**Stash and unstash in Jenkins Pipeline**

* Stash can be used to copy any stuff from stage to stage (even on different node)
* As an example, lets build a game of life and then stash gameoflife.war file from node 1 and unstash to the node 2 where the file will be made available
* For sample usage of stash and unstash refer below

pipeline {

agent { label 'GOL'}

triggers {

cron('H \* \* \* \*')

pollSCM('\* \* \* \* \*')

}

parameters {

string(name: 'BRANCH', defaultValue: 'master', description: 'Branch to build' )

choice(name: 'GOAL', choices: ['package', 'clean package', 'install'], description: 'maven goals')

}

options {

timeout(time: 1, unit: 'HOURS')

retry(2)

}

environment {

CI\_ENV = 'DEV'

}

stages {

stage('scm') {

environment {

DUMMY = 'FUN'

}

steps {

mail subject: 'BUILD Started '+env.BUILD\_ID, to: 'devops@qt.com', from: 'jenkins@qt.com', body: 'EMPTY BODY'

git branch: "${params.BRANCH}", url: 'https://github.com/asquarezone/game-of-life.git'

//input message: 'Continue to next stage? ', submitter: 'qtaws,qtazure'

echo env.CI\_ENV

echo env.DUMMY

}

}

stage('build') {

steps {

echo env.GIT\_URL

timeout(time:10, unit: 'MINUTES') {

sh "mvn ${params.GOAL}"

}

stash includes: '\*\*/gameoflife.war', name: 'golwar'

}

}

stage('devserver'){

agent { label 'RHEL,'}

steps {

unstash name: 'golwar'

}

}

}

post {

success {

archive '\*\*/gameoflife.war'

junit '\*\*/TEST-\*.xml'

mail subject: 'BUILD Completed Successfully '+env.BUILD\_ID, to: 'devops@qt.com', from: 'jenkins@qt.com', body: 'EMPTY BODY'

}

failure {

mail subject: 'BUILD Failed '+env.BUILD\_ID+'URL is '+env.BUILD\_URL, to: 'devops@qt.com', from: 'jenkins@qt.com', body: 'EMPTY BODY'

}

always {

echo "Finished"

}

changed {

echo "Changed"

}

unstable {

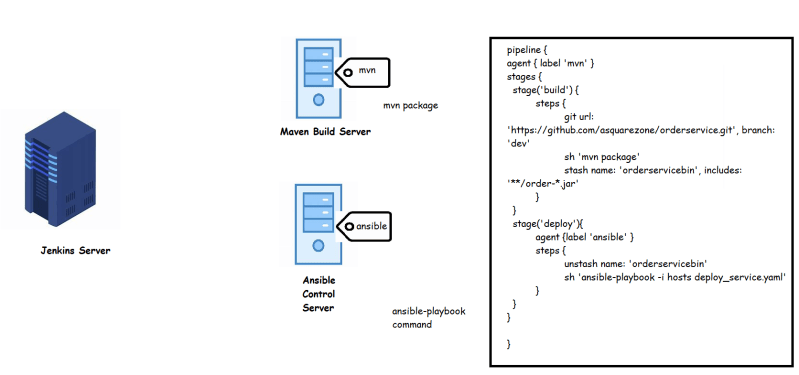
mail subject: 'BUILD Unstable '+env.BUILD\_ID+'URL is '+env.BUILD\_URL, to: 'devops@qt.com', from: 'jenkins@qt.com', body: 'EMPTY BODY'

}

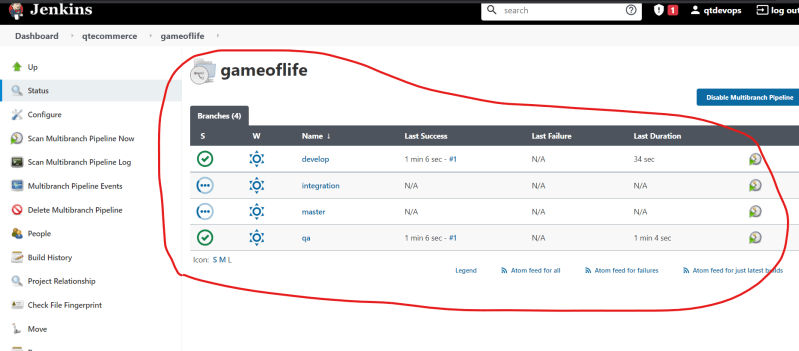
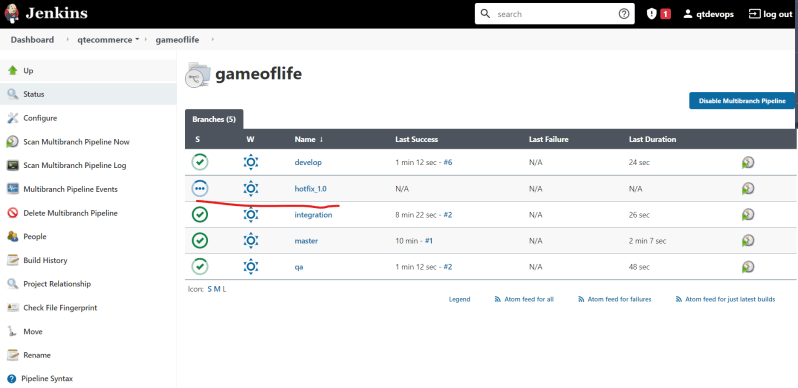
}

}

Scenarios:

Create a CI/CD pipeline where we need to build the application and deploy the application using Ansible 

**Jenkins Project Types**

* Freestyle
* Pipeline Project Type
* Multiconfiguration project type
* Folder
* Multibranch pipeline
* Exercise:
  + Fork the gameoflife project from [Refer Here](https://github.com/asquarezone/game-of-life.git)
  + Now clone the project into your local system
  + Create the following branches in your git repository
    - develop
    - qa
    - integration
  + Now we will have total of 4 branches
  + Lets push all the branches to git repo
  + Now lets create a multi branch project 
  + If we create a new branch then we need to scan multibranch pipeline manually to get the new branch added to build 
  + For every branch we can have different configuration like build triggers, agents etc written in Jenkinsfile.

**Code Quality Scanning**

* SonarQube is an open-source platform for managing code quality in several software areas like
  + Architecture and Design
  + Comments
  + Coding rules
  + Potential Bugs
  + Duplications
  + Unit tests
  + Complexity