Using S4 with Jenkins

To set up Jenkins to work with S4, please follow these steps:

1. For Jenkins and S4 to work together properly, please make sure you install the following plugins:

    a. Pipeline plugin
    b. Pipeline Utility Steps plugin
    c. (optional for SSH support) SSH Build Agents plugin

2. To double check which plugins you have installed. On the Jenkins dashboard, navigate to Manage Jenkins => Manage Plugins => Installed

3. If you are missing a plugin, you can search in the available plugins tab, and install directly to your Jenkins instance

To set up a pipeline that runs S4 scan during Jenkins Build Process, please follow these steps:

1. On Jenkins Dashboard, click New Item. Then click on pipeline, and enter a name for your new pipeline.

2. You should now be seeing the pipeline configuration panel, in here you can choose to add different plugins to your pipeline to help with the build process

3. At the bottom of the configuration panel, there is a script box at the bottom of the page, in here, a user is able to set up different build stages for their pipeline build process. In the script area, please copy and paste the following:

    - If using Linux or Mac, please make sure to adjust any “bat” commands in the code below to “sh” commands

    - If using a Jenkinsfile, please copy and paste the stages Checkout Stage your build process, and the always section into your post section

    - If using Linux or Mac, please make sure to adjust bat 'del "..\\S4.zip"' into `sh ‘rm “../S4.zip”‘`

NOTE: The following pipeline script is for Windows machines running Jenkins:

```
pipeline {
    agent any
    stages {
        stage('Build'){
            steps{
                echo 'Building...'
            }
        }
        stage('Test'){
            steps{
                echo 'Testing...'
            }
        }
        stage(‘S4 Security Scan’) {
            steps{
                //For git, will need to add credentials if repo is not public
                //Can also use SSH url to checkout from branch
                //“ssh://git@github.com:YourRepoName/yourepo.git”
                git branch: '<branch_name>', credentialsId: 'Github', url:
'https://github.com/YourRepoName/yourepo.git'

                echo 'ZipFileing..'
                // Relative zip location, make outside the workspace directory to prevent infinite loops
                zip zipFile: '../S4.zip', dir:''
                echo 'S4ing..'
                //Sends Test.zip to S4 for scanning
                //If running on linux or mac, replace win below with version specific
                bat "cd S4cli && S4-win -user S4Username -pass S4Password -file ../../S4.zip -orgid
s4orgId -s4url https://s4.digitsec.com"
                bat "cd S4cli && python parseResponse.py"
            }
        }
        stage('Deploy') {
            steps {
                echo 'Deploying....'
            }
        }
    }
    post {
        always {
            //Deleted the test.zip file that was created, to prevent future build failures
            echo 'LeaveNoTraceing....'
            bat 'del "..\\S4.zip"'
        }
    }
}
```

NOTE: The following pipeline script is for Linux machines running Jenkins:

```
pipeline {
    agent any
    stages {
        stage('Build'){
            steps{
                echo 'Building...'
            }
        }
        stage('Test'){
            steps{
                echo 'Testing...'
            }
        }
        stage('S4 Security Scan') {
            steps{
                //Can also use SSH url to checkout from branch //“ssh://git@github.com:YourRepoName/YourRepo.git”
                git branch: 'master', credentialsId:'githubs4' , url: 'https://github.com/yourreponame/YourRepo.git'
                echo 'ZipFileing..'
                zip zipFile: '../S4.zip', dir:''
                // Relative zip location, make outside the workspace directory to prevent infinite loops zip zipFile: '../S4.zip', dir:''
                echo 'S4ing..'
                //Sends Test.zip to S4 for scanning
                sh "cd S4cli && chmod +x s4-linux && ./s4-linux -user s4Username -pass s4Password -file ../../S4.zip -orgid s4orgId -s4url https://s4.digitsec.com"
                sh "cd S4cli && python parseResponse.py"
            }
        }
        stage('Deploy') {
            steps {
                echo 'Deploying....'
            }
        }
    }
    post {
        always {
            //Deleted the test.zip file that was created, to prevent future build failures echo 'LeaveNoTraceing....'
            sh 'rm "../S4.zip"'
        }
    }
}
```

4. Once this has been completed, please click the save button.

5. Next, we need to add the S4 CLI to our github repo that we are pulling in this pipeline.

6. To get the CLI, please log into s4.digitsec.com, go to the configure S4 tab, click on integrations, and click the download button next to “Connect S4 with Jenkins”. This will download a zipped file containing the CLI tool to help integrate with Jenkins.

7. Next, unzip the file that was downloaded. You should end up with a folder named S4Cli, you will put this whole folder into the root of your github repo.

8. If you want to adjust the parameters for when a build should fail, feel free to edit the python script that comes bundled with the CLI tool. Initially, the python script will fail a build if there are any findings in your scan report. You can adjust the operands to change how they interact with each bug severity. (ie: “if bugsFound == 0:” can be adjusted to “if bugsFound >= 5:”. Making this change would allow the build to pass if there are less than 5 bugs of that particular severity.)

9. Now, you should be able to go to Jenkins Dashboard and click build now. With the above code, it should start the build => download the github repository => zip your github repository => send the zipped file to S4 for scanning => then delete all traces of us while responding with a JSON file of the scan results

F.A.Q.s

Q: I already have a Jenkins Pipeline set up. How can I use this to scan my repo during the build process?

A: Please copy and paste the checkout stage and the always block, and add those to your Jenkins pipeline configuration, or add them to your Jenkinsfile

Q: Why use pipelines over the regular build process?

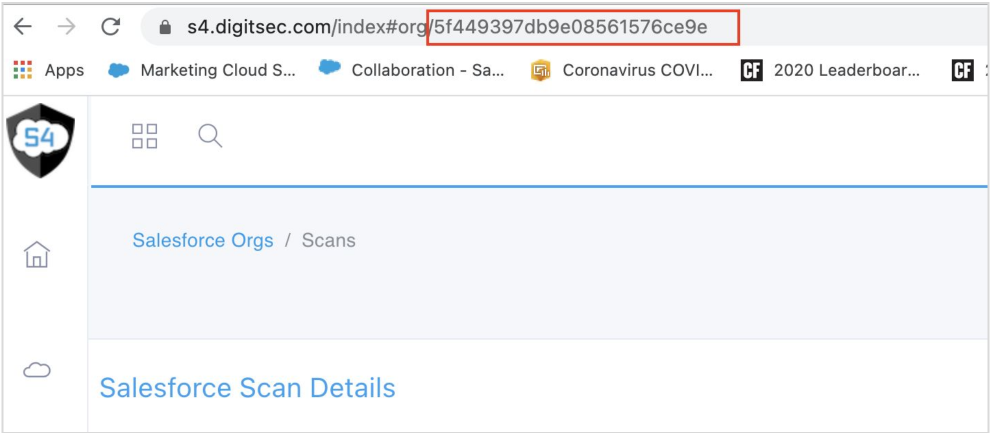
A: Pipelines end up giving a moderator a lot more control over the build process, by using the pipelines you are also able to set up your project to work in stages. By working in stages, you are able to restart your build from a failed stage instead of having to restart the entire build process every time. In addition, the pipelines allow you to code your build process which allows you to have more control over the entire process.

Q: Does S4 plan on creating a plugin for Jenkins?

A: With security being our top priority, we wanted to be able to offer S4 to any users who are using Jenkins to build their projects. By using the CLI tool we are able to have S4 available immediately. We still have a plugin coming in the future, but for the time being, this seems to be the best option for running S4 with Jenkins currently

Q: How to get s4orgId?

A: Once you've registered a placeholder org id using oAuth on S4 using https://s4.digitsec.com. You can view the s4orgId by clicking on **S4 Home** > **View Scans** on the org and getting the s4orgId from the url as shown below:



s4.digitsec.com/index#org/5f449397db9e08561576ce9e

Q: How can I adjust the rules for when a build should fail from an S4 scan?

A: If you look in the S4 CLI folder you placed in the root directory of your repository, there is a file called parseResponse.py. In here you can make adjustments to how your scan results will interact with your build process.

Q: I want to be able to use my username and password to login to github instead of an SSH URL or accessing a public repository, how would I go about doing that?

A: To be able to add credentials to jenkins to use globally within jenkins files or pipelines, please follow the steps in this document. Although, if using this method, please keep in mind that github has said this method of authentication is deprecated, and will eventually stop working.
https://docs.cloudbees.com/docs/cloudbees-ci/latest/cloud-secure-guide/injecting-secrets

Q: I want to be able to have my findings posted to a Jira board, when I run a scan on S4. Is there a way to accomplish this?

A: By setting up a Jira integration through the S4 website, you will be able to have a scan run against your org in your Jenkins Pipeline. Once the scan is completed, (as long as jira is set up) your new findings will be posted to Jira, and the issues that have been resolved will be moved to the done column on Jira.

Q: When I run a scan through Jenkins, sometimes it will say I have too many bugs and fail a build, but the findings says I have 0 findings.

A: If you run into this issue, it would seem that S4 is currently down. We try not to make a habit of being down, but to prevent potentially harmful builds from seeing the light of day, all builds will fail if S4 happens to be down at the time of running a scan. Please reach out to someone at digitsec, and we will make sure to get things up and running again as soon as possible.

Q: I keep trying to build my org, but it fails at the python script every time. How can I resolve this issue?

A: The way the python script is set up out of the box, it will fail the build if the scan finds any issues (low, medium, high, or critical). Feel free to edit the python script to adjust for your own bug policies.