# LU11b - Lokales Jenkinsfile mit Bedingungen

Bei der bisherigen Pipeline haben wir der Einfachheit halber das Pipeline-Script direkt auf dem Jenkins-Server hinterlegt. Das ist zwar schneller, um „herumzuspielen“, aber langfristig ist es sauberer, die Konfiguration in das Repository einzuchecken. Dasselbe gilt auch für andere Konfigurationsdateien wie sonar-project.properties.

Gewisse Variablen wie „BRANCH_NAME“ stehen automatisch zur Verfügung und müssen nicht mehr definiert werden. Es ist jedoch Vorsicht geboten, wenn Branchnamen Zeichen enthalten, die zum Beispiel für Containernamen nicht gültig sind.

Bei SonarQube kann die kostenlose Community-Edition keine Multibranch-Analyse erstellen, dafür wäre die Developer-Edition nötig. Für unseren Fall reicht es jedoch aus, wenn nur der Entwicklungs-Branch analysiert wird. Damit die anderen Branches nicht aus Versehen dazwischenfunken, kann die „When“-Expression verwendet werden.

```
            when {
                branch 'develop'
            }
```

Diese kann auch verwendet werden, um die Branches einzuschränken, welche deployed werden sollen.

```
            when {
                anyOf {
                    branch 'master'
                    branch 'develop'
                }
            }
```

Ein Jenkinsfile (ohne Backend) könnte folgendermassen aussehen:

```
pipeline {
    agent any

    options {
        disableConcurrentBuilds()
        timeout(time: 10, unit: 'MINUTES')
    }

    environment {
        PROJECT_NAME       = "..."
        TARGET_DIR         =
"/var/jenkins_home/projects/${PROJECT_NAME}/${BRANCH_NAME}"
        SONAR_SCANNER_OPTS = "-Xmx512m"
    }

    stages {
```

```
        stage('Checkout') {
            steps {
                checkout scm
            }
        }

        stage('Build Frontend') {
            steps {
                dir('frontend') {
                    sh '''
                    npm ci ...
                        npm run build
                    '''
                }
            }
        }

        stage('SonarQube Analysis') {
            when {
                branch 'develop'
            }
            steps {
                sh """
                    echo "Starting SonarQube analysis of $PROJECT_NAME"
                    echo "SONAR_SCANNER_OPTS=$SONAR_SCANNER_OPTS"
                    echo "NODE_OPTIONS=$NODE_OPTIONS"
                """
                script {
                    def scannerHome = tool 'sonar-scanner'
                    withSonarQubeEnv('SonarQube') {
                        sh """
                        ${scannerHome}/bin/sonar-scanner \
                          -Dsonar.projectKey=${PROJECT_NAME} \
                          -Dsonar.branch.name=${BRANCH_NAME}
                        """
                    }
                }
            }
        }

        stage('Deploy Frontend') {
            when {
                anyOf {
                    branch 'master'
                    branch 'develop'
                }
            }
            steps {
                sh '''
                    echo "Deploying frontend to $TARGET_DIR"
```

```
                    mkdir -p "$TARGET_DIR"
                    rm -rf "$TARGET_DIR"/*

                    cp -r frontend/build/* "$TARGET_DIR"/
                '''
            }
        }
    }

    post {
        always {
            deleteDir()
        }
    }
}
```

Sie finden dieses Beispiel auch in folgendem Repository:
https://github.com/AlexanderPeter/cicd/blob/develop/Jenkinsfile

From:
https://wiki.bzz.ch/ - **BZZ - Modulwiki**

Permanent link:
**https://wiki.bzz.ch/de/modul/ffit/3-jahr/cicd/learningunits/lu11/b**

Last update: **2026/05/12 02:10**