

???????

```
node('docker-agent-01') {
    stage ('Pulling repo and filling constants'){
        sh "rm *"
        git 'https://gitverse.ru/bobrobot/liteconnect_docker.git'
        withEnv(readFile('env.txt').split('\n') as List) {
            sh 'chmod +x replace_env.sh && ./replace_env.sh'
        }
        sh "cp env.txt .env"
    }
}
```

```
node() {
    stage('first') {
        withEnv(["PATH+HUIWAM=${HOME}/.local/bin"])
        {
            sh "printenv PATH"
        }
        sh "printenv PATH"
    }
}
```

```
withEnv([
    "PATH+EXTRA=/home/jenkins/.local/bin"])
{
    node() {
        stage('first') {
            sh "printenv PATH"
        }
    }
}
```

```
pipeline {
    agent {
```

```

        kubernetes (kubernetesAgent(name: 'mini'))
    }
    environment {
        PATH="${HOME}/.local/bin:${PATH}"
    }
    stages {
        stage('Integrate Remote k8s with Jenkins ') {
            steps {
                sh "printenv PATH"
                withCredentials([file(credentialsId: 'mysrc', variable: 'MYFILE')]) {
                    sh 'cp $MYFILE new.sh'
                }
            }
        }
    }
}
}

```

```

// для груви скрипта нужно упаковывать в withenv чтобы были видны определенные в скрипте
//параметры,
//параметры из jenkins видны
//param_in_main - текстовый параметр,
//build_type - второй параметр.
withEnv([
    'ENV1=It work ' + param_in_main,
    'ENV2=ooo',
    'missingfname=thefilereallymissing.txt'])
{
    node('slave') {
        stage('first') {
            sh "mkdir ~/bin"
            sh "printenv PATH"
            sh "echo $param_in_main" //обращение к параметру из параметров в jenkins
            sh "echo $ENV1" //обращение к составному параметру, определенному внутри скрипта
            sh "echo $ENV2" //обращение к независимому параметру внутри скрипта
        }

        stage('check type of pipeline'){
            if (build_type == 'prod') { //если проверяем вне sh - доллар не ставится
                sh "echo 'This is a prod'"
            }
        }
    }
}

```

```
}

stage('test exeptions'){
  try {
    sh 'rm $missingfname'
  }
  catch (exc) {
    echo 'Попытались удалить несуществующий файл ' + missingfname
  }
}

stage('check cycle'){
  sh 'mkdir "firstdir"'
  def fileNames = sh(returnStdout: true, script: 'ls').trim().split()
  for (item in fileNames) {
    echo "Processing item: ${item}"
  }
}
}
}
```

```
def buildPod = {
  def podYaml = """
kind: Pod
metadata:
  name: jenkins-agent
spec:
  containers:
  - name: my-jenkins-agent
    image: jenkins/inbound-agent:latest
    command:
    - cat
    tty: true
    volumeMounts:
    - mountPath: /var/run/docker.sock
      name: docker-sock
  volumes:
  - name: docker-sock
    hostPath:
      path: /var/run/docker.sock
```

```

"""
    return podYaml
}

def label = "jenkins-agent-${UUID.randomUUID().toString()}"

podTemplate(label: label, yaml: buildPod()) {
    node(label) {
        stage('Build') {
            container('my-jenkins-agent') {
                echo 'Start stage build'
                sh 'ls -al'
                echo 'End stage build'
            }
        }
        stage('Deploy') {
            container('my-jenkins-agent') {
                echo 'Start stage deploy'
                try {
                    sh "rm my.txt"
                }
                catch(exc) {
                    echo "The ERROR!"
                    echo "${exc}"
                }
            }
        }
    }
}
}

```

Revision #3

Created 2 March 2025 13:28:43 by Admin

Updated 16 March 2025 15:57:57 by Admin