Rendu Projet CI/CD

RATSIMBAZAFY Tojoniaina Mbolasafidy

Houssem

exercice1:

```
pipeline {
    agent any

    stages {
        stage('Generate System Info Log') {
            steps {
                script {
                    def date = sh(script: 'date "+%Y-%m-%d %H:%M:%S"', returnStdout: true).trim()
                    def hostname = sh(script: 'hostname', returnStdout: true).trim()
                    def nb_coeurs = sh(script: "grep -c '^processor' /proc/cpuinfo", returnStdout: true).trim()
                    def ram = sh(script: "grep MemTotal /proc/meminfo | awk '{printf \"%d MB\", \$2/1024}'", returnStdout: true).trim()
                    def util_disque = sh(script: "df / | tail -1 | awk '{print \$5}'", returnStdout: true).trim()

                    // On crée le fichier dans /var/jenkins_home/tmp pour qu'il soit accessible depuis Windows
                    sh """
                        mkdir -p /var/jenkins_home/tmp
                        cat > /var/jenkins_home/tmp/exercice1.log <<EOF
-----
Exécution de la pipeline Jenkins Exercice 1
Début d'exécution à ${date} sur ${hostname}
Cette machine dispose de ${nb_coeurs} coeurs de CPU, et ${ram} de RAM.
Le disque principal est utilisé à ${util_disque}.
-----
EOF
                    """
                    echo "Fichier exercice1.log généré dans /var/jenkins_home/tmp"
                }
            }
        }
    }
}
```

exercice2:

```
pipeline {
    agent any

    tools {
        nodejs 'nodejs-18'
    }

    stages {

        stage('Checkout') {
            steps {
                echo "Récupération du code depuis GitHub"
```

```
      git branch: 'main', url: 'https://github.com/tojo2803/CiCdProject.git'
    }
  }

  stage('Install Backend') {
    steps {
      dir('Backend') {
        echo "Installation des dépendances backend"
        sh 'npm install'
      }
    }
  }

  stage('Run Backend Tests') {
    steps {
      dir('Backend') {
        echo "Lancement des tests unitaires et d’intégration"
        sh 'npm test'
      }
    }
  }
}
post {
  success {
    echo '🎉 pipline sucess'
  }
```

```
    failure {
      echo '❌ Pipeline échouée'
    }
  }
}
```

exercice3:

Jenkins file:

```
pipeline {
   agent any

  tools {
    nodejs 'nodejs-18'
  }

  stages {
    stage('Checkout') {
      steps {
        echo "Récupération du code depuis GitHub"
        git branch: 'main', url: 'https://github.com/tojo2803/CiCdProject.git'
      }
    }

    stage('Install Backend') {
      steps {
        dir('Backend') {
          echo "Installation des dépendances backend"
          sh 'npm install'
        }
      }
    }

    stage('Run Backend Tests') {
      steps {
        dir('Backend') {
          echo "Lancement des tests unitaires et d’intégration"
          sh 'npm test'
        }
      }
    }
  }

  post {
    success {
      echo 'Pipeline terminé avec succès 🎉'
    }
    failure {
      echo 'Pipeline échoué ❌'
    }
  }
}
```

exercice4:

Jenkins file:

```
pipeline {
    agent any

    tools {
        nodejs 'nodejs-18'
    }

    stages {

        stage('Checkout') {
            steps {
                echo "Récupération du code depuis GitHub"
                git branch: 'main', url: 'https://github.com/tojo2803/CiCdProject.git'
            }
        }

        stage('Install Backend') {
            steps {
                dir('Backend') {
                    echo "Installation des dépendances backend"
                    sh 'npm install'
                }
            }
        }

        stage('Run Backend Tests') {
```

```
        steps {
            dir('Backend') {
                echo "Lancement des tests unitaires et d’intégration"
                sh 'npm test'
            }
        }
    }

    stage('Packager frontend') {
        steps {
           sh 'tar --exclude=Backend/frontend.tar.gz -czf frontend.tar.gz Frontend'
        }
    }

    stage('Packager backend') {
        steps {
            sh '''
                tar \
                --exclude=Backend/node_modules \
                --exclude=Backend/backend.tar.gz \
                -czf backend.tar.gz \
                Backend
            '''
        }
    }
}
```

```
    post {
        success {
            archiveArtifacts artifacts: '*.tar.gz'
            echo '🎉 Livraison continue réussie'
        }
        failure {
            echo '❌ Pipeline échouée'
        }
    }
}
```

exercice5:

Jenkins file:

```
pipeline {
    agent any

    tools {
        nodejs 'nodejs-18'
    }

    environment {
        SERVER_USER = 'root'
        SERVER_IP   = '192.168.1.50'
        SSH_CRED    = 'vm-ssh-key'
    }
```

```
stages {

  stage('Clean workspace') {
    steps {
      deleteDir()
    }
  }

  stage('Checkout') {
    steps {
      echo "Récupération du code depuis GitHub"
      git branch: 'exercice5',
        url: 'https://github.com/tojo2803/CiCdProject.git'
    }
  }

  stage('Install Backend') {
    steps {
      dir('Backend') {
        echo "Installation des dépendances backend"
        sh 'npm install'
      }
    }
  }
```

```
stage('Run Backend Tests') {
  steps {
    dir('Backend') {
      echo "Lancement des tests unitaires et d’intégration"
      sh 'npm test'
    }
  }
}

stage('Package Frontend') {
  steps {
   sh 'tar --exclude=Backend/frontend.tar.gz -czf frontend.tar.gz Frontend'
  }
}

stage('Package Backend') {
  steps {
    sh '''
      tar --exclude=Backend/node_modules -czf backend.tar.gz Backend
    '''
  }
}

stage('Deploy Frontend') {
  steps {
    sshagent(credentials: [env.SSH_CRED]) {
      sh '''
        # Copier l’archive sur la VM
        scp frontend.tar.gz ${SERVER_USER}@${SERVER_IP}:/tmp/

        # Déployer dans /var/www/html
        ssh ${SERVER_USER}@${SERVER_IP} "
          sudo rm -rf /var/www/html/*
          sudo tar -xzf /tmp/frontend.tar.gz -C /var/www/html --strip-components=1
        "
      '''
    }
  }
}

stage('Deploy Backend') {
  steps {
    sshagent(credentials: [env.SSH_CRED]) {
      sh '''
        scp backend.tar.gz ${SERVER_USER}@${SERVER_IP}:/tmp/

        ssh ${SERVER_USER}@${SERVER_IP} "
          rm -rf /opt/backend/*
          tar -xzf /tmp/backend.tar.gz -C /opt/backend --strip-components=1
          cd /opt/backend
          npm install
          pm2 restart backend || pm2 start index.js --name backend
```

```
                        "
                    '''
                }
            }
        }
    }

    post {
        success {
            archiveArtifacts artifacts: '*.tar.gz'
            echo '🚀 Déploiement continu réussi'
        }
        failure {
            echo '❌ Déploiement échoué'
        }
    }

}
```

exercice6:

Jenkins file:

```
pipeline {
    agent any

    tools {
        nodejs 'nodejs-18'
    }
    parameters {
        string(name: 'APP_PORT', defaultValue: '3000', description: 'Numéro de port pour le backend')
    }

    environment {
        SERVER_USER = 'root'
        SERVER_IP   = '192.168.1.50'
        SSH_CRED    = 'vm-ssh-key'
        APP_PORT    = "${params.APP_PORT}"
    }

    stages {

        stage('Clean workspace') {
            steps {
                deleteDir()
            }
        }

        stage('Checkout') {
            steps {
                echo "Récupération du code depuis GitHub"
                git branch: 'exercice6',
                    url: 'https://github.com/tojo2803/CiCdProject.git'
    }
}

stage('Install Backend') {
    steps {
        dir('Backend') {
            echo "Installation des dépendances backend"
            sh 'npm install'
        }
    }
}

stage('Run Backend Tests') {
    steps {
        dir('Backend') {
            echo "Lancement des tests unitaires et d’intégration"
            sh 'npm test'
        }
    }
}

stage('Package Frontend') {
    steps {
       sh 'tar --exclude=Backend/frontend.tar.gz -czf frontend.tar.gz Frontend'
    }
}
```

```
stage('Package Backend') {
    steps {
        sh '''
            tar --exclude=Backend/node_modules -czf backend.tar.gz Backend
        '''
    }
}

stage('Deploy Frontend') {
    steps {
        sshagent(credentials: [env.SSH_CRED]) {
            sh """
                # Copier l’archive sur la VM
                scp frontend.tar.gz ${SERVER_USER}@${SERVER_IP}:/tmp/

                # Déployer dans /var/www/html
                ssh ${SERVER_USER}@${SERVER_IP} "
                    sudo rm -rf /var/www/html/*
                    sudo tar -xzf /tmp/frontend.tar.gz -C /var/www/html --strip-components=1
                "
            """
        }
    }
}
```

```
    stage('Deploy Backend') {
      steps {
        sshagent(credentials: [env.SSH_CRED]) {
          sh """
            scp backend.tar.gz ${SERVER_USER}@${SERVER_IP}:/tmp/

            ssh ${SERVER_USER}@${SERVER_IP} "
              rm -rf /opt/backend/*
              tar -xzf /tmp/backend.tar.gz -C /opt/backend --strip-components=1
              cd /opt/backend
              npm install

              # Définir le port du backend via la variable Jenkins
              export PORT=${APP_PORT}

              pm2 restart backend || pm2 start index.js --name backend --env PORT=${APP_PORT}
            "
          """
        }
      }
    }
  }

  post {
    success {
            archiveArtifacts artifacts: '*.tar.gz'

            echo '🚀 Déploiement continu réussi'

        }

        failure {

            echo '❌ Déploiement échoué'

        }

    }

}
```