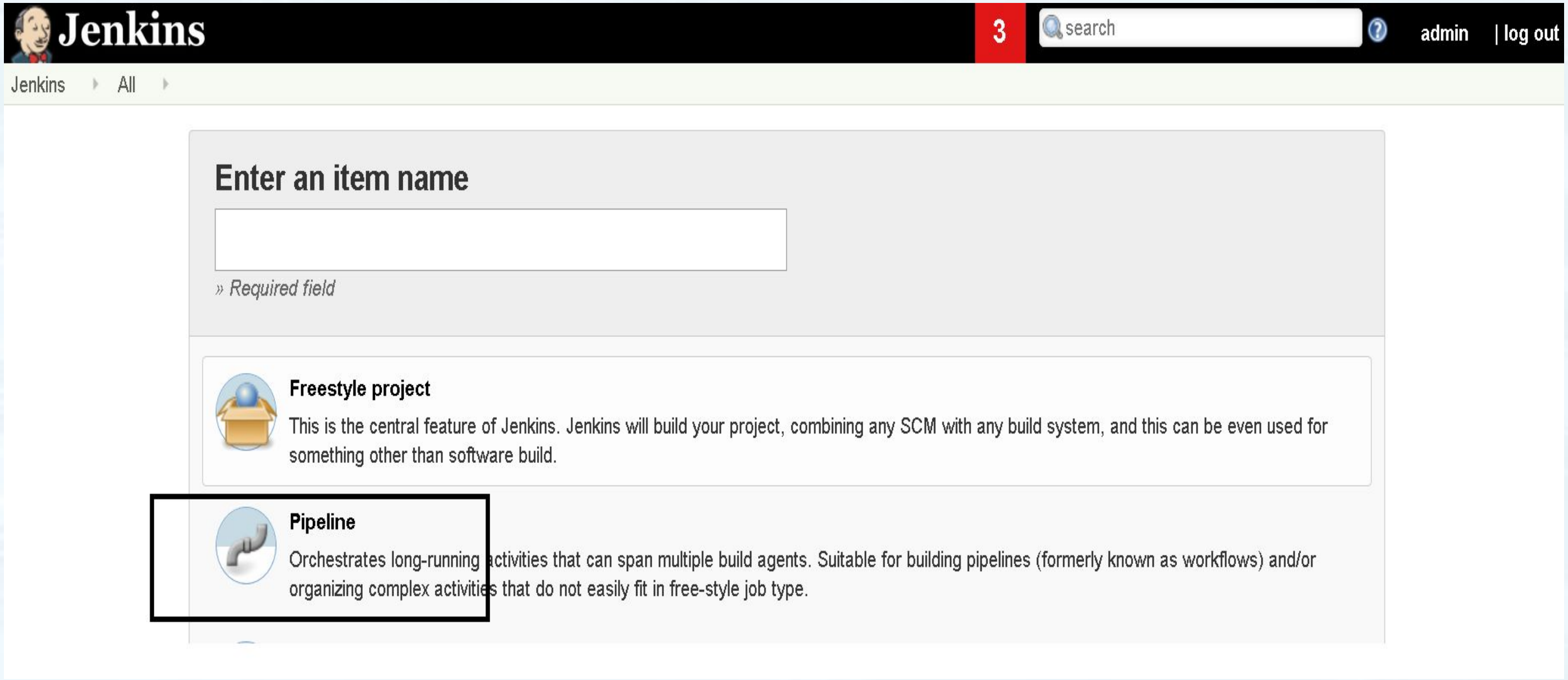
Create Pipeline Job

- Click on new item & select pipeline from available options
- 

Enter an item name

» Required field

Freestyle project

This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

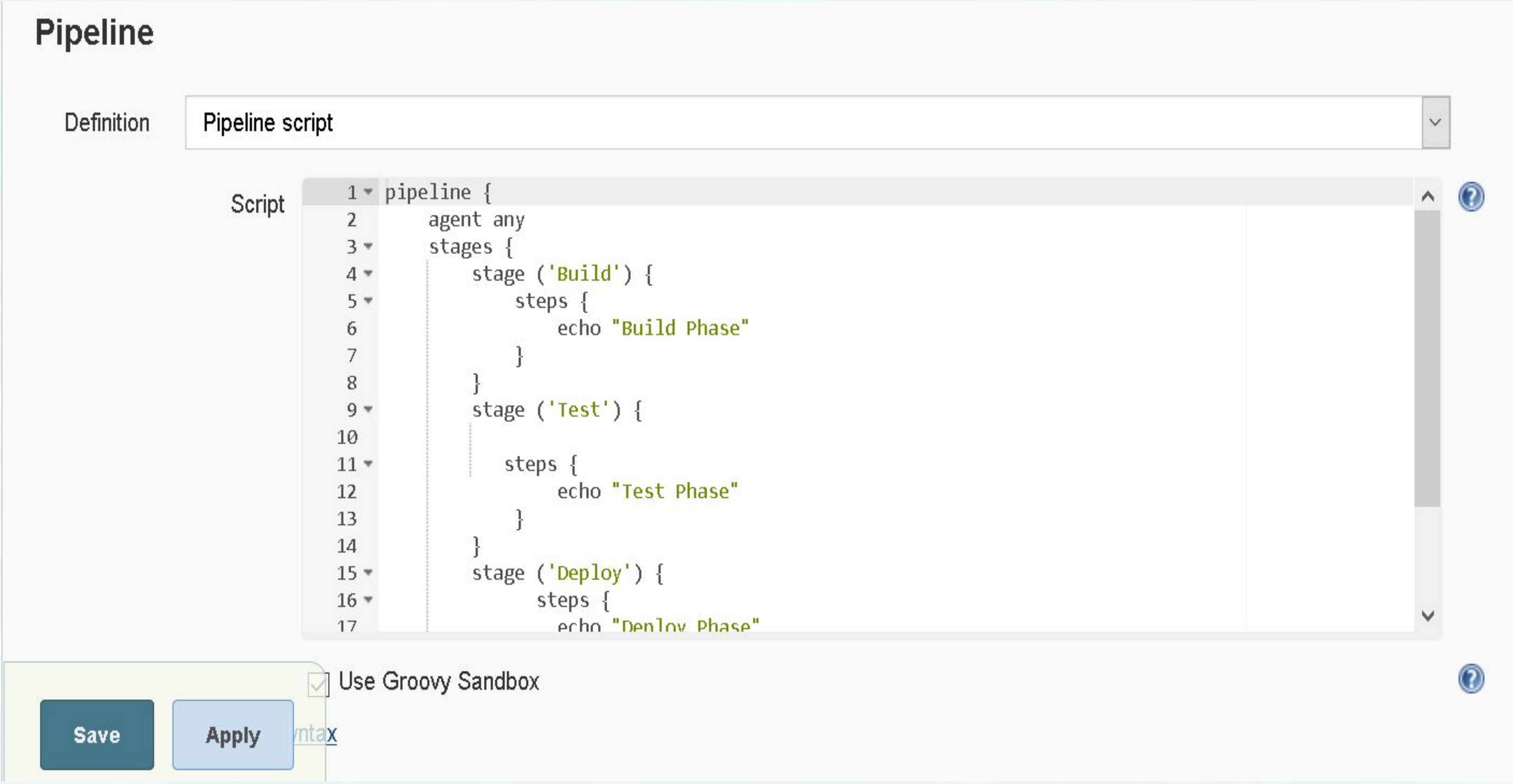
Pipeline

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Types of Pipeline

- 1) **Declarative** : Script is written within the pipeline box of jenkins box.
- 2) **Scripted** : Pipeline job is written in Jenkinsfile, which is committed to github or other vcs.
- Pipeline supports groovy language

Declarative Pipeline



Pipeline

Definition: Pipeline script

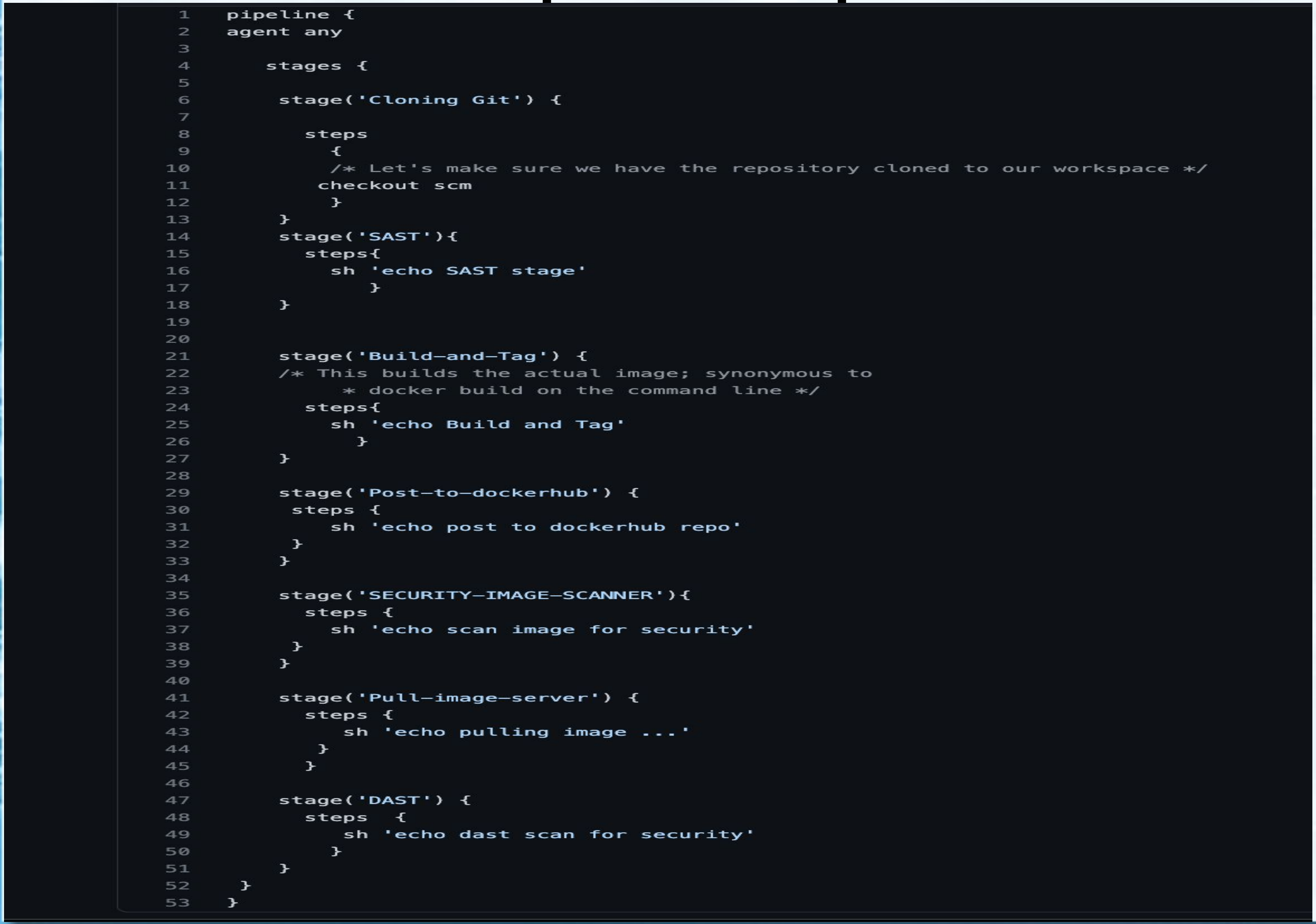
Script:

```
pipeline {
    agent any
    stages {
        stage ('Build') {
            steps {
                echo "Build Phase"
            }
        }
        stage ('Test') {

           steps {
                echo "Test Phase"
            }
        }
        stage ('Deploy') {
              steps {
                echo "Deploy Phase"
```

☑ Use Groovy Sandbox

Scripted Pipeline



```
pipeline {
agent any

   stages {

    stage('Cloning Git') {

      steps
        {
        /* Let's make sure we have the repository cloned to our workspace */
       checkout scm
        }
    }
    stage('SAST'){
      steps{
        sh 'echo SAST stage'
           }
    }


    stage('Build-and-Tag') {
    /* This builds the actual image; synonymous to
         * docker build on the command line */
      steps{
        sh 'echo Build and Tag'
          }
    }

    stage('Post-to-dockerhub') {
     steps {
        sh 'echo post to dockerhub repo'
     }
    }

    stage('SECURITY-IMAGE-SCANNER'){
      steps {
        sh 'echo scan image for security'
     }
    }

    stage('Pull-image-server') {
      steps {
         sh 'echo pulling image ...'
       }
      }

    stage('DAST') {
      steps  {
         sh 'echo dast scan for security'
        }
    }
 }
}
```