

Jenkins: Cannot define variable in pipeline stage

I'm trying to create a declarative Jenkins pipeline script but having issues with simple variable declaration.

Here is my script:

```
pipeline {
  agent none
  stages {
    stage("first") {
      def foo = "foo" // fails with "WorkflowScript: 5: Expected a step @ line 5, column 13."
      sh "echo ${foo}"
    }
  }
}
```

However, I get this error:

```
org.codehaus.groovy.control.MultipleCompilationErrorsException: startup failed:
WorkflowScript: 5: Expected a step @ line 5, column 13.
      def foo = "foo"
      ^
```

I'm on Jenkins 2.7.4 and Pipeline 2.4.

jenkins

groovy

jenkins-pipeline

The Declarative model for Jenkins Pipelines has a restricted subset of syntax that it allows in the `stage` blocks - see the syntax guide for more info. You can bypass that restriction by wrapping your steps in a `script { ... }` block, but as a result, you'll lose validation of syntax, parameters, etc within the `script` block.

Thanks for this. Doc is located here: jenkins.io/doc/book/pipeline/syntax/#declarative-steps – Japster24 Feb 7 '17 at 21:11

I think error is not coming from the specified line but from the first 3 lines. Try this instead :

```
node {
  stage("first") {
    def foo = "foo"
    sh "echo ${foo}"
  }
}
```

I think you had some extra lines that are not valid...

EDIT

From declarative pipeline model documentation, it seems that you have to use an `environment` declaration block to declare your variables, e.g.:

```
pipeline {
  environment {
    FOO = "foo"
  }

  agent none
  stages {
    stage("first") {
      sh "echo ${FOO}"
    }
  }
}
```

- He's using the new declarative pipeline model syntax, that's why it's different to what we know ;-) – StephenKing Oct 3 '16 at 16:01
- 1 Oops sorry didn't see that new declarative pipeline model, I edited my post accordingly ! – Pom12 Oct 4 '16 at 7:45

Agree with @Pom12, @abayer. To complete the answer you need to add script block

Try somethign like this:

```
pipeline {
  agent any
  environment {
    ENV_NAME = "${env.BRANCH_NAME}"
  }

  // -----

  stages {
    stage('Build Container') {
      steps {
```

```
    echo 'Building Container..'

    script {
        if (ENVIRONMENT_NAME == 'development') {
            ENV_NAME = 'Development'
        } else if (ENVIRONMENT_NAME == 'release') {
            ENV_NAME = 'Production'
        }
    }
    echo 'Building Branch: ' + env.BRANCH_NAME
    echo 'Build Number: ' + env.BUILD_NUMBER
    echo 'Building Environment: ' + ENV_NAME

    echo "Running your service with environemnt ${ENV_NAME} now"
}
}
```

Notice that this example assumes that there is already a defined environment variable "ENVIRONMENT_NAME" accessible to jenkins. – Alberto Nov 8 '17 at 10:36