How to use Jenkins Pipeline global variable on another stages?

Asked 3 years, 6 months ago Modified 3 years, 6 months ago Viewed 6k times

Part of CI/CD Collective Collective



I have defined global variable in Jenkins pipeline

1

```
def BUILDNRO = '0'
pipeline { ...
```

(1)

Then i manipulate variable with shell script to enable running builds parallel by using job build number as identifier so we don't mix different docker swarms.

i get value out from previos stage and trying to get global variable on next stage

```
stage('DOCKER: Init docker swarm') {
    steps {
        echo "BUILDNRO is: ${BUILDNRO}" --> Value is here.
        sh '''#!/bin/bash
        echo Buildnro is: ${BUILDNRO} --> This is empty.
        ...
}
```

This will out give global variable empty. why? in previous stage there was value in it.

EDIT 1. Modified code blocks to reflect current status.



Sorted by:



3

I managed to figure it out. Here is solution how i managed to did it. BUILDNRO is groovy variable and if wanting to used in bash variable it have to pass using withEnv. BUILD_NUMBER in first stage is bash variable hence it can be used directly script in first stage.

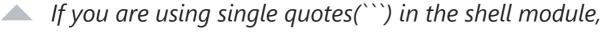


```
def BUILDNRO = '0'
pipeline {
  stages {
    stage('Handle BUILD_NUMBER') {
      steps {
        script {
          BUILDNRO = sh( script: '''#!/bin/bash
          Build='echo ${BUILD_NUMBER} | grep -o '..$''
          ''',returnStdout: true).trim()
        }
      }
    }
    stage('DOCKER: Init docker swarm') {
      steps {
        dir("prose_env/prose_api_dev_env") {
          withEnv(["MYNRO=${BUILDNRO}"]) {
            sh(returnStdout: false, script: '''#!/bin/bash
            echo Buildnro is: ${MYNRO}`
            '''.stripIndent())
        }
     }
   }
 }
}
```

Share Improve this answer Follow

edited Sep 26, 2019 at 10:21







Jenkins treats every variable as a bash variable. The solution



is using double quotes(""") but then if you made bash



variable you have to escape it. Below an example with

working your use case and escaped bash variable

```
pipeline {
    agent any
    stages {
        stage('Handle BUILD_NUMBER') {
            steps {
                script {
```

```
BUILDNRO = sh(script: 'pwd', returnStdout: true).trim()
                    echo "BUILDNRO is: ${BUILDNRO}"
                }
            }
        }
        stage('DOCKER: Init docker swarm') {
            steps {
                sh """#!/bin/bash
            echo Buildnro is: ${BUILDNRO}
            variable=world
            echo "hello \${variable}"
            sh """
            }
        }
    }
}
```

output of the second stage:

```
Buildnro is: /var/lib/jenkins/workspace/stack1 hello world
```

Share Improve this answer Follow

Just curious as \${BUILD_NUMBER} is environment variable which is used in "Handle BUILD_NUMBER" stage. BUILDNRO is global variable which isn't imported inside the sh script. Value is returned from the script to it or is it because even it's defined outside but it's used under script {} part. – Juge Sep 26, 2019 at 4:51

problem with double quotes(""") is that it's harder then do string manipulation etc. – Juge Sep 26, 2019 at 8:54

\${BUILD_NUMBER} is a bash variable. You can check all bash variables provided by Jenkins here: url-to-your-jenkinsl/env-vars.html/ or by passing "env" in your shell script in Jenkins. – rafal1337 Sep 26, 2019 at 9:00

what exact is harder? if you mean escaping bash variables - yes it could be problematic sometimes. Try to use groovy, is most natural for Jenkins. – rafal1337 Sep 26, 2019 at 9:04

Ok. Now that's explains why \${BUILD_NUMBER} is accepted. I stand corrected. Not escaping bash variables but doing shell scripting. using single quotes("") vs. double quotes(""") when doing shell script string manipulation is way easier. i got tons of trouble with double quotes. – Juge Sep 26, 2019 at 10:20