**Devops CICD**

Problem: as we are progressing with CICD pipeline creation using maven as an language.

Where we have define multiple stages

1. Connecting to a pvt repo

2. Building the code using maven

3. Testing the code using maven

4. Code quality and code coverage testing.

5. Integration with sonar quebe

6. Conver the application into an docker container.

7. Create an ansible pipeline to install docker.

8. And finally deploy the container using ansible

Task: build the same pipeline in Jenkins using grunt or nodejs.

Dockerfile:

pipeline {

    agent any

    environment {

        dockerImage = ''

    }

    stages {

        stage('Poll SCM') {

            steps {

                git credentialsId: 'git-pass', url: 'git@github.com:avi207/devops\_node.git'

            }

        }

        stage('npm install') {

            steps {

                sh 'npm install'

            }

        }

        stage('test') {

            steps {

                sh 'npm run test'

            }

        }

        stage('Sonar'){

            steps {

                sh '''

                    sonar-scanner \

                        -Dsonar.projectKey=devopsnode \

                        -Dsonar.sources=. \

                        -Dsonar.host.url=http://172.173.248.120:9000 \

                        -Dsonar.login=sqp\_f98c8f509ab60596b3487c125f2b72054d7d3204

                '''

            }

        }

        stage('docker build') {

            steps{

                script{

                    dockerImage = docker.build("avi207/devops-node:latest")

                }

            }

        }

        stage('docker login and push') {

            steps {

                script {

                    withDockerRegistry(credentialsId: 'docker-key', url: "") {

                        dockerImage.push()

                    }

                }

            }

        }

        stage ('Install docker'){

            steps {

               ansiblePlaybook credentialsId: 'git-pass', disableHostKeyChecking: true, inventory: 'ansible/dev.inv', playbook: 'ansible/install-docker.yml'

            }

        }

        stage ('deploy image'){

            steps {

                ansiblePlaybook credentialsId: 'git-pass', disableHostKeyChecking: true, inventory: 'ansible/dev.inv', playbook: 'ansible/deploy-image.yml', vaultCredentialsId: 'vault-pass'

            }

        }

    }

}

Pipeline ss:

