Declarative Jenkins Pipeline; How to declare a variable and use it in sc ript or mail notification?

(update below)

I have a declarative pipeline job which can take an argument VERSION.

```
pipeline {
  parameters {
    string(name: VERSION, defaultValue: '')
  }

  // ...
}
```

If no VERSION is given, like when Gitlab send a hook to this job, I want to compute it from git, so I do something like this

```
stages {
  stage('Prepare') {
    steps {
      // ...
      if (! env.VERSION) {
        VERSION = sh(script: "git describe", returnStdout: true).trim()
      }
    }
  }
}
```

Now I want to "inject" this variable to

- my build script. It needs to find "VERSION" in the environment variables
- to the jenkins mail notificator. And get it to retreive ${VERSION} in subject or body text

I tried changing above code with

```
stages {
  stage('Prepare') {
    steps {
      // ...
      if (! env.VERSION) {
        env.VERSION = sh(script: "git describe", returnStdout: true).trim()
      }
    }
  }
}
```

Got this error `groovy.lang.MissingPropertyException: No such property: VERSION for class: groovy.lang.Binding`

I then tried to add a "environment" step below

```
environment {
    VERSION = ${VERSION}
}
```

but it didn't solve my problem.

I'm looking for any help to solve it.

UPDATE

I now have a working pipeline which looks like

```
pipeline {
    agent any
    parameters {
        string(name: 'VERSION', defaultValue: '')
    }
    environment {
        def VERSION = "${params.VERSION}"
    }
    stages {
        stage('Prepare & Checkout') {
            steps {
                script {
                    if (! env.VERSION) {
                       VERSION = sh(script: "date", returnStdout: true).trim()
                    }
                    echo "** version: ${VERSION} **"
                }
            }
        }
        stage('Build') {
            steps {
                // sh "./build.sh"
                echo "** version2: ${VERSION} **"
            }
        }
    } // stages
    post {
        always {
            mail to: 'foo@example.com',
                 subject: "SUCCESS: ${VERSION}",
                 body: """<html><body><p>SUCCESS</p></body></html>""",
                 mimeType: 'text/html',
                 charset: 'UTF-8'
            deleteDir()
        }
    }
} // pipeline
```

I needed to add the "environment" step to be able to get $VERSION in all Stages (not only in the one it is manipulated).

I still need to find a way to inject this $VERSION variable in the environment variables, so that my build script can find it.

jenkins | environment-variables | jenkins-pipeline

2 Answers

If you want to inject the variable in the environment so that you can use it later, you could define another variable that is equal to env.VERSION or the output of the shell scrip. Then use that variable in your pipeline eg:

```
pipeline {
  parameters {
    string(name: VERSION, defaultValue: '')
  }
```

```
    def version = env.VERSION
    stages {
    stage('Prepare') {
      steps {
        // ...
        if (!version) {
          version = sh(script: "git describe", returnStdout: true).trim()
        }
      }
    }

    mail subject: "$version build succeeded", ...
}
```

If you want other jobs to be able to access the value of VERSION after the build is run, you can write it in a file and archive it.

Edit: In order for your script to be able to use the version variable, you can either make your script take version as a parameter or you can use the withEnv step.

This code works well, but is there anyway I can "inject" my ${VERSION} in environnement variables so that my bash build script can find it? It runs inside a separate stage like this state('Build') { steps { sh "./build.sh" } } – daks May 22 at 8:26

I updated my initial question – daks May 22 at 8:47

You can use the withEnv step to wrap your script call, or make version a param to the script. withEnv doc: jenkins.io/doc/pipeline/steps/workflow-basic-steps/… – Vasiliki Siakka May 22 at 21:01

It works. Not ideal because I need to specify withEnv explicitely where I need it but i'll do the job, thanks :) – daks May 23 at 8:22

Assuming you are using Parametrized pipelines, you should call variable as `${params.parameterName}`

> Although parameters are available in env they currently are created before the first time the pipeline is run, therefore you should access them via params:

In your case:

```
${params.VERSION}
```

I wasn't aware of this potential problem with params/env but I'm not sure it helps to solve my problem (except I changed my IF statement to check in params not env) because what I want is inject a new envvar (or simply a variable) and use it later. – daks May 21 at 14:51

"inject" or "update" – daks May 21 at 15:06