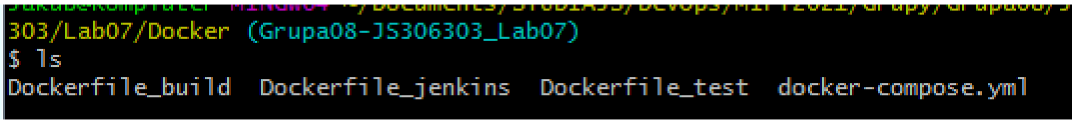
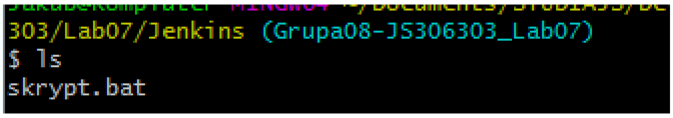
DevOps 7 JS306303

Zad 1

Stworzyłem folder Docker, do którego dodałem pliki źródłowego technologii Docker, których używałem do pracy z DeltaChat i Jenkinsem oraz analogicznie folder Jenkins

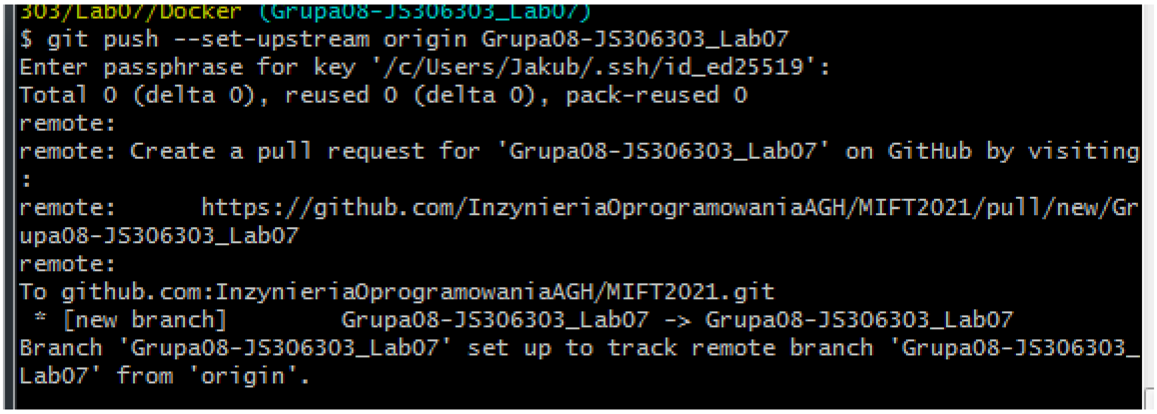


```
303/Lab07/Docker (Grupa08-JS306303_Lab07)
$ ls
Dockerfile_build  Dockerfile_jenkins  Dockerfile_test  docker-compose.yml
```



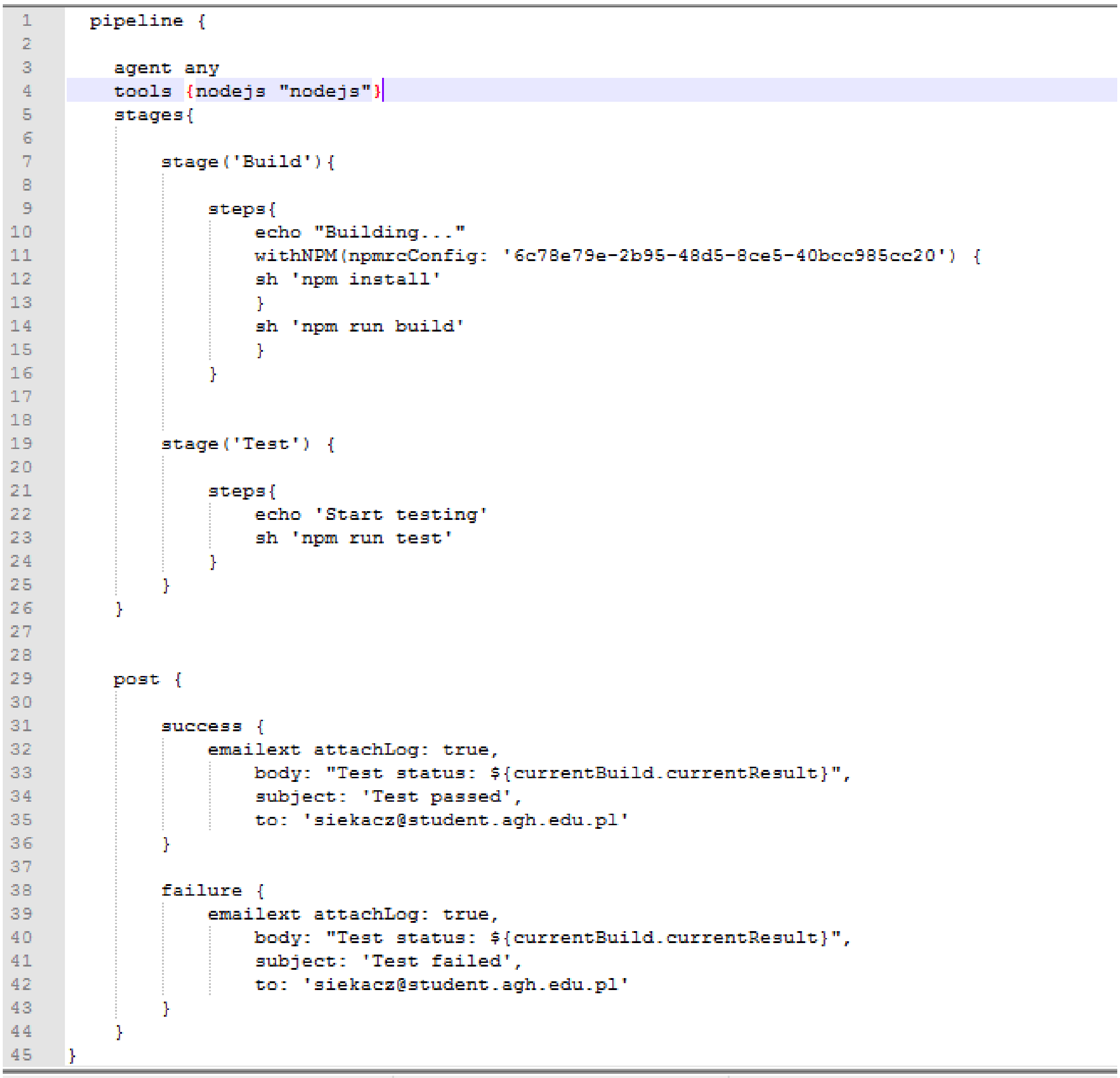
```
303/Lab07/Jenkins (Grupa08-JS306303_Lab07)
$ ls
skrypt.bat
```

Jako, że było to lokalnie, to musiałem zrobić odpowiedniego commit'a i git push'a by wrzucić to do repozytorium



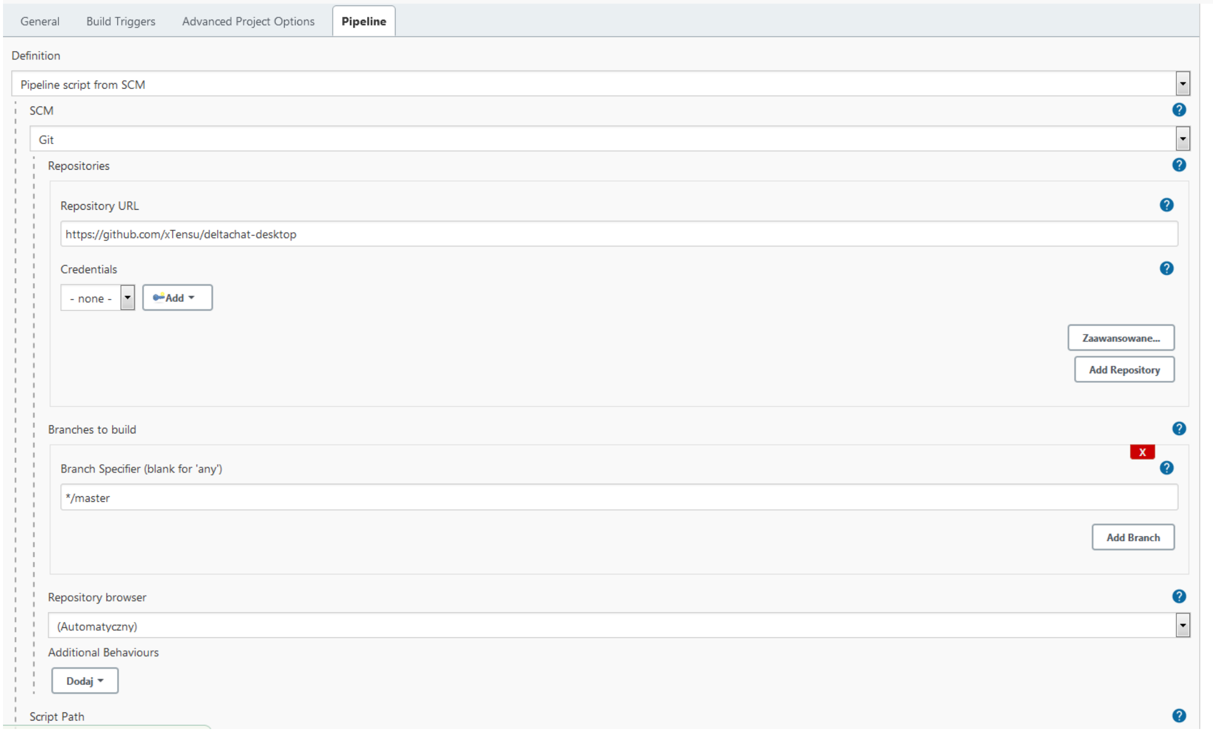
```
303/Lab07/Docker (Grupa08-JS306303_Lab07)
$ git push --set-upstream origin Grupa08-JS306303_Lab07
Enter passphrase for key '/c/Users/Jakub/.ssh/id_ed25519':
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'Grupa08-JS306303_Lab07' on GitHub by visiting
:
remote:      https://github.com/InzynieriaOprogramowaniaAGH/MIFT2021/pull/new/Gr
upa08-JS306303_Lab07
remote:
To github.com:InzynieriaOprogramowaniaAGH/MIFT2021.git
 * [new branch]        Grupa08-JS306303_Lab07 -> Grupa08-JS306303_Lab07
Branch 'Grupa08-JS306303_Lab07' set up to track remote branch 'Grupa08-JS306303_
Lab07' from 'origin'.
```

Stworzyłem Jenkins file:



```
  pipeline {

    agent any
    tools {nodejs "nodejs"}
    stages{

        stage('Build'){

            steps{
                echo "Building..."
                withNPM(npmrcConfig: '6c78e79e-2b95-48d5-8ce5-40bcc985cc20') {
                sh 'npm install'
                }
                sh 'npm run build'
                }
            }


        stage('Test') {

            steps{
                echo 'Start testing'
                sh 'npm run test'
            }
        }
    }


    post {

        success {
            emailext attachLog: true,
                body: "Test status: ${currentBuild.currentResult}",
                subject: 'Test passed',
                to: 'siekacz@student.agh.edu.pl'
        }

        failure {
            emailext attachLog: true,
                body: "Test status: ${currentBuild.currentResult}",
                subject: 'Test failed',
                to: 'siekacz@student.agh.edu.pl'
        }
    }
}
```

Następnie użyłem forka na Deltachat-desktop i stworzyłem Pipeline



Definition

Pipeline script from SCM

SCM

Git

Repositories

Repository URL

https://github.com/xTensu/deltachat-desktop

Credentials

- none -

Branches to build

Branch Specifier (blank for 'any')

*/master

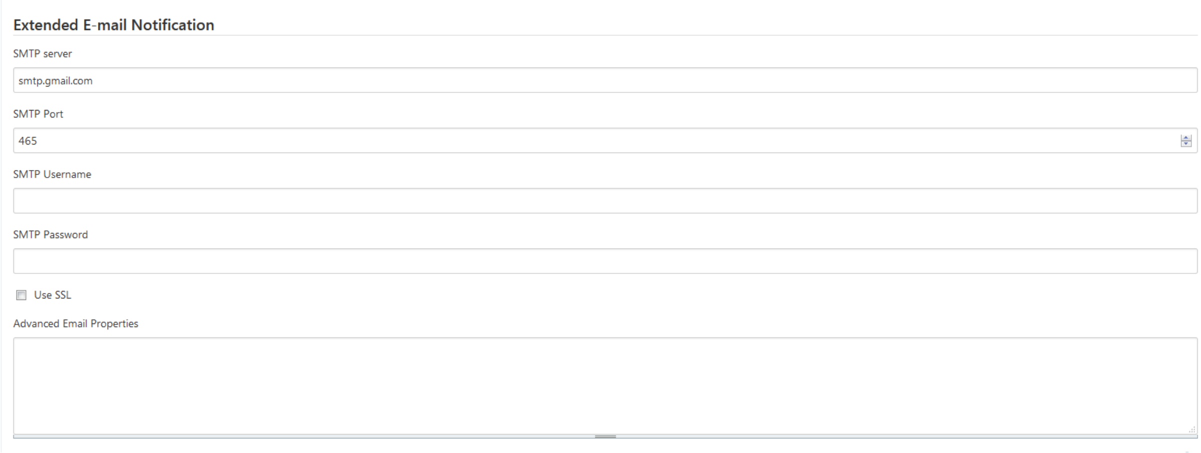
Repository browser

(Automatyczny)

Additional Behaviours

Script Path

Musiałem skonfigurować serwer SMTP w Jenkinsie



Extended E-mail Notification

SMTP server

smtp.gmail.com

SMTP Port

465

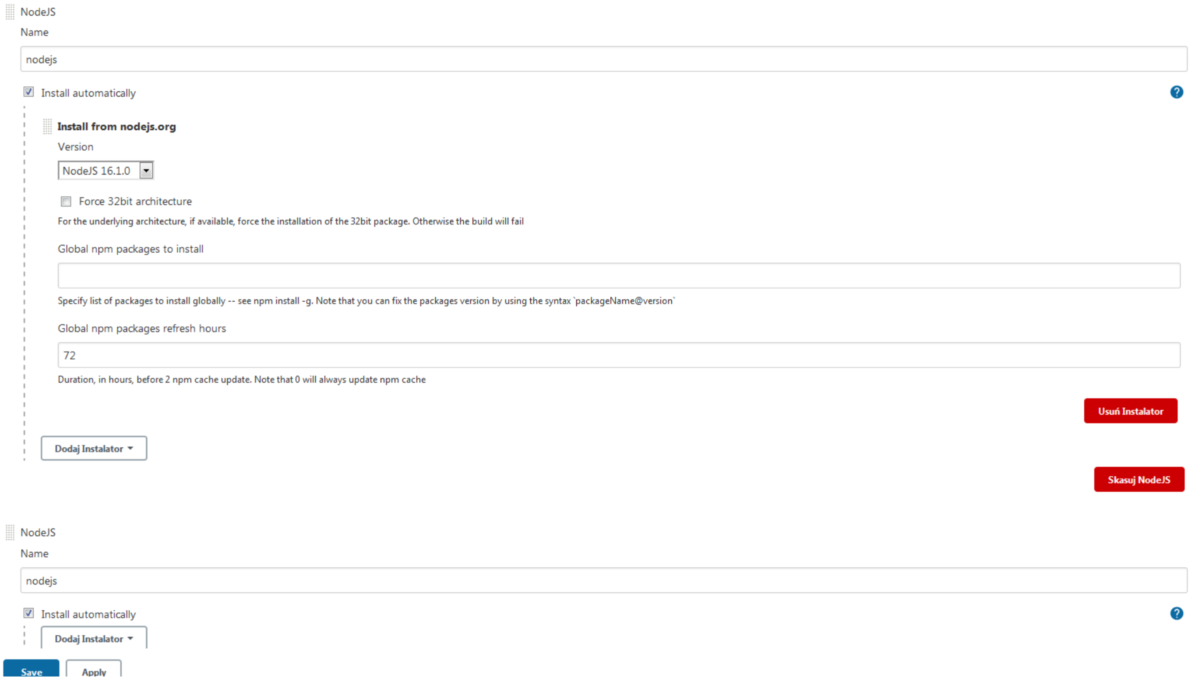
SMTP Username

SMTP Password

☐ Use SSL

Advanced Email Properties

I pobrać NodeJS, by mieć npm na Jenkinsie, po czym go skonfigurować



NodeJS

Name

nodejs

☑ Install automatically

Install from nodejs.org

Version

NodeJS 16.1.0

□ Force 32bit architecture

For the underlying architecture, if available, force the installation of the 32bit package. Otherwise the build will fail

Global npm packages to install

Specify list of packages to install globally -- see npm install -g. Note that you can fix the packages version by using the syntax `packageName@version`

Global npm packages refresh hours

72

Duration, in hours, before 2 npm cache update. Note that 0 will always update npm cache

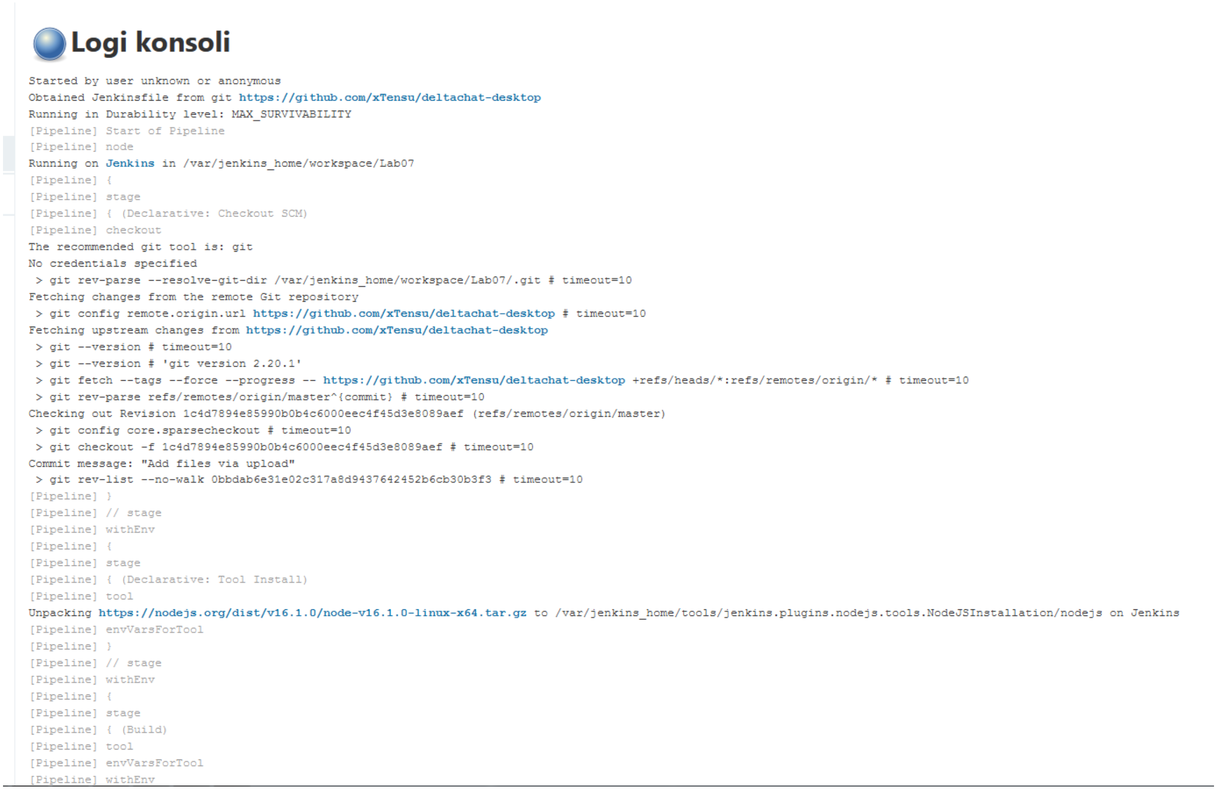
NodeJS

Name

nodejs

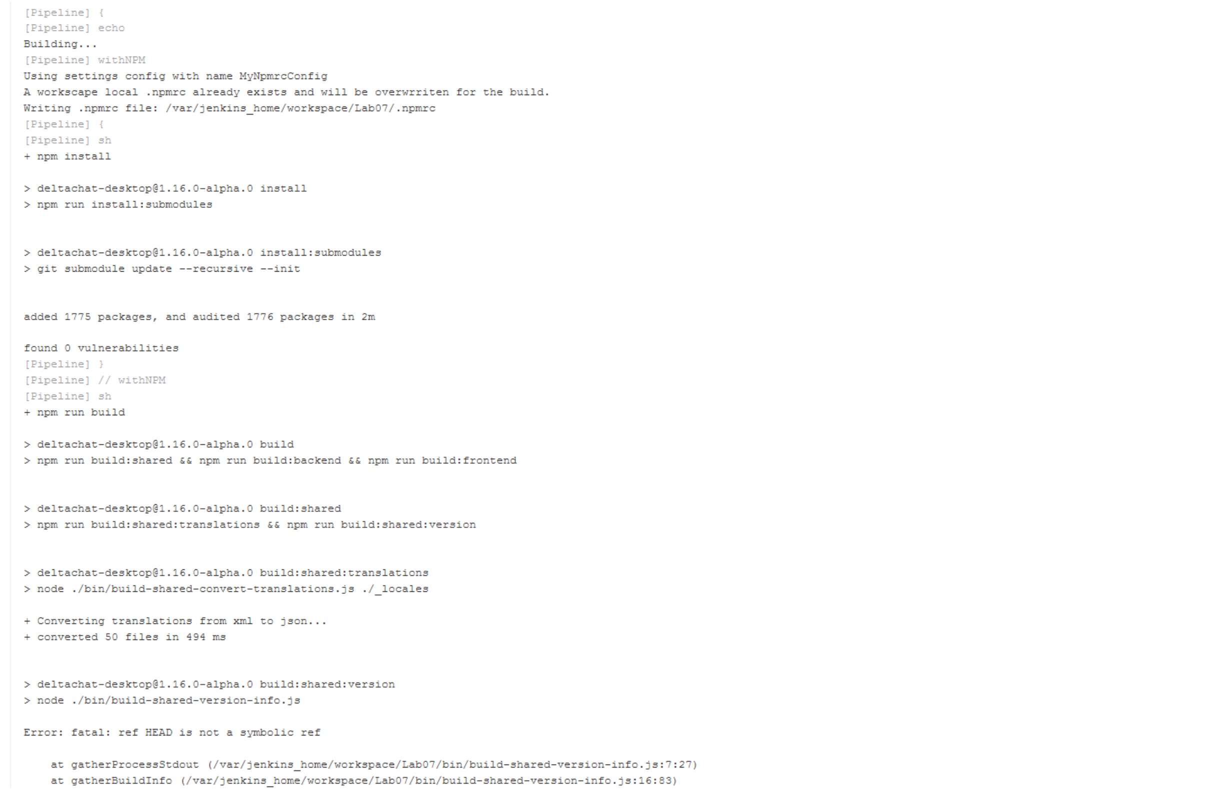
☑ Install automatically

I ostatecznie się udało:

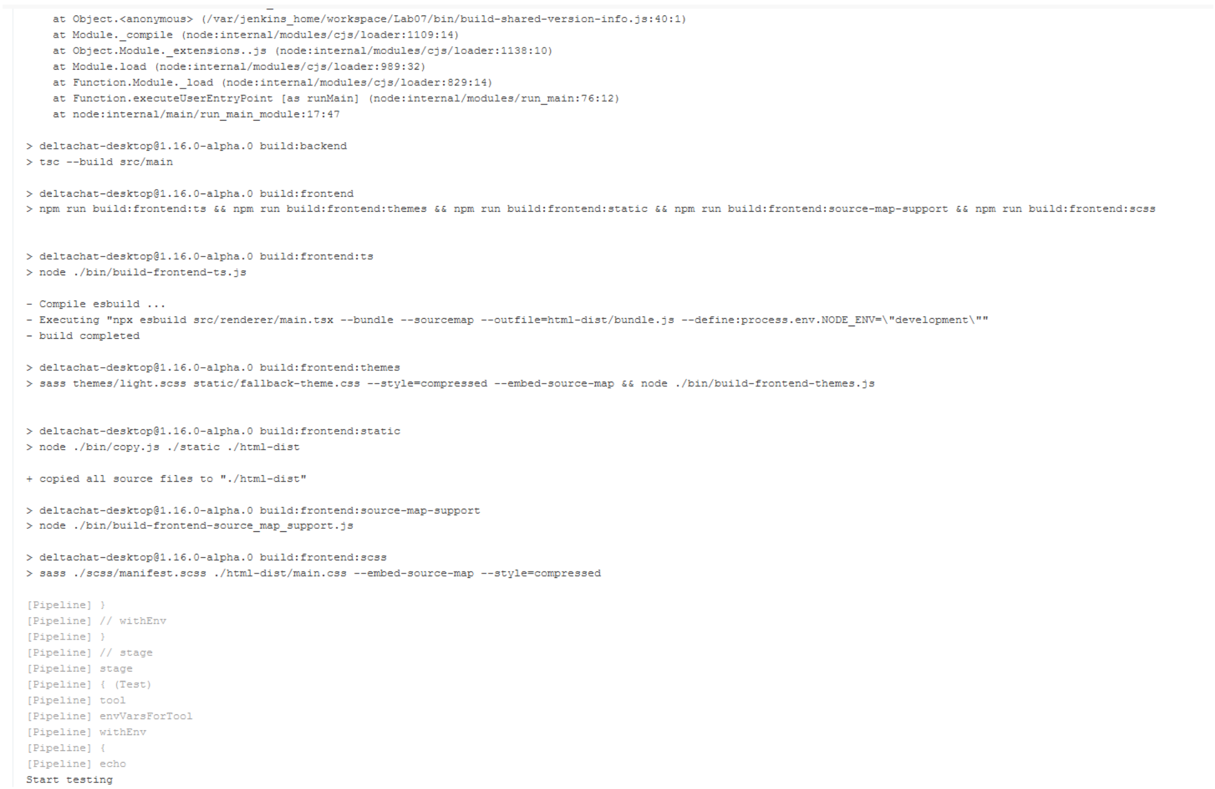


Logi konsoli

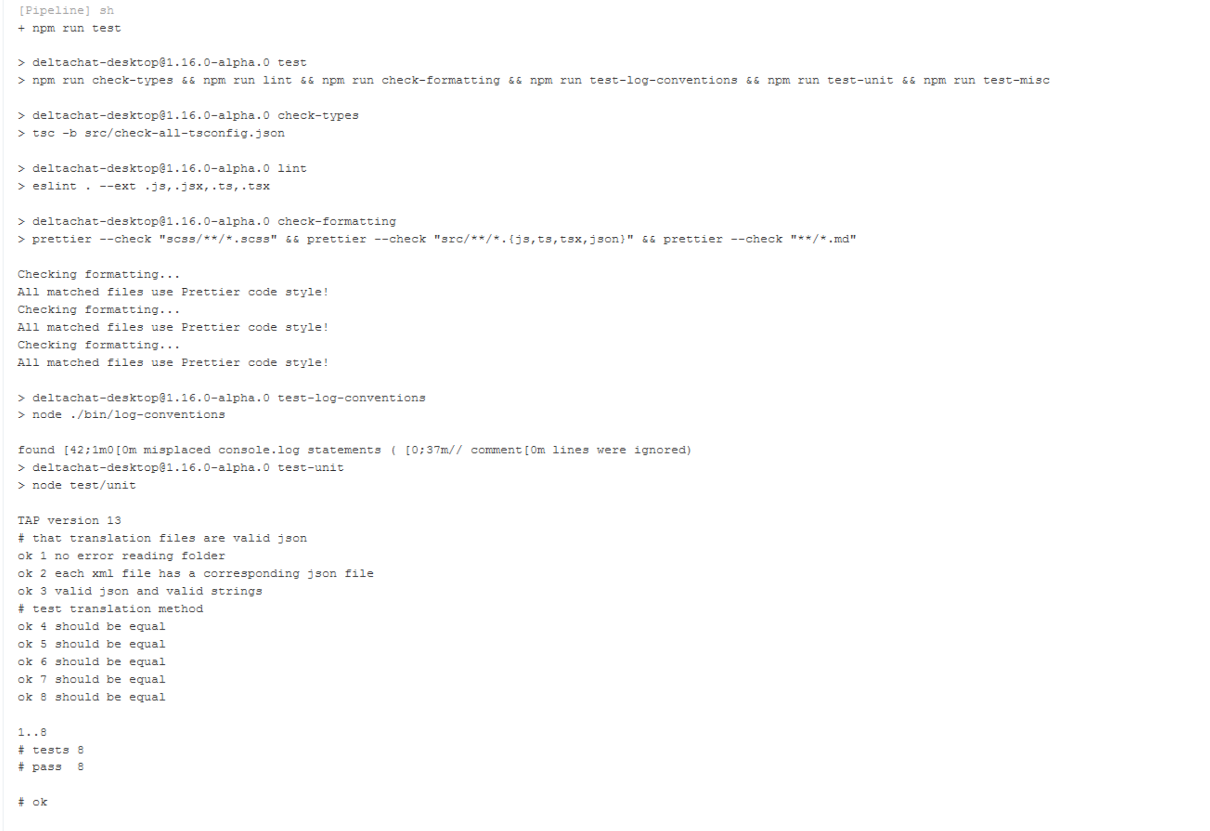
```
Started by user unknown or anonymous
Obtained Jenkinsfile from git https://github.com/xTensu/deltachat-desktop
Running in Durability level: MAX_SURVIVABILITY
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/jenkins_home/workspace/Lab07
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
The recommended git tool is: git
No credentials specified
 > git rev-parse --resolve-git-dir /var/jenkins_home/workspace/Lab07/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/xTensu/deltachat-desktop # timeout=10
Fetching upstream changes from https://github.com/xTensu/deltachat-desktop
 > git --version # timeout=10
 > git --version # 'git version 2.20.1'
 > git fetch --tags --force --progress -- https://github.com/xTensu/deltachat-desktop +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision 1c4d7894e85990b0b4c6000eec4f45d3e8089aef (refs/remotes/origin/master)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 1c4d7894e85990b0b4c6000eec4f45d3e8089aef # timeout=10
Commit message: "Add files via upload"
 > git rev-list --no-walk 0bbdab6e31e02c317a8d9437642452b6cb30b3f3 # timeout=10
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Tool Install)
[Pipeline] tool
Unpacking https://nodejs.org/dist/v16.1.0/node-v16.1.0-linux-x64.tar.gz to /var/jenkins_home/tools/jenkins.plugins.nodejs.tools.NodeJSInstallation/nodejs on Jenkins
[Pipeline] envVarsForTool
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Build)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] withEnv
```



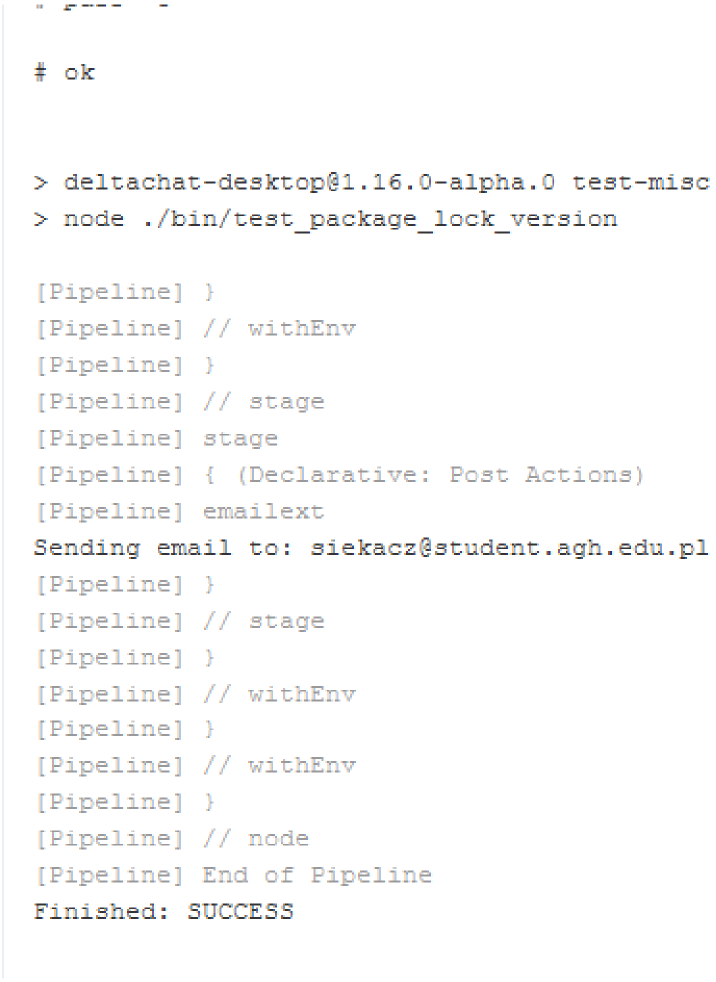
```
[Pipeline] {
[Pipeline] echo
Building...
[Pipeline] withNPM
Using settings config with name MyNpmrcConfig
A workscape local .npmrc already exists and will be overwrriten for the build.
Writing .npmrc file: /var/jenkins_home/workspace/Lab07/.npmrc
[Pipeline] {
[Pipeline] sh
+ npm install

> deltachat-desktop@1.16.0-alpha.0 install
> npm run install:submodules


> deltachat-desktop@1.16.0-alpha.0 install:submodules
> git submodule update --recursive --init


added 1775 packages, and audited 1776 packages in 2m

found 0 vulnerabilities
[Pipeline] }
[Pipeline] // withNPM
[Pipeline] sh
+ npm run build

> deltachat-desktop@1.16.0-alpha.0 build
> npm run build:shared && npm run build:backend && npm run build:frontend


> deltachat-desktop@1.16.0-alpha.0 build:shared
> npm run build:shared:translations && npm run build:shared:version


> deltachat-desktop@1.16.0-alpha.0 build:shared:translations
> node ./bin/build-shared-convert-translations.js ./_locales

+ Converting translations from xml to json...
+ converted 50 files in 494 ms


> deltachat-desktop@1.16.0-alpha.0 build:shared:version
> node ./bin/build-shared-version-info.js

Error: fatal: ref HEAD is not a symbolic ref

    at gatherProcessStdout (/var/jenkins_home/workspace/Lab07/bin/build-shared-version-info.js:7:27)
    at gatherBuildInfo (/var/jenkins_home/workspace/Lab07/bin/build-shared-version-info.js:16:83)
```



```
    at Object.<anonymous> (/var/jenkins_home/workspace/Lab07/bin/build-shared-version-info.js:40:1)
    at Module._compile (node:internal/modules/cjs/loader:1109:14)
    at Object.Module._extensions..js (node:internal/modules/cjs/loader:1138:10)
    at Module.load (node:internal/modules/cjs/loader:989:32)
    at Function.Module._load (node:internal/modules/cjs/loader:829:14)
    at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:76:12)
    at node:internal/main/run_main_module:17:47

> deltachat-desktop@1.16.0-alpha.0 build:backend
> tsc --build src/main

> deltachat-desktop@1.16.0-alpha.0 build:frontend
> npm run build:frontend:ts && npm run build:frontend:themes && npm run build:frontend:static && npm run build:frontend:source-map-support && npm run build:frontend:scss


> deltachat-desktop@1.16.0-alpha.0 build:frontend:ts
> node ./bin/build-frontend-ts.js

- Compile esbuild ...
- Executing "npx esbuild src/renderer/main.tsx --bundle --sourcemap --outfile=html-dist/bundle.js --define:process.env.NODE_ENV=\"development\""
- build completed

> deltachat-desktop@1.16.0-alpha.0 build:frontend:themes
> sass themes/light.scss static/fallback-theme.css --style=compressed --embed-source-map && node ./bin/build-frontend-themes.js


> deltachat-desktop@1.16.0-alpha.0 build:frontend:static
> node ./bin/copy.js ./static ./html-dist

+ copied all source files to "./html-dist"

> deltachat-desktop@1.16.0-alpha.0 build:frontend:source-map-support
> node ./bin/build-frontend-source_map_support.js

> deltachat-desktop@1.16.0-alpha.0 build:frontend:scss
> sass ./scss/manifest.scss ./html-dist/main.css --embed-source-map --style=compressed

[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Test)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] withEnv
[Pipeline] {
[Pipeline] echo
Start testing
```



```
[Pipeline] sh
+ npm run test

> deltachat-desktop@1.16.0-alpha.0 test
> npm run check-types && npm run lint && npm run check-formatting && npm run test-log-conventions && npm run test-unit && npm run test-misc

> deltachat-desktop@1.16.0-alpha.0 check-types
> tsc -b src/check-all-tsconfig.json

> deltachat-desktop@1.16.0-alpha.0 lint
> eslint . --ext .js,.jsx,.ts,.tsx

> deltachat-desktop@1.16.0-alpha.0 check-formatting
> prettier --check "scss/**/*.scss" && prettier --check "src/**/*.{js,ts,tsx,json}" && prettier --check "**/*.md"

Checking formatting...
All matched files use Prettier code style!
Checking formatting...
All matched files use Prettier code style!
Checking formatting...
All matched files use Prettier code style!

> deltachat-desktop@1.16.0-alpha.0 test-log-conventions
> node ./bin/log-conventions

found [42;1m0[0m misplaced console.log statements ( [0;37m// comment[0m lines were ignored)
> deltachat-desktop@1.16.0-alpha.0 test-unit
> node test/unit

TAP version 13
# that translation files are valid json
ok 1 no error reading folder
ok 2 each xml file has a corresponding json file
ok 3 valid json and valid strings
# test translation method
ok 4 should be equal
ok 5 should be equal
ok 6 should be equal
ok 7 should be equal
ok 8 should be equal

1..8
# tests 8
# pass  8

# ok
```



```
# pass  1

# ok


> deltachat-desktop@1.16.0-alpha.0 test-misc
> node ./bin/test_package_lock_version

[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] emailext
Sending email to: siekacz@student.agh.edu.pl
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```