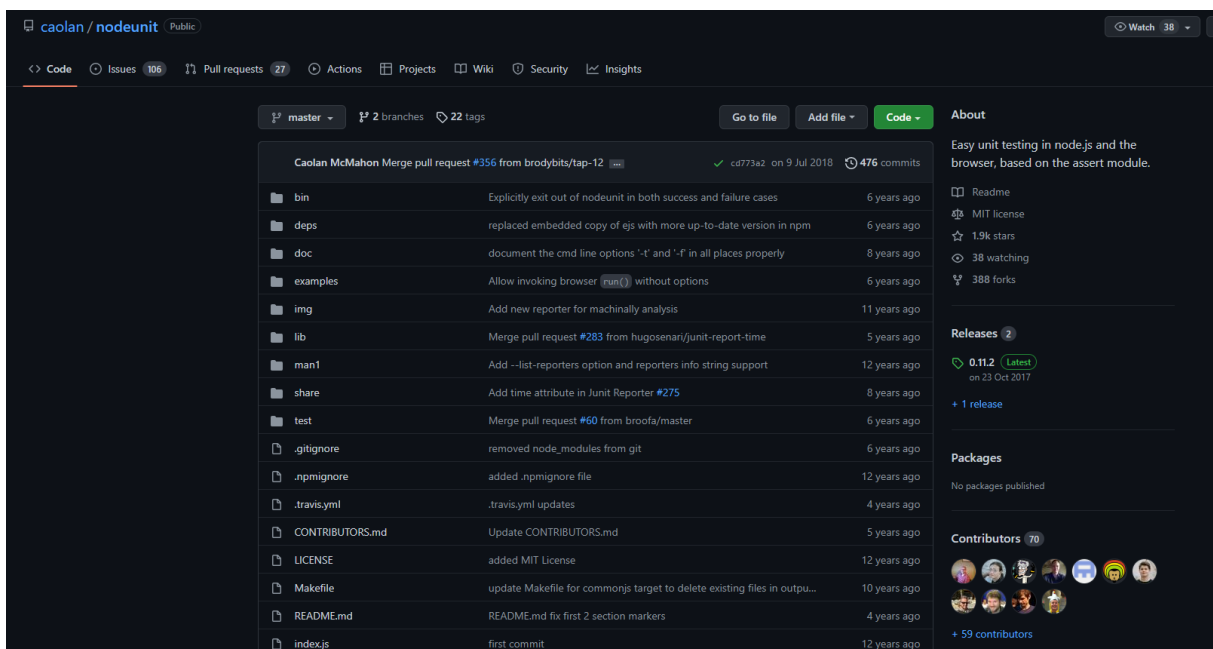


Metody DevOps	Karolina Czader
Sprawozdanie 5	
lab 05	15.05.2022

**Cel:** przygotować pipeline dla build, test, deploy i publish programu używanego we wcześniejszych laboratoriach.

Technologie użyte w laboratorium: Jenkins, Docker, git, npm.

Wybrany repozytorium do laboratorium jest aplikacja będąca frameworkiem do testów dla node.js.



Link do repozytorium: <https://github.com/caolan/nodeunit>

Instalowanie aplikacji i uruchomienie testów

```
karolina@karolina-VirtualBox:~$ ls
Dokumenty  examples.desktop  Muzyka  nodeunit  Obrazy  Pobrane  Publiczny  Pulpit  Szablony  Wideo
karolina@karolina-VirtualBox:~$ cd nodeunit/
karolina@karolina-VirtualBox:~/nodeunit$ ls
bin      deps  examples  index.js  LICENSE  man1      package.json  share
CONTRIBUTORS.md  doc  img      lib      Makefile  nodelint.cfg  README.md    test
karolina@karolina-VirtualBox:~/nodeunit$ npm install
npm WARN deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/3142
npm WARN deprecated har-validator@5.1.5: this library is no longer supported
npm WARN deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances
, which is known to be problematic. See https://v8.dev/blog/math-random for details.

> ejs@2.7.4 postinstall /home/karolina/nodeunit/node_modules/ejs
> node ./postinstall.js

nodeunit@0.11.2 /home/karolina/nodeunit
```

```
karolina@karolina-VirtualBox:~/nodeunit$ npm test

> nodeunit@0.11.3 test /home/karolina/nodeunit
> node ./bin/nodeunit

test-base
  testCalled
  testOk
  testEquals
  testSame
  testEqual
  testNotEqual
  testDeepEqual
  testNotDeepEqual
```

```
test-testcase-legacy
  testTestCase
  tearDownAfterError
  catchSetUpError
  setUpErrorCallback
  catchTearDownError
  tearDownErrorCallback
  testErrorAndtearDownError
  testCaseGroups
  nestedTestCases
  deepNestedTestCases

OK: 227 assertions (1862ms)
karolina@karolina-VirtualBox:~/nodeunit$
```

## KOLEJNE KROKI

Na początek należało utworzyć nowego użytkownika jenkinsa.

Kind

Username with password

Scope ?

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

Username ?

karczader

☒ Treat username as secret ?

Password ?

.....

ID ?

Description ?

OK

Kolejnym krokiem było utworzenie nowego pipeline'a i dodanie w konfiguracji linku do repozytorium na githubie, które zostało użyte do sklonowania.

Stworzony został dockerfile, który będzie potrzebny do stworzenia środowiska do zbudowania programu.

```
karolina@karolina-VirtualBox:~/Dokumenty/DevOps/MDO2022_S/ITE/GCL05/KC403251/Lab04$ nano jenkins_df
FROM jenkins/jenkins:2.332.1-jdk11
USER root
RUN apt-get update && apt-get install -y lsb-release
RUN curl -fsSL /usr/share/keyrings/docker-archive-keyring.asc \
    https://download.docker.com/linux/debian/gpg
RUN echo "deb [arch=$(dpkg --print-architecture) \
    signed-by=/usr/share/keyrings/docker-archive-keyring.asc] \
    https://download.docker.com/linux/debian \
    $(lsb_release -cs) stable" > /etc/apt/sources.list.d/docker.list
RUN apt-get update && apt-get install -y docker-ce-cli
RUN groupadd docker
RUN usermod -aG docker jenkins
USER jenkins
RUN jenkins-plugin-cli --plugins "blueocean:1.25.3 docker-workflow:1.28"
karolina@karolina-VirtualBox:~/Dokumenty/DevOps/MDO2022_S/ITE/GCL05/KC403251/Lab04$
```

## GOTOWY PIPELINE

```
pipeline {
  agent any

  stages {
    stage('Build') {
      steps {
        git branch: 'KC403251', credentialsId: '528a11c7-cc24-462b-b769-140748927e5b', url:
'https://github.com/InzynieriaOprogramowaniaAGH/MDO2022_S'
        dir('ITE/GCL05/KC403251/Lab03'){
          sh "docker build -t app -f Dockerfile_build ."
        }
        sh "mkdir -p publish"
      }
      post
      {
        failure {
          sh "rm -rf publish"
        }
      }
    }
    stage('CopyBuild'){
      agent
      {
        docker{
          image 'app'
          args '-v input-volume:/input-volume -v output-volume:/output-volume --user root'
          reuseNode true
        }
      }
      steps{
        sh "cp -r /app/nodeunit/* /output-volume"
        sh "pwd"
        sh "ls /"
      }
    }
  }
  stage('Test') {
```

```

    steps {
        dir("ITE/GCL05/KC403251/Lab03"){
            sh "docker build -t test -f Dockerfile_test ."
        }
    }
}
stage('CopyTest'){
    agent
    {
        docker{
            image 'test'
            args '-v output-volume:/output-volume -v deploy-volume:/deploy-volume --user root'
            reuseNode true
        }
    }
    steps{
        sh "cd /output-volume && npm test"
        sh "cd /output-volume && npm pack && ls && cp nodeunit*.tgz /deploy-volume"
        sh "ls /deploy-volume"
    }
}
stage("Deploy"){
    agent {
        docker {
            image 'app'
            args '-v deploy-volume:/deploy-volume --user root'
            reuseNode true
        }
    }
    steps {
        sh "ls /"
        sh "npm install /deploy-volume/nodeunit*.tgz"
        sh "cp /deploy-volume/nodeunit*.tgz ./publish"
    }
}
stage("Publish"){
    steps {
        sh "ls -la"
        sh "ls publish"
        archiveArtifacts artifacts:"publish/nodeunit*.tgz", fingerprint: true
        sh 'cat publish/nodeunit*.tgz | sha512sum > publish/checksum.txt'
        archiveArtifacts artifacts:"publish/checksum.txt", fingerprint: true
    }
}
}
}
}

```

**Stage Build** - kroki wykonywane wewnątrz tego stage: git branch odpowiada za połączenie z gitem za pomocą pluginu jenkinsowego, określony jest branch i podany link do repozytorium. Post – wewnątrz stage, po stepsach. Zdefiniowane jest co się dzieje, jeśli pipeline zwróci błąd w tym miejscu, w tym przypadku usuwa się plik publish, aby się one nie nawarstwiały.

**Stage CopyBuild:** kontener, który się buduje na obrazie, który już wykonał builda. Podpinamy dwa volumine: input i output, do kontenera wyjściowego kopiujemy zawartość pliku.

**Stage Test:** agent dockera na podstawie dockerfile buduje obraz, a potem w dirze jesr określone w jakim miejscu w workspace znajduje się ten obraz.

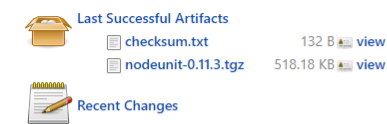
**Stage CopyTest:** przechodzimy na volumine wyjściowy i pakujemy program do formatu .tgz przy pomocy npm packa. Skopiowanie paczki z volumine wyjściowego na volume deploy.

**Stage Deploy:** uruchomienie agenta dockera na image node:latest z podpiętym volumine deploy, aby sprawdzić, czy paczka zbudowana w kroku testowym i skopiowana na volume jest spakowana poprawnie i zainstaluje się. Potem plik jest kopiowany z deploy volumine do publish.

**Stage Publish:** pakowanie artefaktu, który jest biblioteką spakowaną npm packiem, nadany mu zostaje fingerprint i na koniec liczona jest suma kontrolna, aby porównać, że to ja jestem autorem paczki i ją udostępniam, jest ona także udostępniona jako artefakt.

## DZIAŁANIE PIPELINE

Uruchomienie pipeline'a:



### Stage View



		Build	CopyBuild	Test	CopyTest	Deploy	Publish
Average stage times: (Average full run time: ~51s)		21s	5s	509ms	5s	7s	1s
#39	May 12 11:48	1min 29s	9s	786ms	7s	8s	1s



Build zakończony sukcesem:

## Build #39 (May 12, 2022, 9:48:36 AM)



### Build Artifacts

 [checksum.txt](#) 132 B  [view](#)

 [nodeunit-0.11.3.tgz](#) 518.18 KB  [view](#)



Started by user [Karolina Czader](#)



**Revision:** df88adee3a2dc90c232c02cac38bd628d567a925

**Repository:** [https://github.com/InzynieriaOprogramowaniaAGH/MDO2022\\_S](https://github.com/InzynieriaOprogramowaniaAGH/MDO2022_S)

- [refs/remotes/origin/KC403251](#)

Przykładowe logi z builda:

## Console Output

```
Started by user Karolina Czader
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/jenkins_home/workspace/Pipeline_v1
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Build)
[Pipeline] git
The recommended git tool is: NONE
using credential 528a11c7-cc24-462b-b769-140748927e5b
> git rev-parse --resolve-git-dir /var/jenkins_home/workspace/Pipeline_v1/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/InzynieriaOprogramowaniaAGH/MDO2022_S # timeout=10
Fetching upstream changes from https://github.com/InzynieriaOprogramowaniaAGH/MDO2022_S
> git --version # timeout=10
> git --version # 'git version 2.30.2'
using GIT_ASKPASS to set credentials
> git fetch --tags --force --progress -- https://github.com/InzynieriaOprogramowaniaAGH/MDO2022_S +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/KC403251^{commit} # timeout=10
Checking out Revision df88adee3a2dc90c232c02cac38bd628d567a925 (refs/remotes/origin/KC403251)
> git config core.sparsecheckout # timeout=10
> git checkout -f df88adee3a2dc90c232c02cac38bd628d567a925 # timeout=10
> git branch -a -v --no-abbrev # timeout=10
> git branch -D KC403251 # timeout=10
> git checkout -b KC403251 df88adee3a2dc90c232c02cac38bd628d567a925 # timeout=10
Commit message: "KC403251"
> git rev-list --no-walk df88adee3a2dc90c232c02cac38bd628d567a925 # timeout=10
[Pipeline] dir
Running in /var/jenkins_home/workspace/Pipeline_v1/ITE/GCL05/KC403251/Lab03
[Pipeline] {
[Pipeline] sh
+ docker build -t app -f Dockerfile_build .
Sending build context to Docker daemon 1.034MB
```

```
[Pipeline] archiveArtifacts
Archiving artifacts
Recording fingerprints
[Pipeline] sh
+ sha512sum
+ cat publish/nodeunit-0.11.3.tgz
[Pipeline] archiveArtifacts
Archiving artifacts
Recording fingerprints
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

## PRZETESTOWANIE DZIAŁANIA

Pobranie i zainstalowanie artefaktu z jenkinsa:

```
karolina@karolina-VirtualBox: ~/Pobrane
karolina@karolina-VirtualBox:~/Pobrane$ ls
nodeunit-0.11.3.tgz
karolina@karolina-VirtualBox:~/Pobrane$ node -v
v8.10.0
karolina@karolina-VirtualBox:~/Pobrane$ npm install nodeunit-0.11.3.tgz
npm WARN deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/3142
npm WARN deprecated har-validator@5.1.5: this library is no longer supported
npm WARN deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.
> ejs@2.7.4 postinstall /home/karolina/Pobrane/node_modules/ejs
> node ./postinstall.js
/home/karolina/Pobrane
├─ nodeunit@0.11.3
├─ ejs@2.7.4
└─
```

Instalacja sie powiodła:

```
├─ write-file-atomic@2.4.3
├─ graceful-fs@4.2.10
├─ imurmurhash@0.1.4
└─ yapool@1.0.0
npm WARN enoent ENOENT: no such file or directory, open '/home/karolina/Pobrane/package.json'
npm WARN Pobrane No description
npm WARN Pobrane No repository field.
npm WARN Pobrane No README data
npm WARN Pobrane No license field.
karolina@karolina-VirtualBox:~/Pobrane$ ls
node_modules nodeunit-0.11.3.tgz
karolina@karolina-VirtualBox:~/Pobrane$
```