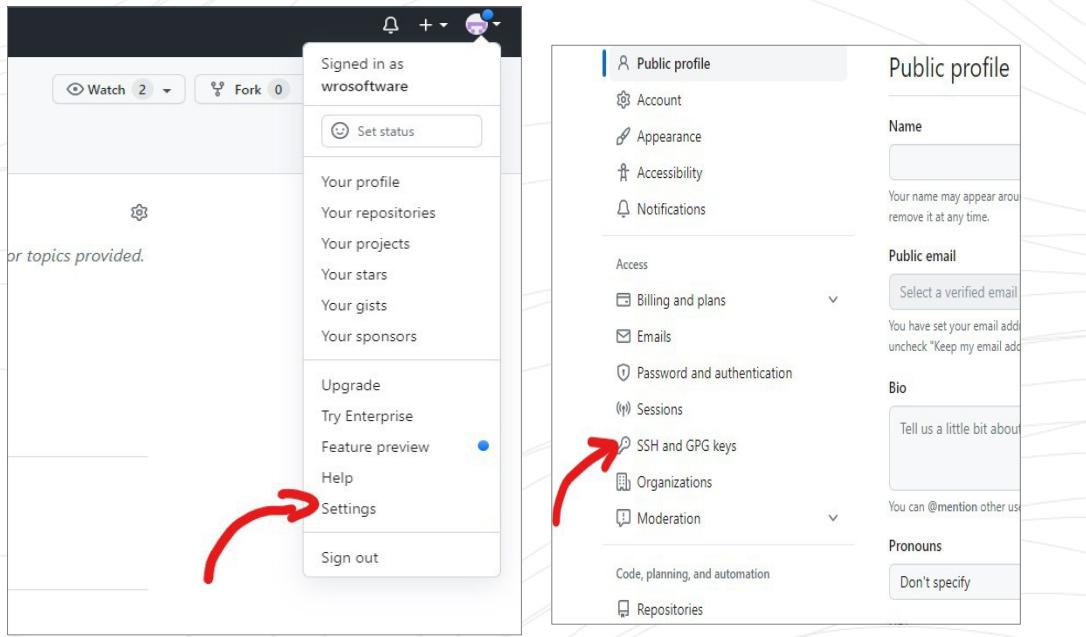
Integracja Git - Jenkins

Klucz publiczny idzie do git'a, ponieważ on będzie tylko odczytywać komendy.

- Wykonujemy komende: `cat ${sciezka_do_klucza_publicznego}`
- Kopiujemy klucz publiczny

```
Generating public/private rsa key pair.
Enter file in which to save the key (/var/jenkins_home/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /var/jenkins_home/.ssh/id_rsa
Your public key has been saved in /var/jenkins_home/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:ae+RPiko9t09AD9OYDxGquQ7USQNAFur+BVX7Y6QRNU wrosoftware@gmail.com
The key's randomart image is:
+---[RSA 3072]---+
| ..o..+..oo |
| o .. +.. E |
| . . .+= . |
| .. .o= B.. |
| o o.o +SB |
| . .+ ...*. |
| . o . o++ |
| = ...++o. |
| . +. .+. . |
+---[SHA256]---+
$ cat /var/jenkins_home/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQgQDGhSO/PjyLB3AxRQ5It0Qccp9y76AfxgEeNm+pM2oVo6g6ehme4Vbe7/6QsedaIwd8lcMRE0Wys2krsrtaqijx27D
AlcxpSBc4kBbZKKL/Bh9auSsdadMBA0PIC55p4fyrvfOsWiZENIS62E6dWWjhZnw4e01ELxSbg5IzD9t5aCKgSq3zGT3JumD8lBzoKmpFcf0HyG1xPZrmON/xkVtB
tFiRogsGPGg6eUGLUeDr98JpWZpAOgDzXFFRab5Tuv6kxM/+0AVgcv7EryzS1lG5FB3abwUMwDS4fFMmoiLrK8D+mp2JTPuyXbriybJeskAm8t6q1WeNX/Mj5o3V6S
rB8o2MLEWXjqqjlvrRJ9Tdh+xbpvcKPhTbw0RgOVHjI+JzOKwYrc7vBmnctFCZWieptTzauG+A1gg/rhF4giT9yJWpxI7T13zWKb4kRcQc2dsSemXkGoT455S14lzbxH
U09BwrMzx2s2RNObU4eHjkhyRQoS95jGoGyRwRQJoA8= wrosoftware@gmail.com
$
```

- Przechodzimy do ustawień profilu
- Wybieramy SSH and GPG keys
- Klikamy w przycisk [New ssh key]
- Wklejamy klucz publiczny, title - dowolna nazwa



SSH keys / Add new

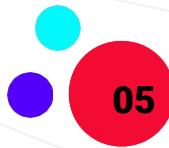
Title: jenkins-demo

Key type: Authentication Key

Key:

```
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQgQDGHSO/PjLB3AxRQ5It0Qccp9y76AfxgEeNm+pm2oVo6g6ehme4Vbe7/6QsedaIwd8lcMRE0Wys2krsrtaqjjx27DA1cxpSBc4kBbZKKL/Bh9auSsdadMBA0PIC55p4fyrvefxOsWiZENIS62E6dWWjhZnw4e01ELxSbg5IzD9t5aCKgSq3zGT3JumD8IBzoKmpFcfOHyG1xPZrmON/xkVtBtFiRogsPGGg6eUGIueDr98JpWZpAOgDzXFFRab5Tu v6kxM+/OAvgcv7EryzS1IG5FB3abwUMwDS4fFMmoiLrK8D+mp2JTPuyXbriybJeskAm8t6q1WeNX/Mj5o3V6SrB8o2MLE WXjqqjlvrRJ9Tdh+xbpvckPhTbw0RgOVHjl+JzOKwYrc7vBmnctFCZWiepTzauG+A1gg/rhF4giT9yJWpxl7T13zWKb4kRcQc2 dsSemXkGoT455S14lzbxHU09BwrMzx2s2RNObU4eHjkhyRQoS95jGoGyRwRQJoA8= wrosoftware@gmail.com
```

Add SSH key



05

Integracja Git - Jenkins

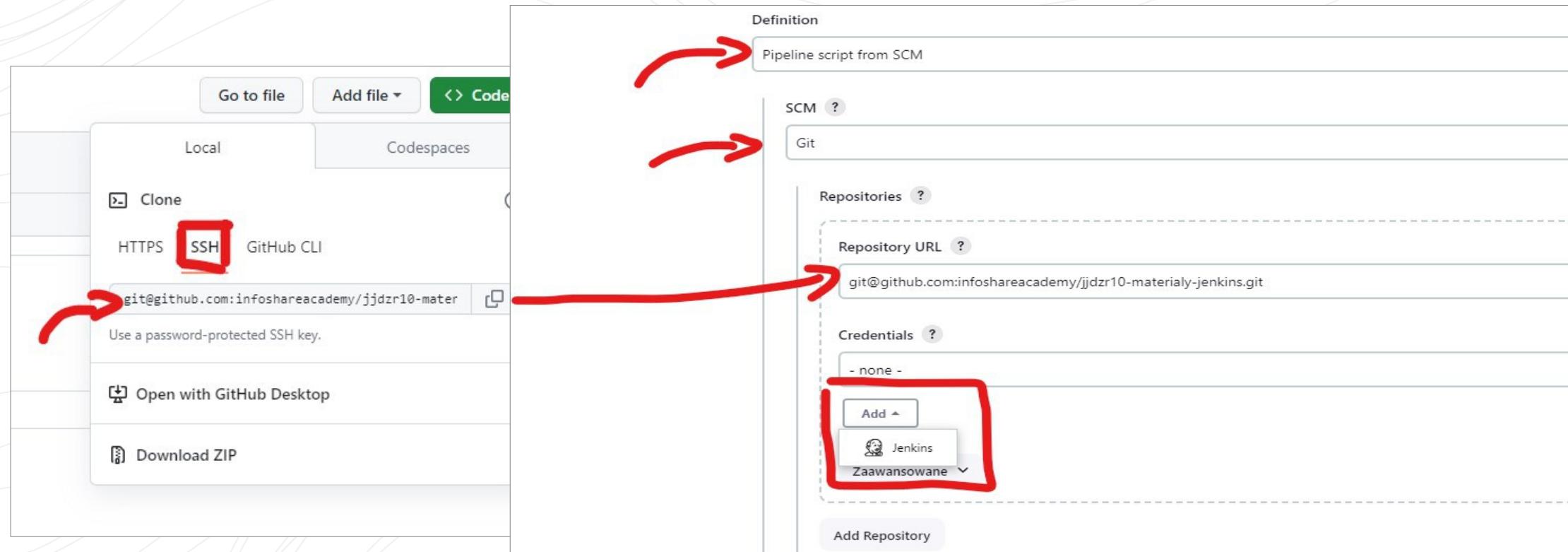
Przechodzimy do ustawień profilu

Wybieramy SSH and GPG keys

Klikamy w przycisk [New ssh key]

Wklejamy klucz publiczny, title - dowolna nazwa

- Wybieramy pipeline definition jako Pipeline script from SCM
- SCM – wybieramy Git
- Repository url wklejamy link ssh
- Credentials- będziemy dodawać nowe, bo jeszcze nie mamy



- **Kind:** ssh username with private key
- **Scope:** zostaje jak jest
- **ID:** unikalne, nazwa dowolna, co by wiedzieć czym jest
- **Description:** pomaga później w wyborze
- **Username:** nazwa profilu na github

Jenkins Credentials Provider: Jenkins

Add Credentials

Domain
Global credentials (unrestricted)

Kind
SSH Username with private key

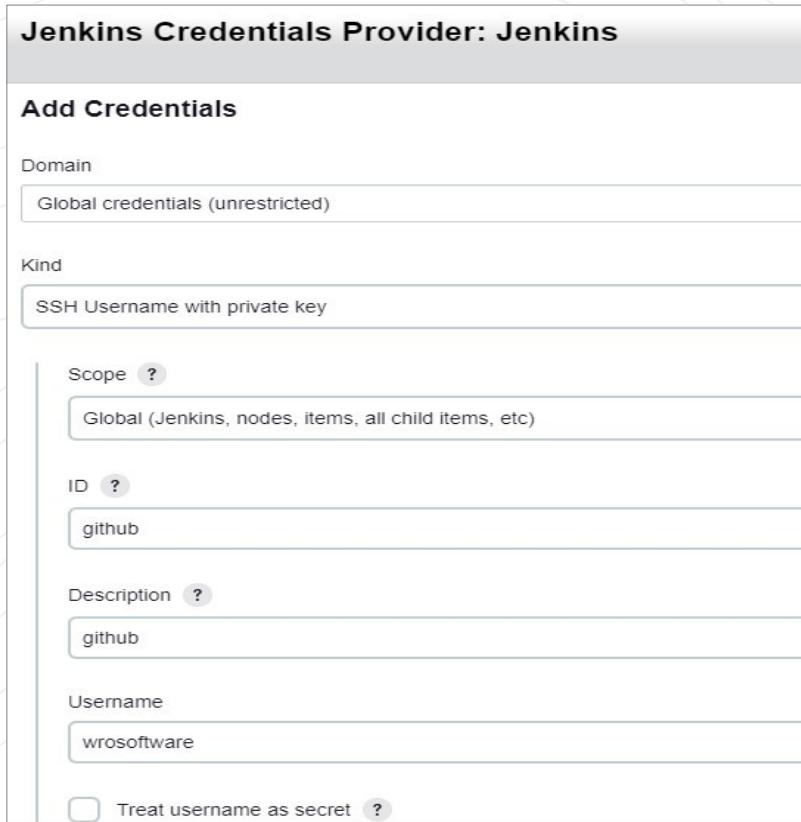
Scope ?
Global (Jenkins, nodes, items, all child items, etc)

ID ?
github

Description ?
github

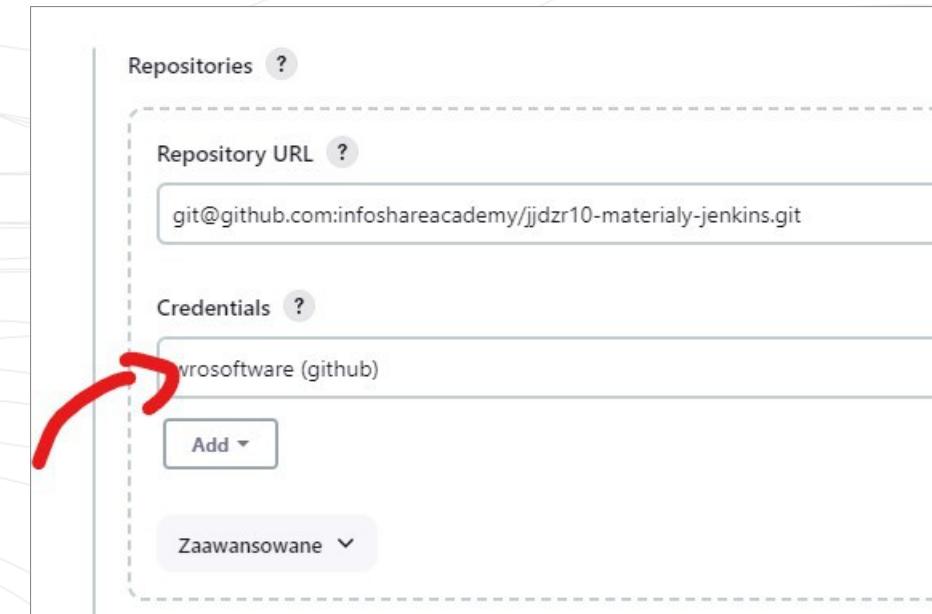
Username
wrosoftware

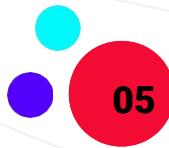
Treat username as secret ?



- Zaznaczmy Enter directly
- Klikamy w przycisk Add po prawej
- Kopujemy klucz (cały!!!) z konsoli, analogicznie jak publiczny
- Wklejamy w Jenkinsie
- Klikamy w przycisk Add i teraz możemy wybrać z Credentials nasz nowy klucz

```
Enter same passphrase again:  
Your identification has been saved in /var/jenkins_home/.ssh/id_rsa.  
Your public key has been saved in /var/jenkins_home/.ssh/id_rsa.pub  
The key fingerprint is:  
SHA256:ae+RPiko9t09AD90YDxGquQ7USQNAFur+BVX7Y6QRNU wrosoftware@gmail.com  
The key's randomart image is:  
+---[RSA 3072]---+  
|...o...+..oo  
|o ... +... E  
|.. .+= .  
|.. .o= B..  
|o o.o +SB  
| ..+ ...*.  
| . o . o++  
|= ...++o.  
| .+ ...+...  
+---[SHA256]---+  
  
$ cat /var/jenkins_home/.ssh/id_rsa.pub  
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAgQDGhSO/PjyLB3AxRQ5It0Qccp9y76LfxgEeNm-  
qijx27DA1cxpSBc4kBbZKKL/Bh9auSsdadMBA0PIC55p4fyrvefxOsWiZENIS62EcdWWjhZnw4e  
1xPZrmON/xkVtBtFiRogsGPGg6eUGlUeDr98JpWZpAOgDzXFFRab5Tuv6kxM/+OAVgcv7Eryz9  
kAm8t6q1WeNX/Mj5o3V6SrB8o2MLEWXjqqjlvrRJ9Tdh+xbpvcKPhTbw0D.gOVhjI+JzOKwYrc7v  
4kRcQc2dsSemXkGoT455S14lzbxHU09BwrMzx2s2RNObU4eHikhyPQ895jGoGyRwRQJoA8= wj  
$ cat /var/jenkins_home/.ssh/id_rsa  
-----BEGIN OPENSSH PRIVATE KEY-----  
b3BlbnNzaC1rZXktdjEAAAAABG5vbmuAAAAEb9uZQAAAAAAAAABAABlwAAAAdzc2gtcn  
NhAAAAAwEAAQAAAYEAxoUjvz48iwdwMUUOSLdEHHKfcu+gh8YBHjZvqZtqFaOoOnoZnuFW  
3u/+kLHnWiMHFJXDERNFsrNpK7K7Wqoo8duuwNXMaUgXOJAW2Sii/wYfWrkrHWnTAQNDyA  
ueaeH8q73n8TrFomRDSEuthOnVlo4WZ80HtNRC8Um40SMw/beWgioEqt8xk9ybpg/JQc6C  
pqRXHzh8htcT2a5jjf8ZFbQbRYkaILBjxoOnlBpVHg6/fCaVmaQDoA81xRUWm+U7r+pMTP
```





05 Jenkins integracja z Gitem

I jak zwykle nic nie działa...

Repositories ?

Repository URL ?

git@github.com:infoshareacademy/jjdr10-materialy-jenkins.git

!

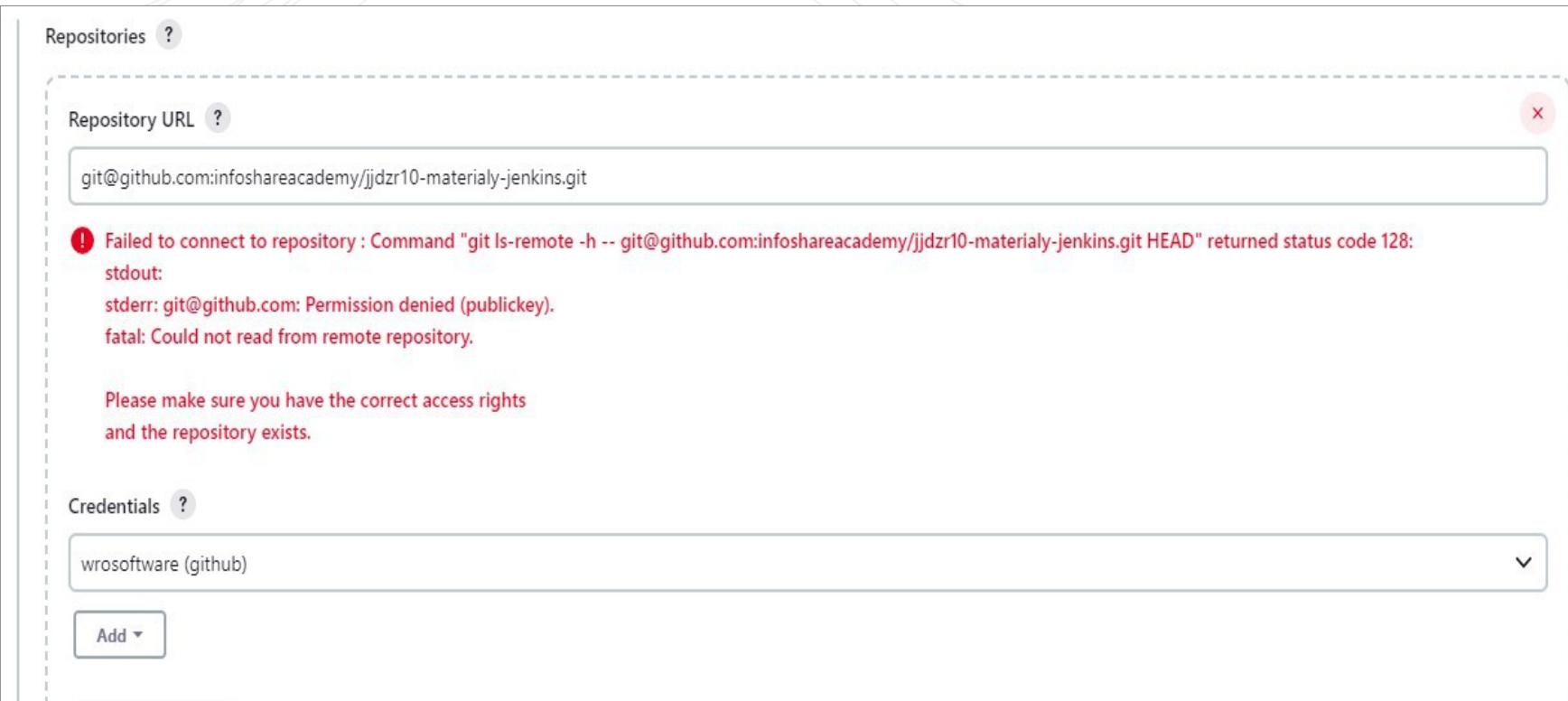
Failed to connect to repository : Command "git ls-remote -h -- git@github.com:infoshareacademy/jjdr10-materialy-jenkins.git HEAD" returned status code 128:
stdout:
stderr: git@github.com: Permission denied (publickey).
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.

Credentials ?

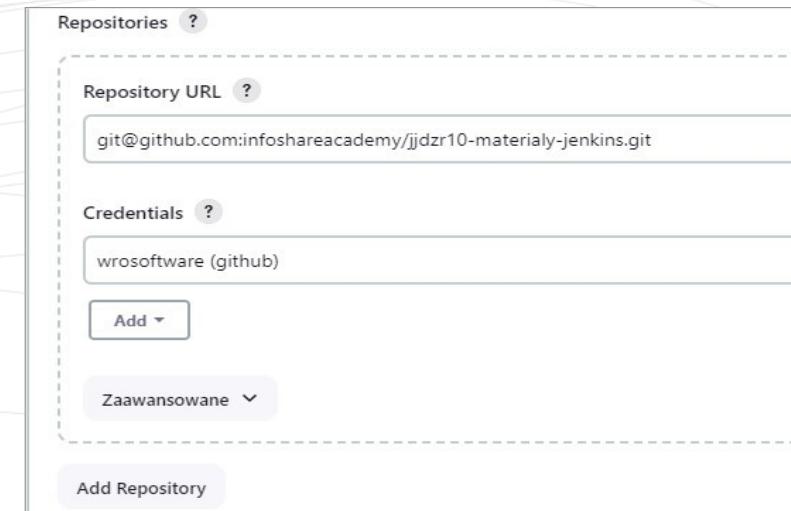
wrosoftwae (github)

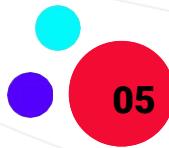
Add ▾



- Kopiujemy komendę która była podana na poprzednim
- Wracamy do konsoli i ją wywołujemy
- Zapyta się czy na pewno wiemy co robimy... więc skłamiemy ;)
- Teraz już powinno być ok na Jenkinsie

```
The authenticity of host 'github.com (140.82.121.4)' can't be established.  
ECDSA key fingerprint is SHA256:p2QAMXNIC1TJYWeI0ttrVc98/R1BUFWu3/LiyKgUFQM.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes ↗  
Warning: Permanently added 'github.com,140.82.121.4' (ECDSA) to the list of known hosts.  
$ git ls-remote -h -- git@github.com:wrosoftware/jenkins_expl.git HEAD  
Warning: Permanently added the ECDSA host key for IP address '::ffff:140.82.121.3' to the list of known hosts.  
$
```

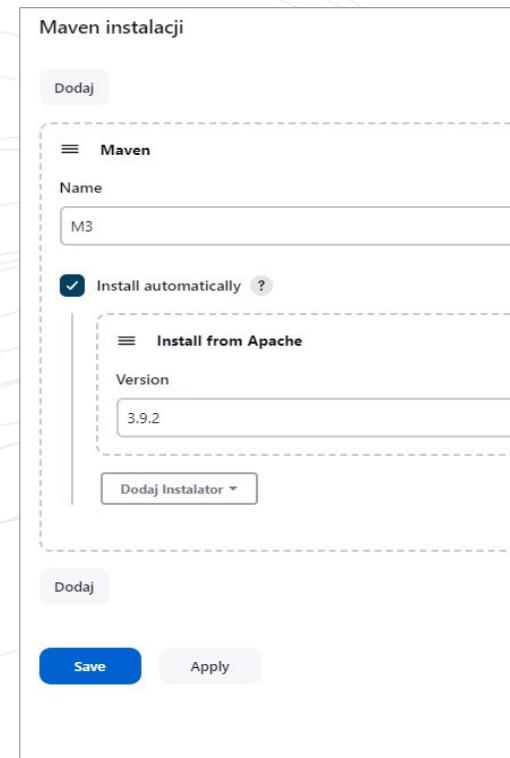
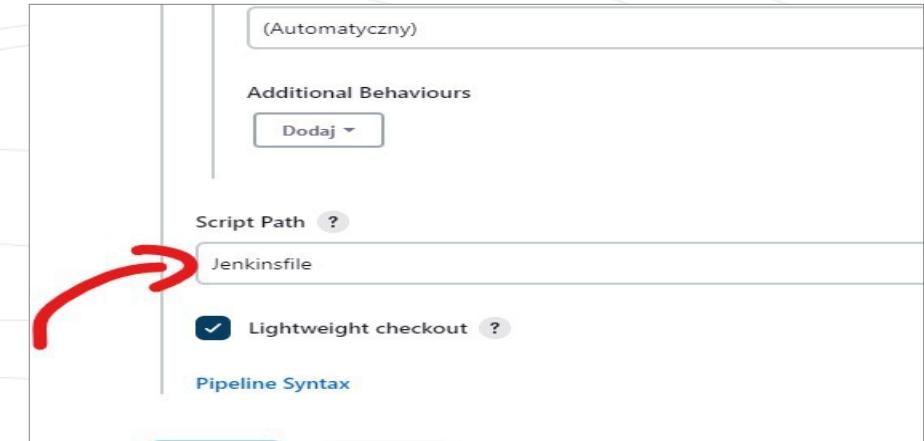




05

Jenkinsfile

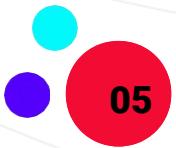
- Nazwa pliku (ściezka) zdefiniowana w path
- Tworzymy plik Jenkinsfile w projekcie
- Kopiujemy skrypt z notatnika (ten przykładowy)
- W skrypcie jest zdefiniowany tool maven jako M3
- Musimy to skonfigurować w Jenkins
- Przechodzimy po ścieżce Tablica / Zarządzaj Jenkinsem / Tools
- Szukamy sekcji Maven
- Dodajemy i nazywamy M3
- Zapisujemy zmiany



Modyfikujemy nasz skrypt w Jenkinsfile, następnie możemy uruchomić pipeline w Jenkinsie

```
pipeline {  
    agent any  
  
    tools {  
        maven "M3"  
    }  
  
    stages {  
        stage('Build') {  
            steps {  
                sh "mvn clean compile"  
            }  
        }  
        stage('Test') {  
            steps {  
                sh "mvn test"  
            }  
        }  
    }  
}
```

- *Stage - etap*
- *Step - pojedynczy krok w etapie, najczęściej jedna komenda*



05

Integracja Heroku - Jenkins

W Heroku przechodzimy do account settings

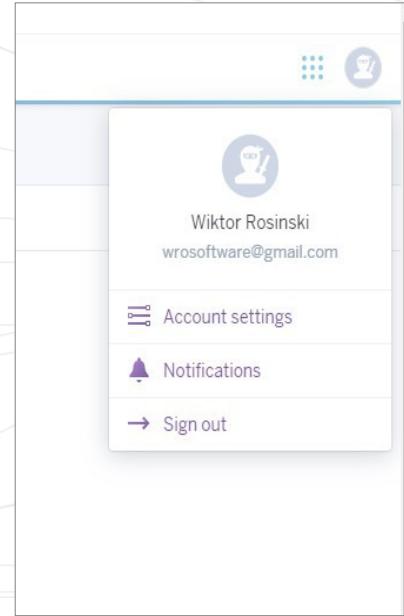
Generujemy API key, chyba, że już mamy

Kopiujemy API key i lecimy do Jenkinsa

Idziemy do ścieżki Tablica/Zarządzanie Jenkinsem / System

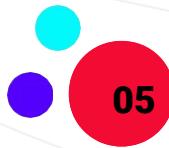
Szukamy checkbox zmienne środowiskowe - zaznaczamy

Dodajemy klucz: HEROKU_API_KEY, wartość wklejamy z Heroku



The screenshot shows the Jenkins Global properties configuration screen. At the top, there are two unchecked checkboxes: 'Disable deferred wipeout on this node' and 'Lokalizacja narzędzi'. Below them is a checked checkbox labeled 'Zmienne środowiskowe'. Underneath this, there is a table titled 'Lista par klucz-wartość' with one row visible. The row has columns 'nazwa' (containing 'HEROKU_API_KEY') and 'wartość'. A red arrow points from the left towards the 'HEROKU_API_KEY' cell in the table. At the bottom of the table is a 'Dodaj' button.

nazwa	wartość
HEROKU_API_KEY	



05

Deploy Heroku

- Logujemy się do Heroku
- Zakładamy nowy projekt
- Przechodzimy do settings i szukamy Config vars
- Dodajemy zmienną KEY: PORT, VALUE: 8080 (będziemy nasłuchiwać tego portu)

The screenshot shows the Heroku Settings page for the app "jenkins-test". The top navigation bar includes tabs for Deploy, Metrics, Activity, Access, and Settings. The Settings tab is active. The main configuration details are listed below:

- App Name: jenkins-test
- Region: Europe
- Stack: heroku-22
- Framework: heroku-maven-plugin
- Heroku git URL: <https://git.heroku.com/jenkins-test.git>

In the "Config Vars" section, there is a table with one row:

KEY	VALUE
PORT	8080

A "Hide Config Vars" button is located in the top right corner of the "Config Vars" section.

- W projekcie w pom.xml dodajemy plugin heroku
- Dodajemy do Jenkinsfile step z komendą: sh "mvn clean heroku:deploy"

```
<plugin>
  <groupId>com.heroku.sdk</groupId>
  <artifactId>heroku-maven-plugin</artifactId>
  <version>3.0.7</version>
  <configuration>
    <appName>Nazwa_projektu_z_heroku</appName>
    <processTypes>
      <web>java -jar -Dserver.port=$PORT target/${project.build.finalName}.jar</web>
    </processTypes>
  </configuration>
</plugin>
```

```
stages {
  stage('Build') {
    steps {
      sh "mvn clean compile"
    }
  }
  stage('Test') {
    steps {
      sh "mvn test"
    }
  }
  stage('Deploy') {
    steps {
      sh "mvn clean heroku:deploy"
    }
  }
}
```

- Uruchamiamy pipeline
- Klikamy w job
- Przechodzimy do Logów

Jenkins

Tablica > jenkins_expl >

Status

Changes

Uruchom →

Konfiguruj

Usuń Pipeline

Full Stage View

Zmień nazwę

Pipeline Syntax

Historia zadań

trend

Full Stage View

Zmień nazwę

Pipeline Syntax

Historia zadań

trend

szukaj

#	Datum
#21	17 cze 2023, 02:43
#20	17 cze 2023, 02:38
#19	17 cze 2023, 02:33
#18	17 cze 2023, 02:22

Build #21

Status

Changes

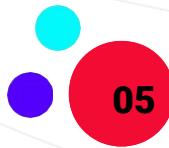
Console Output →

Edit Build Information

Usuń zadanie '#21'

Wystartowane

git Revision: 5cf4c

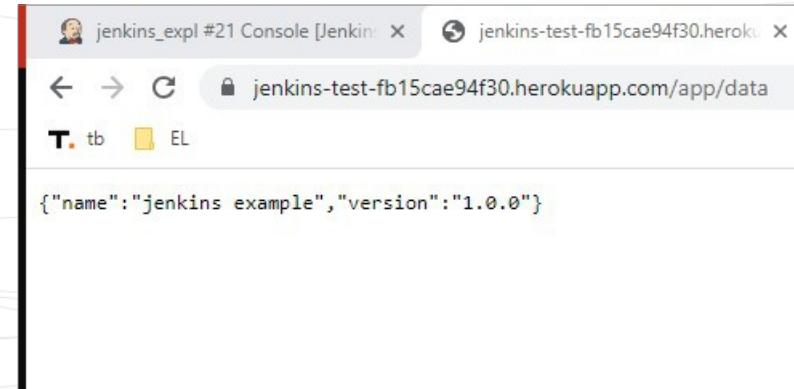


05

Uruchomienie CD

W Logach scrolujemy na sam dół. Tam będzie link do naszej aplikacji:

```
[INFO] remote: -----> heroku-maven-plugin app detected
[INFO] remote: -----> Installing OpenJDK 17... done
[INFO] remote: -----> Discovering process types
[INFO] remote:       Procfile declares types -> web
[INFO] remote:
[INFO] remote: -----> Compressing...
[INFO] remote:       Done: 95.5M
[INFO] remote: -----> Launching...
[INFO] remote:       Released v8
[INFO] remote:       https://jenkins-test-fb15cae94f30.herokuapp.com/ deployed to Heroku
[INFO] remote:
[INFO] -----> Done
[INFO]
[INFO] BUILD SUCCESS
[INFO]
```



Jest to miejsce gdzie trzymamy nasze biblioteki





Pobranie kodu
z
repozytorium



Zbudowanie
projektu



Publikacja do
artifact
repository

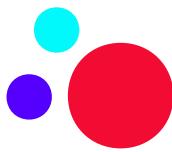
A gdzie trzymamy obrazy dockera?





ADDITIONAL MATERIALS





- ✓ <https://www.jenkins.io/doc/book/pipeline/getting-started/>
- ✓ <https://hub.docker.com/r/jenkins/jenkins>
- ✓ <https://www.baeldung.com/spring-boot-ci-cd>
- ✓ <https://medium.com/@prashantsde/deploy-java-application-with-complete-ci-cd-pipeline-1f3a2a2a2a2a>

