

## Lesson 09 Demo 02

# Integrating SCA Tools into Jenkins for Enhanced Vulnerability Detection

**Objective:** To automate SCA scans by integrating the Snyk plugin with Jenkins, enhancing the efficiency of vulnerability detection within Jenkins build jobs

**Tools required:** Jenkins, Snyk Plugin

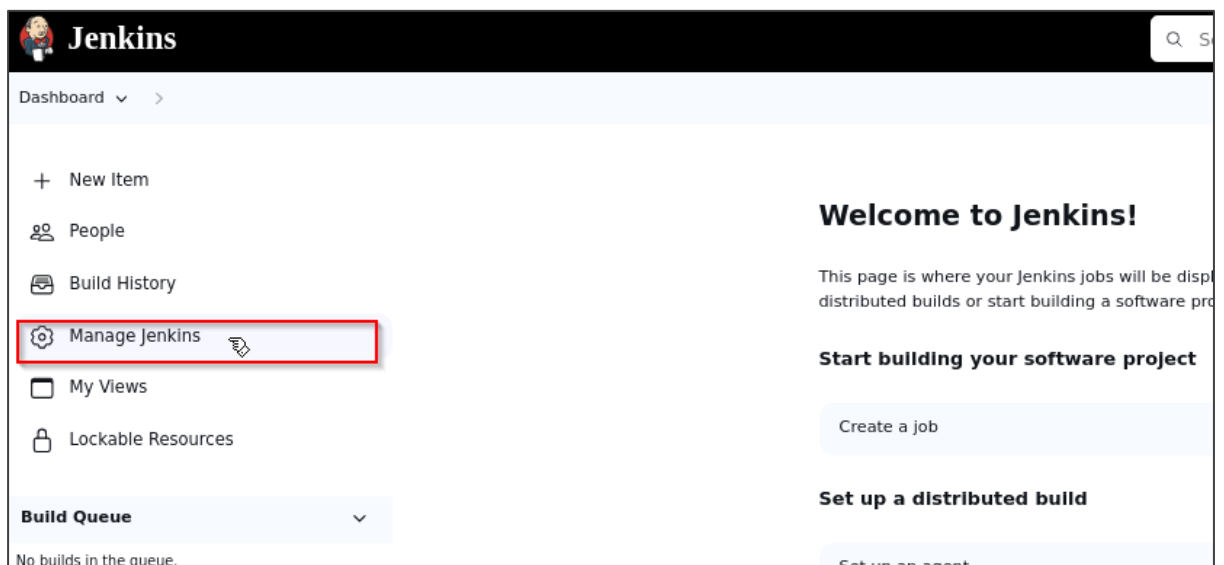
**Prerequisites:** Basic knowledge of Jenkins and Snyk

Steps to be followed:

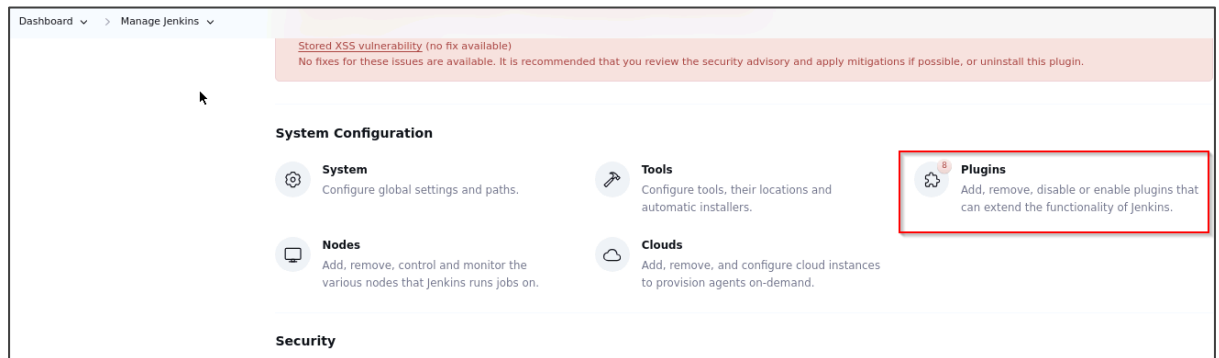
1. Install Snyk and Maven plugins in the Jenkins
2. Configure the Maven and Snyk installations
3. Create a new Jenkins pipeline job

### Step 1: Install Snyk and Maven plugins in the Jenkins

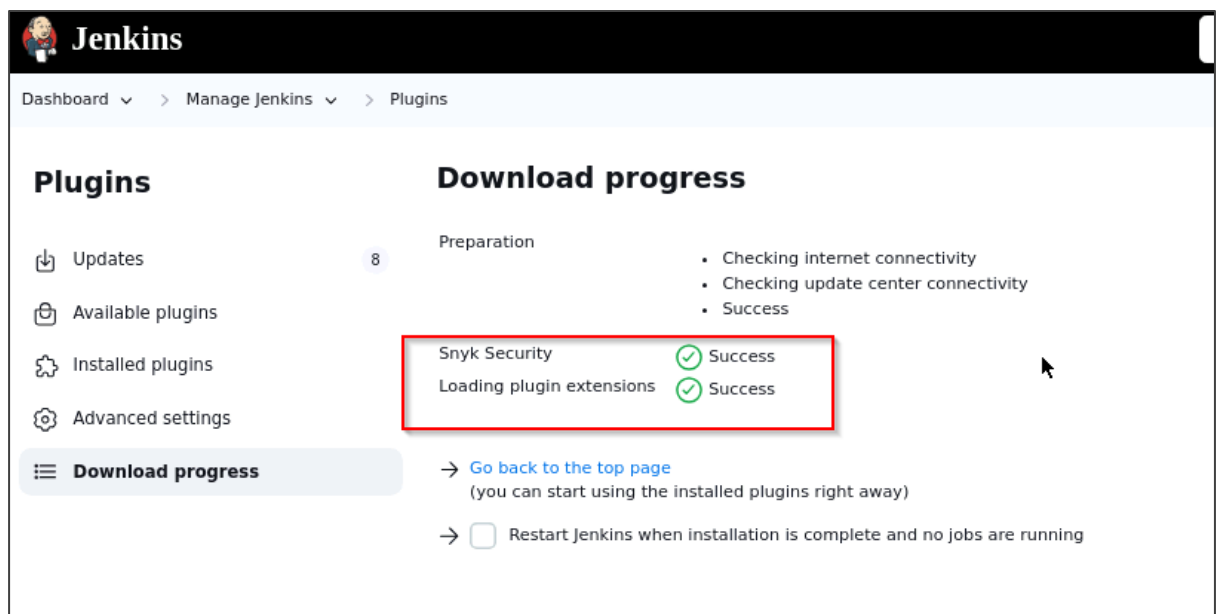
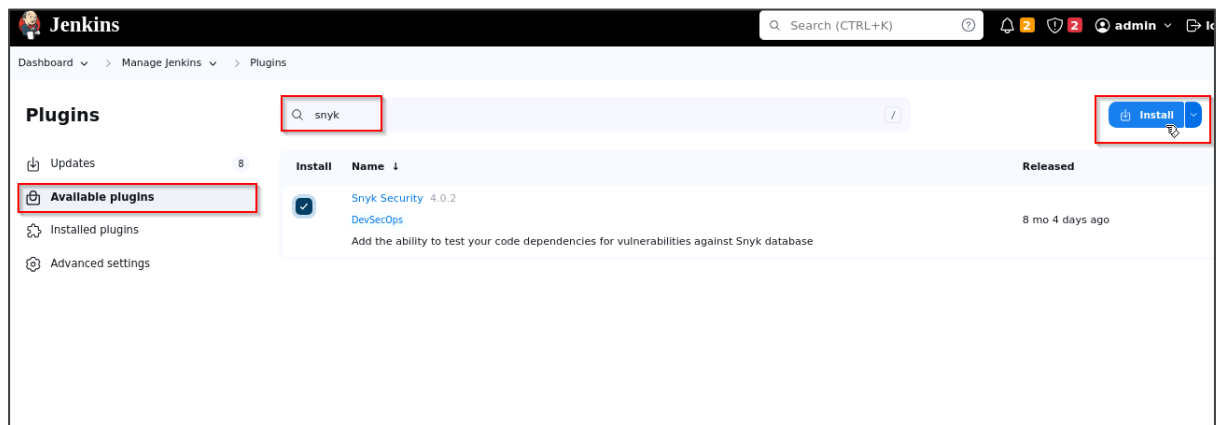
1.1 Log in to your Jenkins dashboard and click on **Manage Jenkins**



## 1.2 Navigate to **System Configuration** and click on **Plugins**

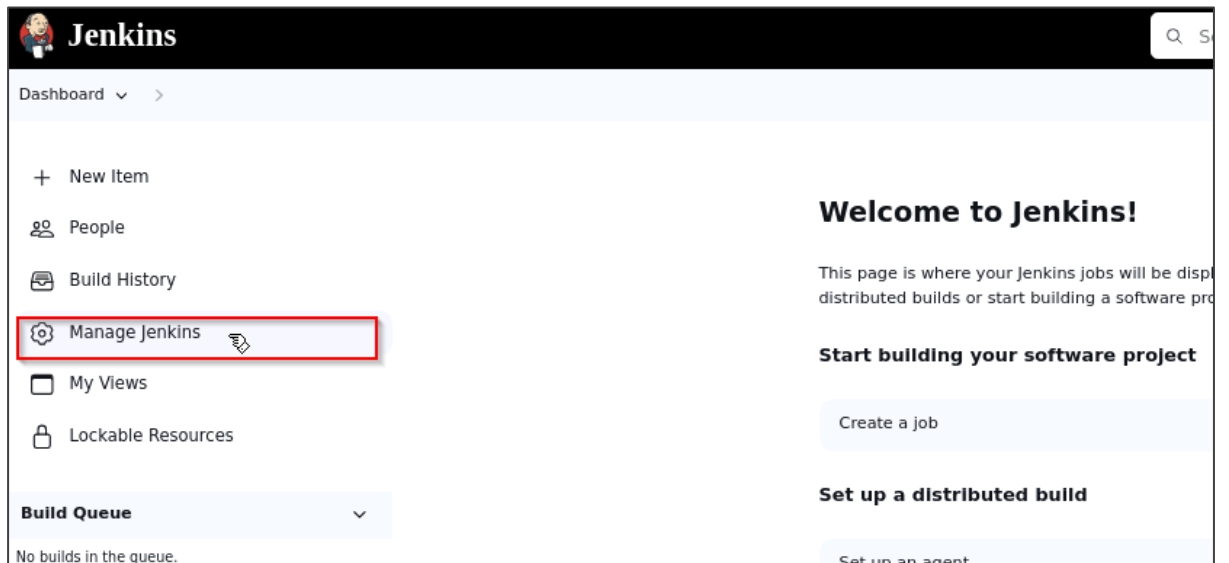


## 1.3 In the **Available Plugins** section, enter **snyk** in the search bar, select the **Snyk Security** plugin, and click **Install**

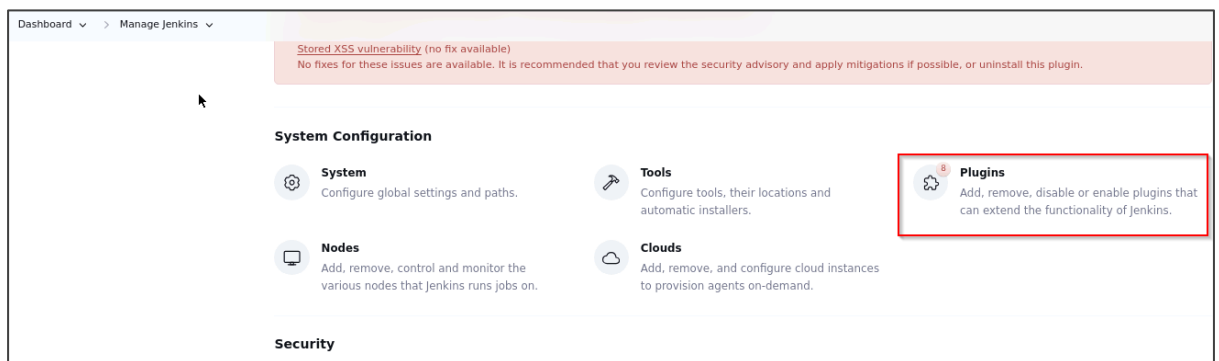


The **Snyk Security** plugin is successfully installed.

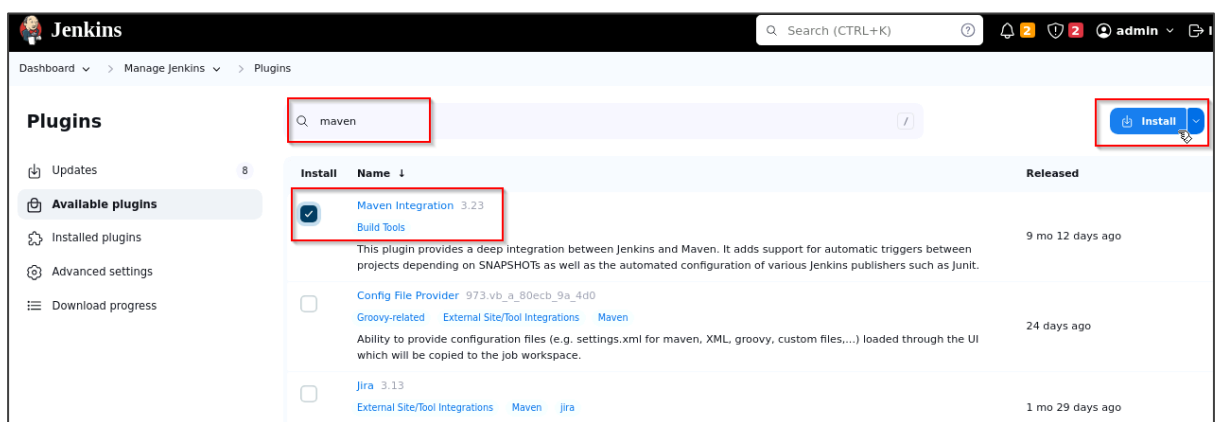
## 1.4 Go to **Manage Jenkins** in the Jenkins dashboard

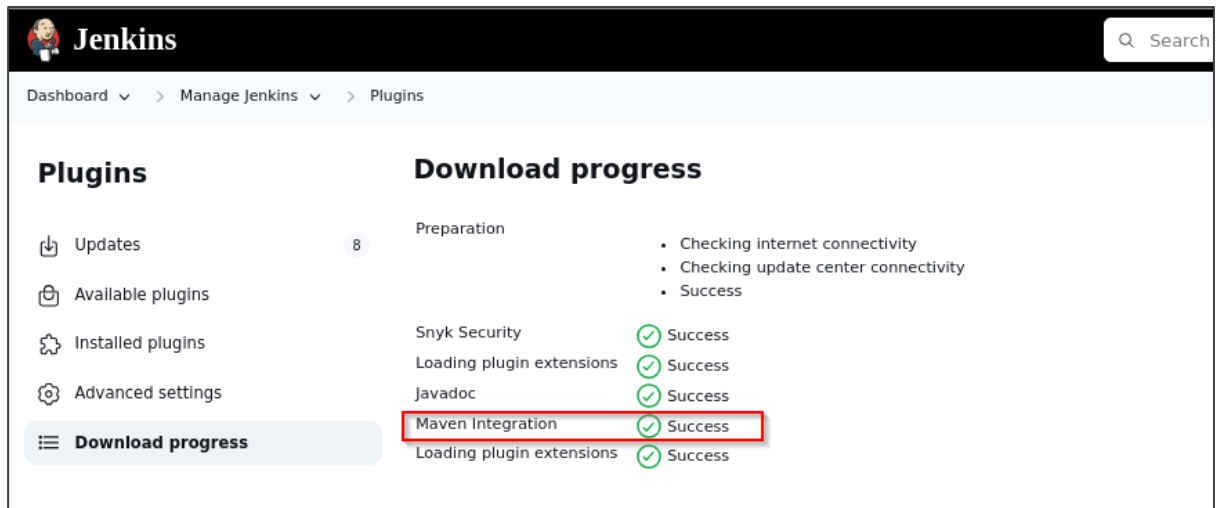


## 1.5 Navigate to **System Configuration** and click on **Plugins**



## 1.6 In the **Available Plugins** section, enter **maven** in the search bar, select the **Maven Integration** plugin, and click **Install**





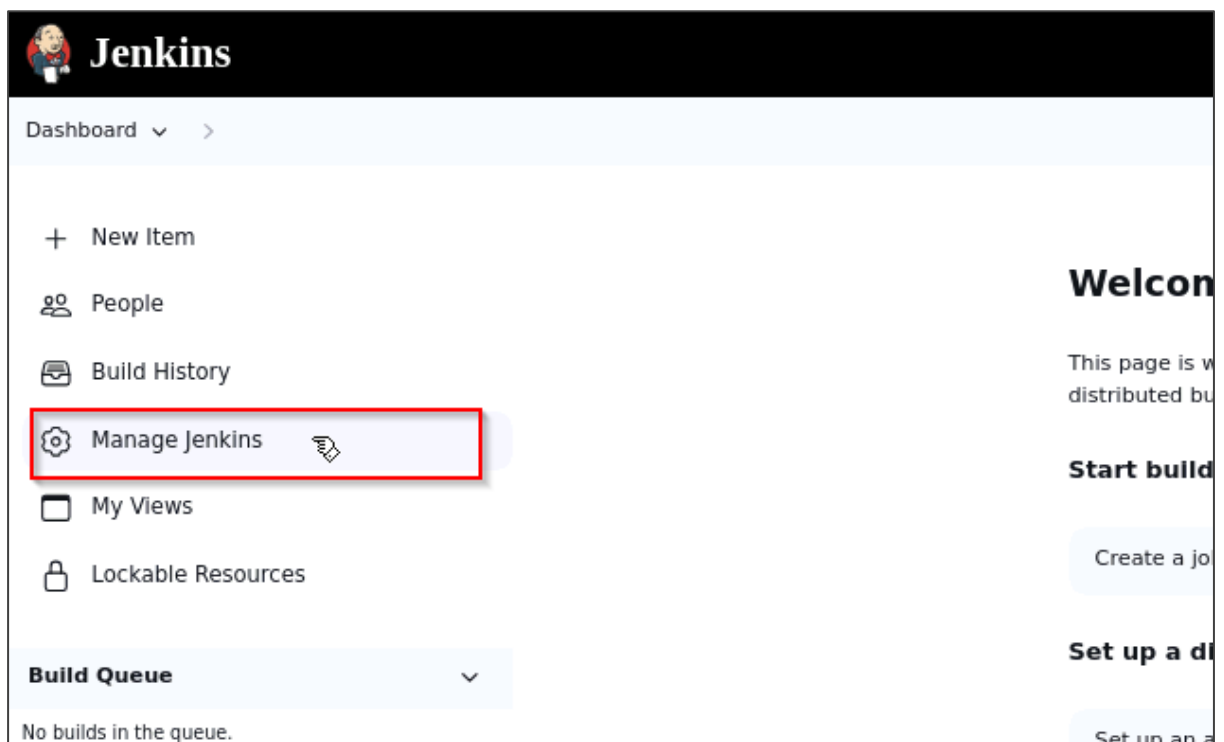
The screenshot shows the Jenkins 'Plugins' page with the 'Download progress' tab selected. The left sidebar contains links for Updates, Available plugins, Installed plugins, Advanced settings, and Download progress. The main content area shows the progress of downloading plugins. The 'Maven Integration' plugin is highlighted with a red box, showing a 'Success' status. Other plugins like 'Snyk Security', 'Loading plugin extensions', and 'Javadoc' also show 'Success' status.

Plugin	Status
Snyk Security	Success
Loading plugin extensions	Success
Javadoc	Success
Maven Integration	Success
Loading plugin extensions	Success

The **Maven Integration** plugin is successfully installed.

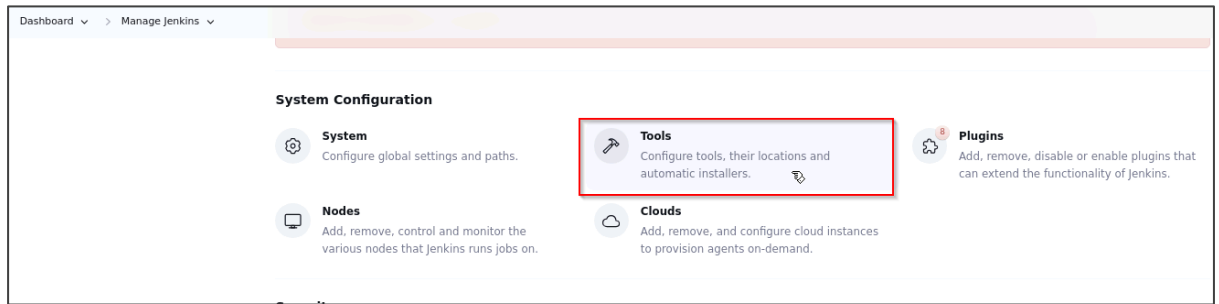
## Step 2: Configure the Maven and Snyk installations

2.1 Go to the Jenkins dashboard and click on **Manage Jenkins**

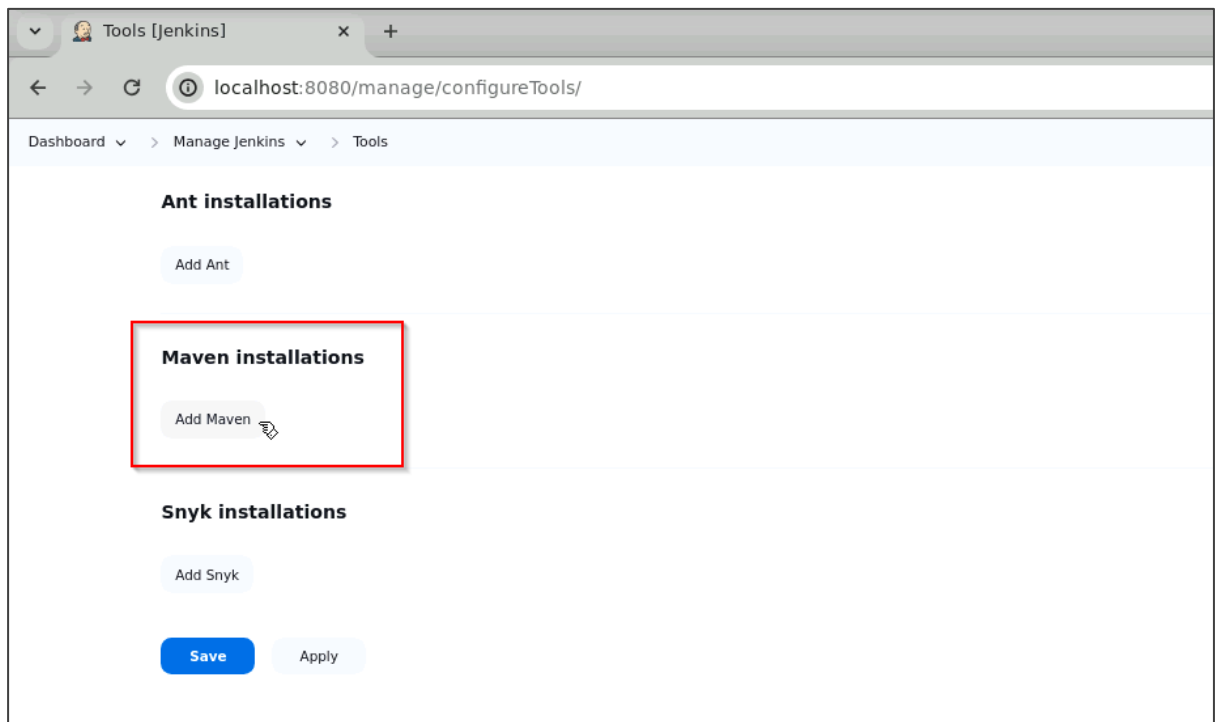


The screenshot shows the Jenkins dashboard. The left sidebar contains links for New Item, People, Build History, Manage Jenkins, My Views, and Lockable Resources. The 'Manage Jenkins' link is highlighted with a red box. The right sidebar contains a 'Welcome' message and a 'Start build' button.

## 2.2 Navigate to **System Configuration** and click on **Tools**



## 2.3 Under **Maven Installations**, click on **Add Maven**



2.4 Type **Maven** in the **Name** field and then click on **Save**

Dashboard > Manage Jenkins > Tools

### Maven installations

Add Maven

≡ **Maven**

Name

Maven

☒ Install automatically ?

≡ **Install from Apache**

Version

3.9.6

Add Installer ▾

Save

Apply

2.5 Under **Snyk Installations**, click on **Add Snyk**

Dashboard > Manage Jenkins > Tools

### Ant installations

Add Ant

### Maven installations

Maven installations ▾ Edited

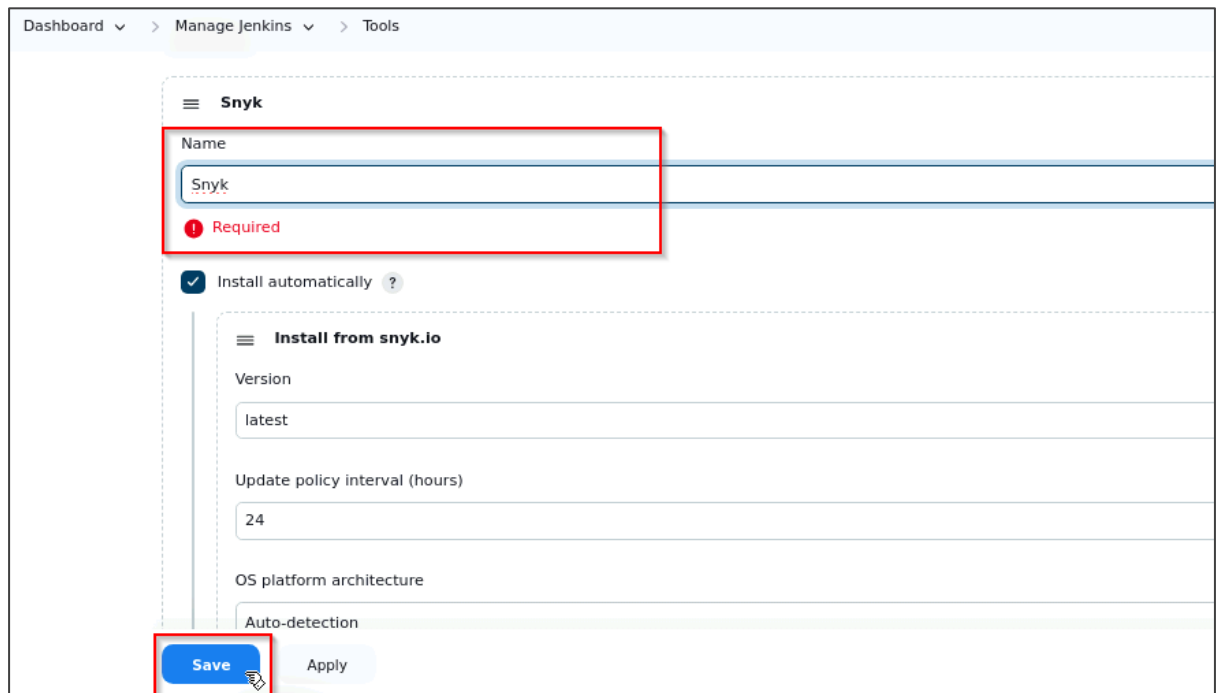
### Snyk installations

Add Snyk

Save

Apply

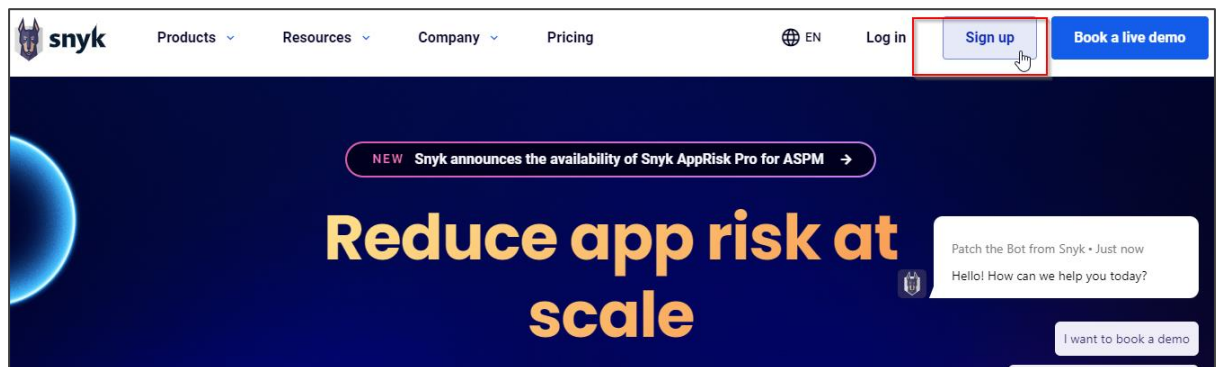
2.6 Type **Snyk** in the **Name** field and then click on **Save**



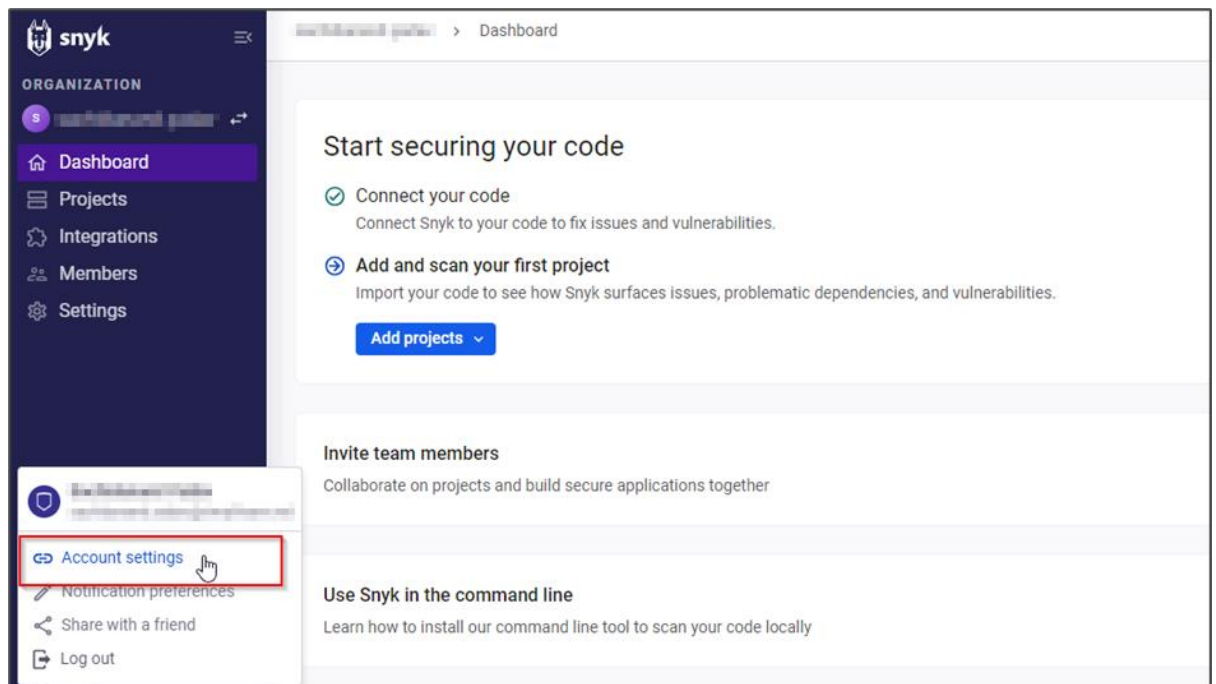
The screenshot shows the Jenkins 'Tools' configuration page for 'Snyk'. The breadcrumb navigation at the top reads 'Dashboard > Manage Jenkins > Tools'. The main section is titled 'Snyk'. It contains a 'Name' field with the value 'Snyk', which is highlighted by a red rectangle. Below the field is a red error message 'Required'. There is a checked checkbox for 'Install automatically'. Below this is a section titled 'Install from snyk.io' with fields for 'Version' (set to 'latest'), 'Update policy interval (hours)' (set to '24'), 'OS platform architecture' (set to 'Auto-detection'), and 'Auto-detection'. At the bottom, there are two buttons: 'Save' (highlighted with a red rectangle) and 'Apply'.

Now you need to get the API token from Snyk, so follow the below steps for getting the API token.

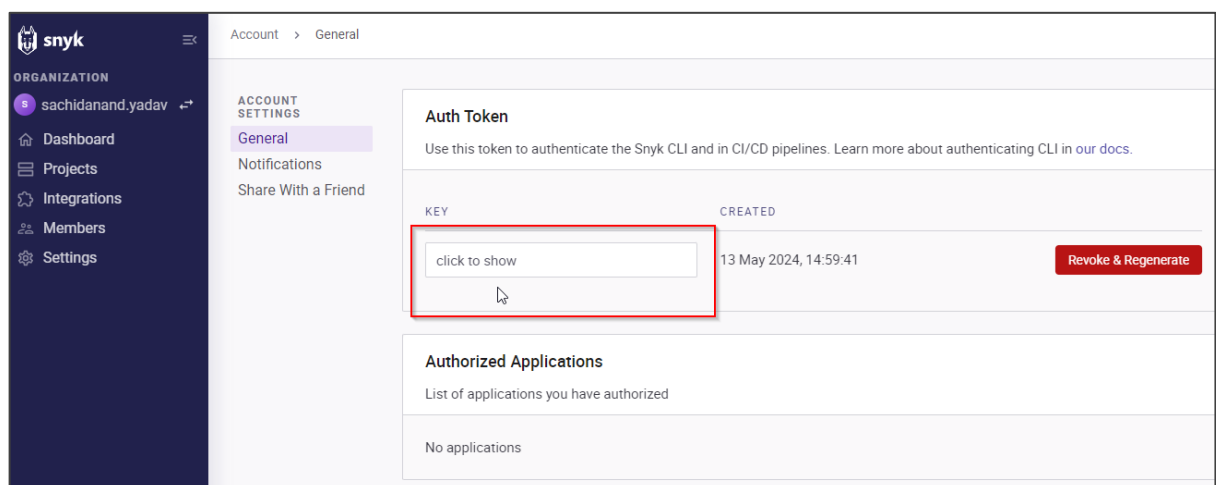
2.7 Visit <https://snyk.io/> and create a new Snyk account



## 2.8 Click on the **Account settings** option

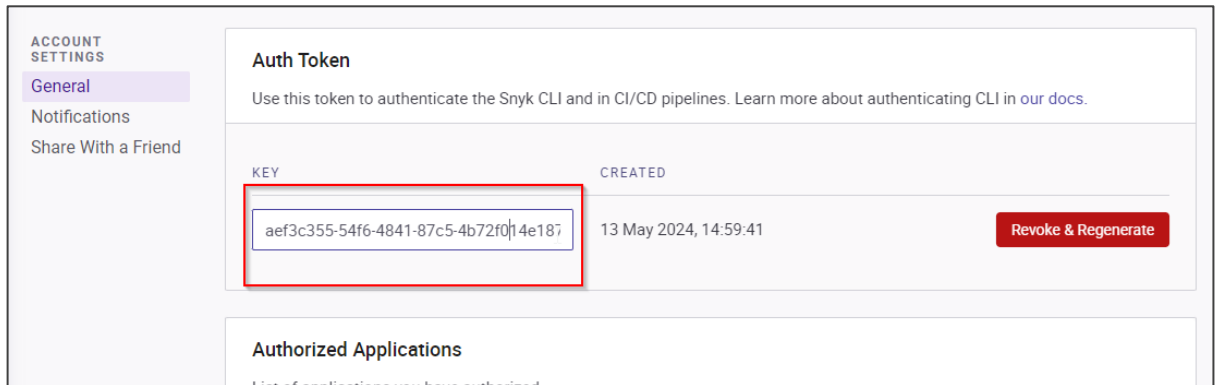


## 2.9 Under the **Auth Token** section, click the **click to show** button under the **KEY** field to reveal the hidden token key

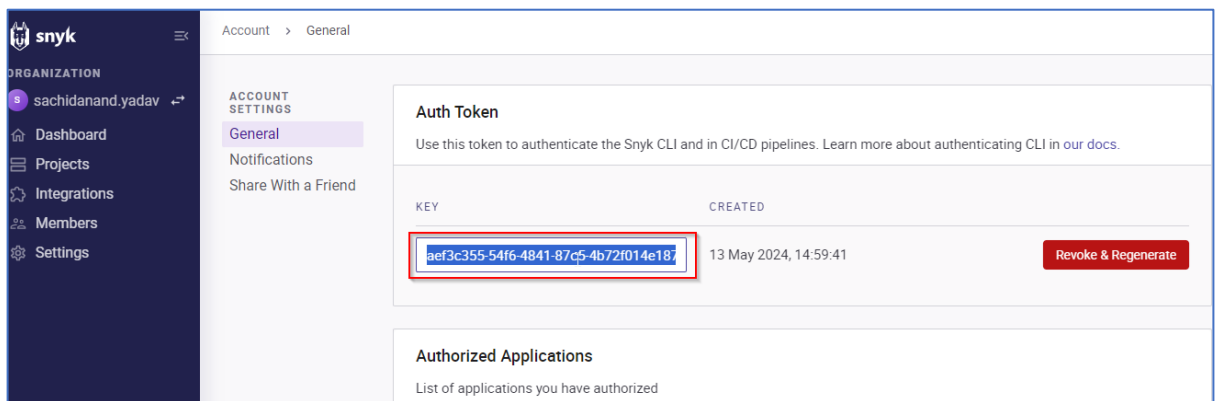




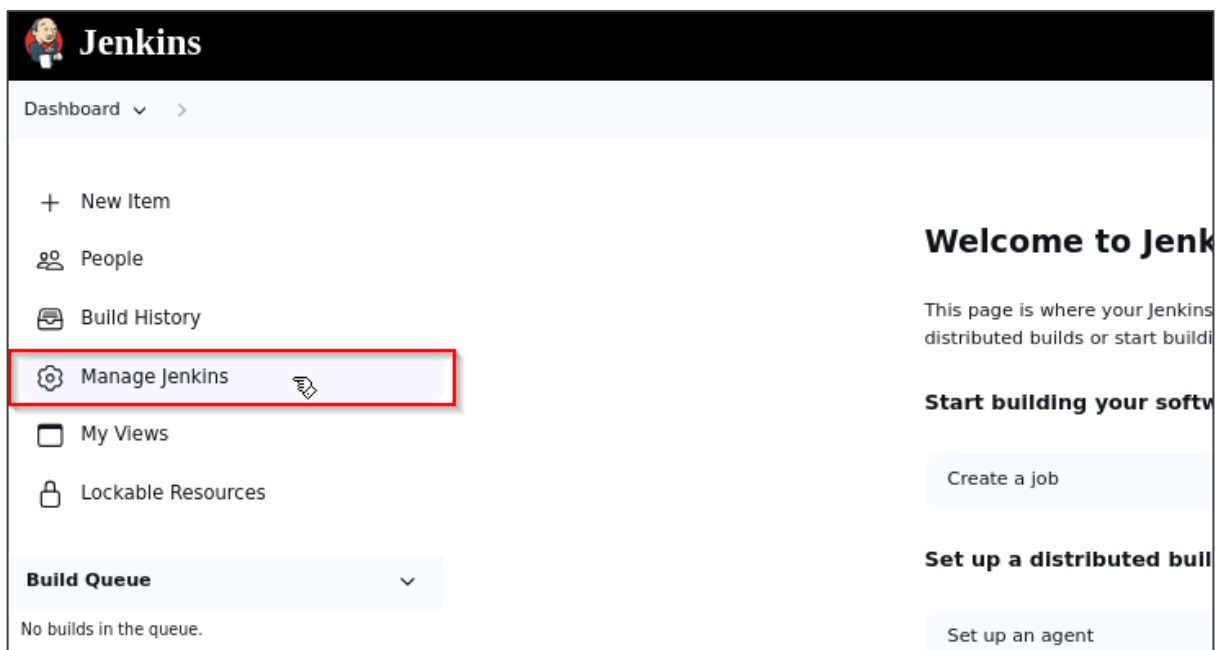
- 2.10 Under the **Auth Token** section, click the **Click to Show** button under the **KEY** field to reveal the hidden token key



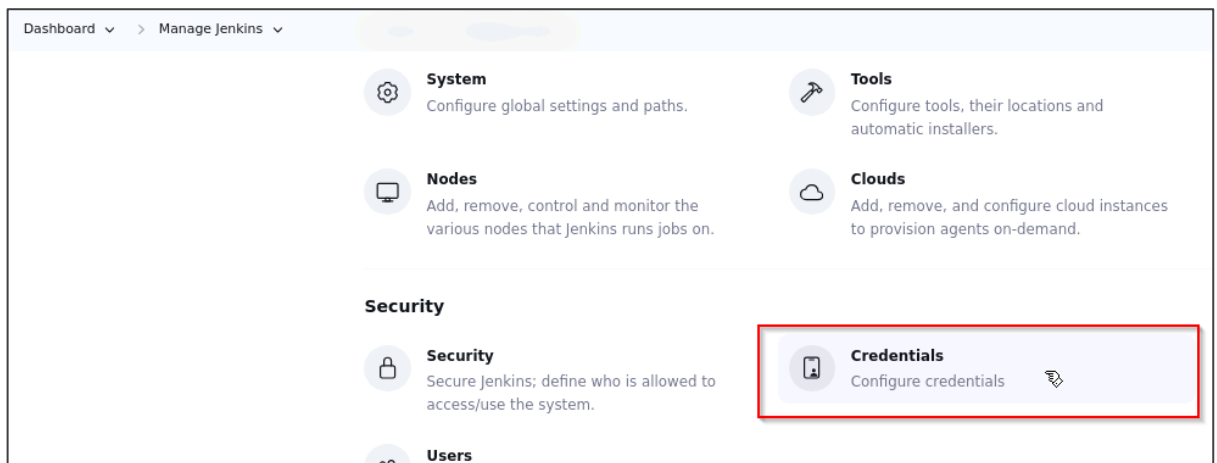
- 2.11 Copy the Snyk authentication token



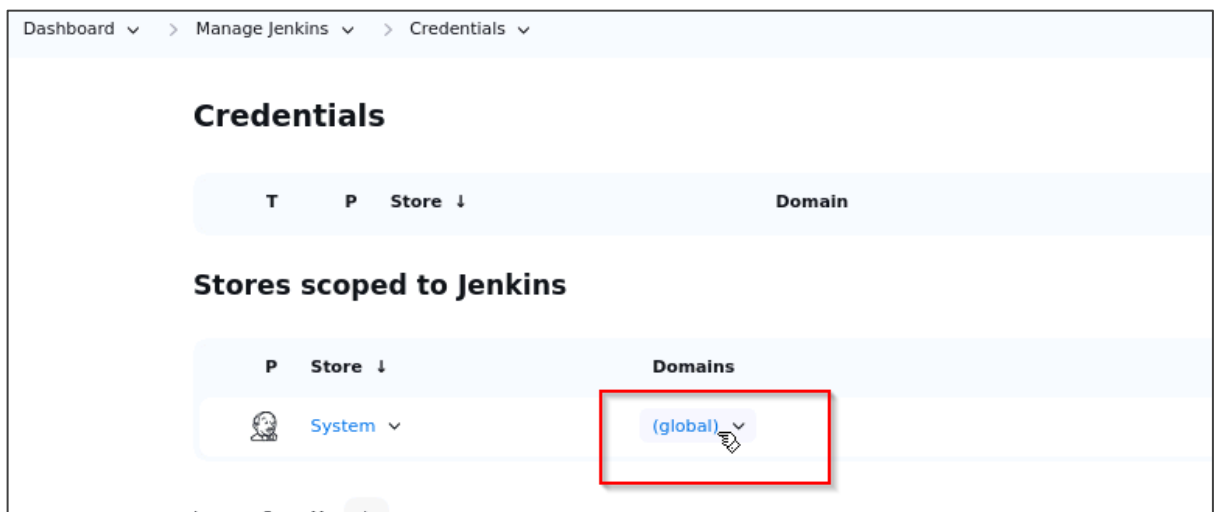
- 2.12 Navigate back to the Jenkins dashboard and click on **Manage Jenkins**



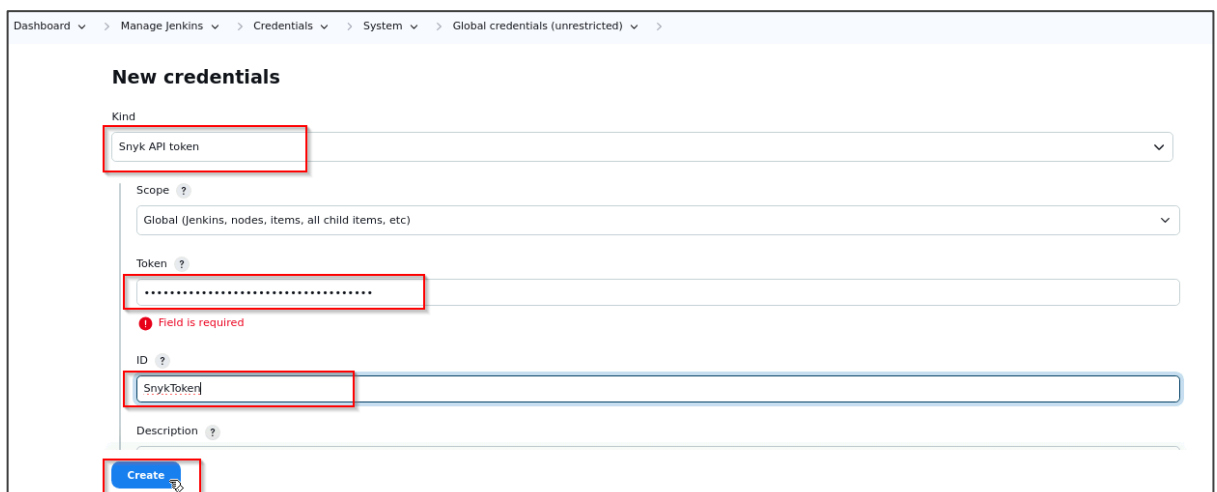
## 2.13 Select the **Credentials** option under **Manage Jenkins**



## 2.14 Click on the **global** option under the **Domains** column

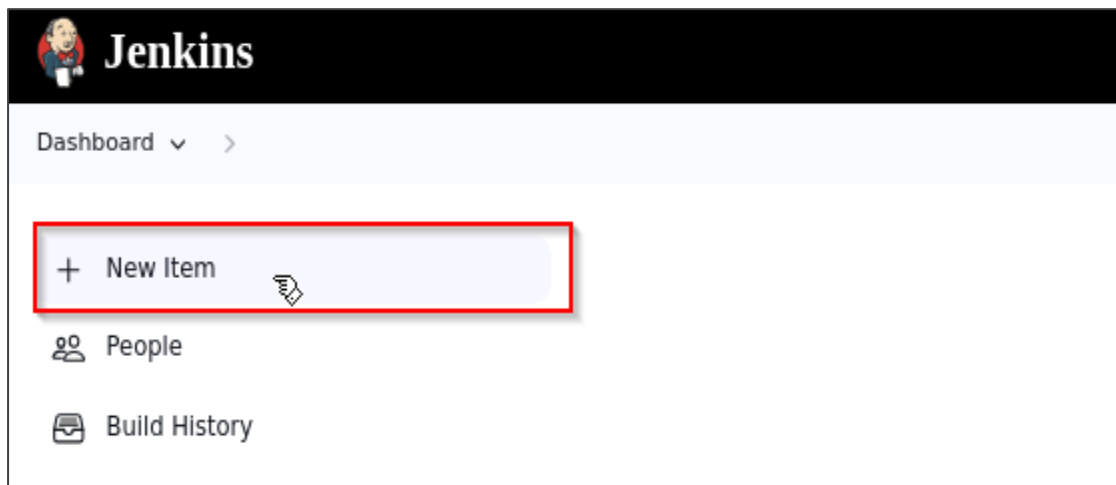


## 2.15 Select **Snyk API token** under the **Kind** field, paste the copied key from step 2.11 in the **Token** field, select ID as **SnykToken**, and click on **Create**

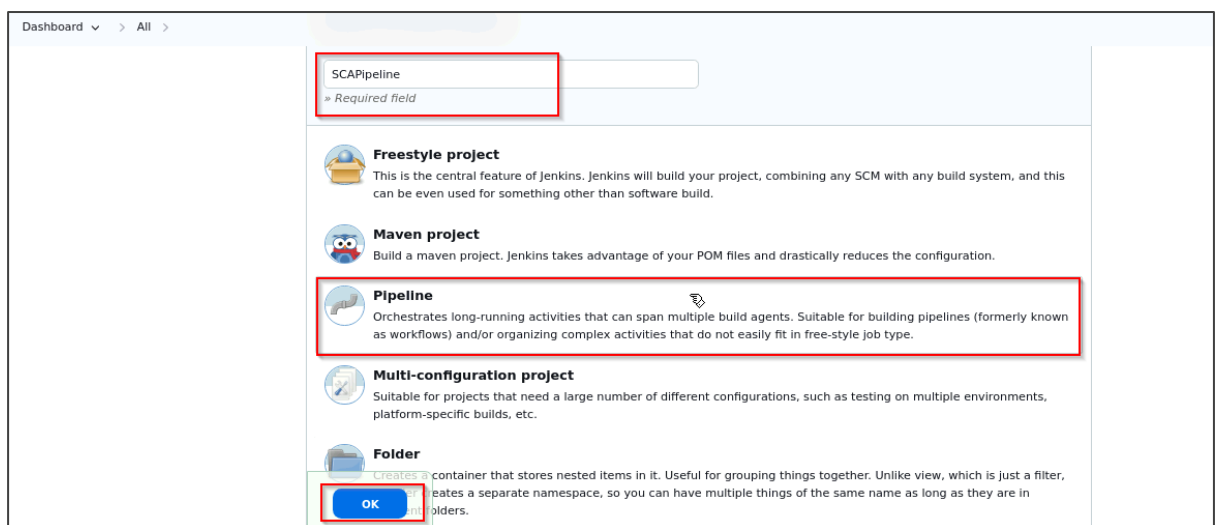


### Step 3: Create a new Jenkins pipeline job

3.1 Go to the Jenkins dashboard and click on **New Item**



3.2 Enter the item name as **SCAPipeline**, select the type as **Pipeline**, and click on the **OK** button



3.3 Enter the below code in the pipeline script and click on **Save**:

```
pipeline {  
  agent any  
  
  tools {  
    maven "Maven"  
    snyk "Snyk"  
  }  
  
  stages {  
    stage('Build & Test Automation') {  
      steps {
```

```

// Get some code from a GitHub repository
git 'https://github.com/anujdevopslearn/SonarQubeCoverageJava/'

// Run Maven on a Unix agent.
sh "mvn -Dmaven.test.failure.ignore=true clean package"
}

post {
    success {
        junit '**/target/surefire-reports/*.xml'
        archiveArtifacts 'target/*.jar'
    }
}
}
stage('SCA Scan') {
    steps {
        snykSecurity snykInstallation: 'Snyk', snykTokenId: 'SnykToken'
    }
}
}
}
}

```

Dashboard > SCAPipeline > Configuration

### Configure

- General
- Advanced Project Options
- Pipeline**

### Pipeline

Definition

Pipeline script

Script

```

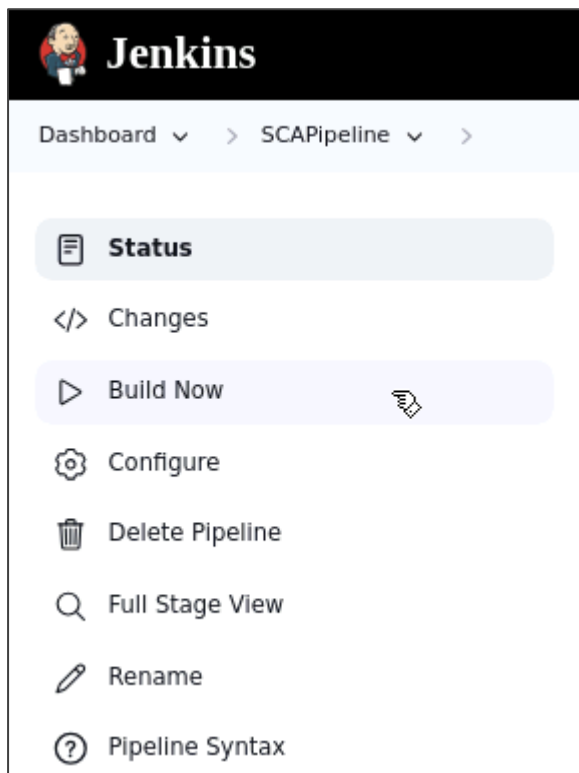
1 pipeline {
2   agent any
3   tools {
4     maven 'maven';
5     snyk 'snyk';
6   }
7   stages {
8     stage('Build & Test Automation') {
9       steps {
10        // Get some code from a GitHub repository
11        git
12        'https://github.com/anujdevopslearn/SonarQubeCoverageJava/';
13        // Run Maven on a Unix agent.
14        sh 'mvn -Dmaven.test.failure.ignore=true clean package';
15      }
16    }
17    post {
18      success {
19        junit '**/target/surefire-reports/*.xml';
20        archiveArtifacts 'target/*.jar';
21      }
22    }
23  }
24 }

```

☒ Use Groovy Sandbox

**Save** Apply

### 3.4 Click on the **Build Now** option to execute the job

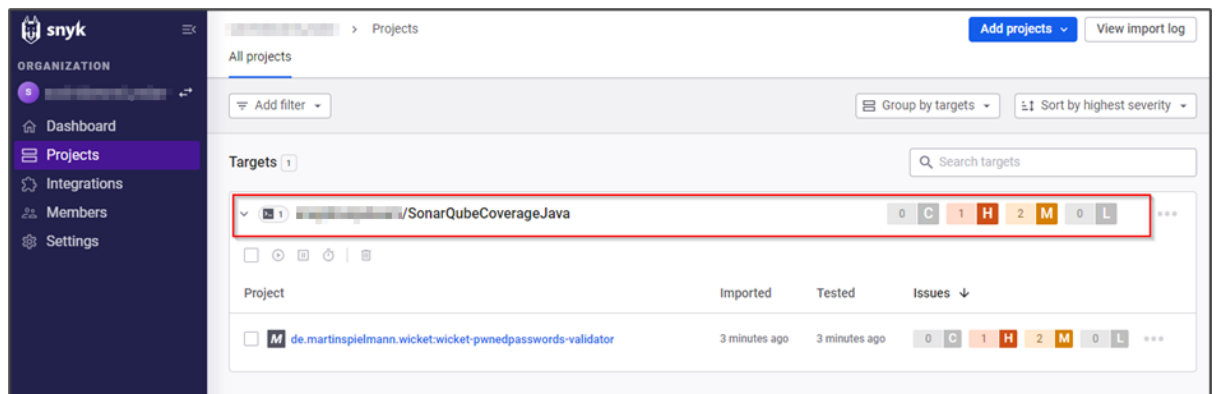


The output of the build will be as shown below:

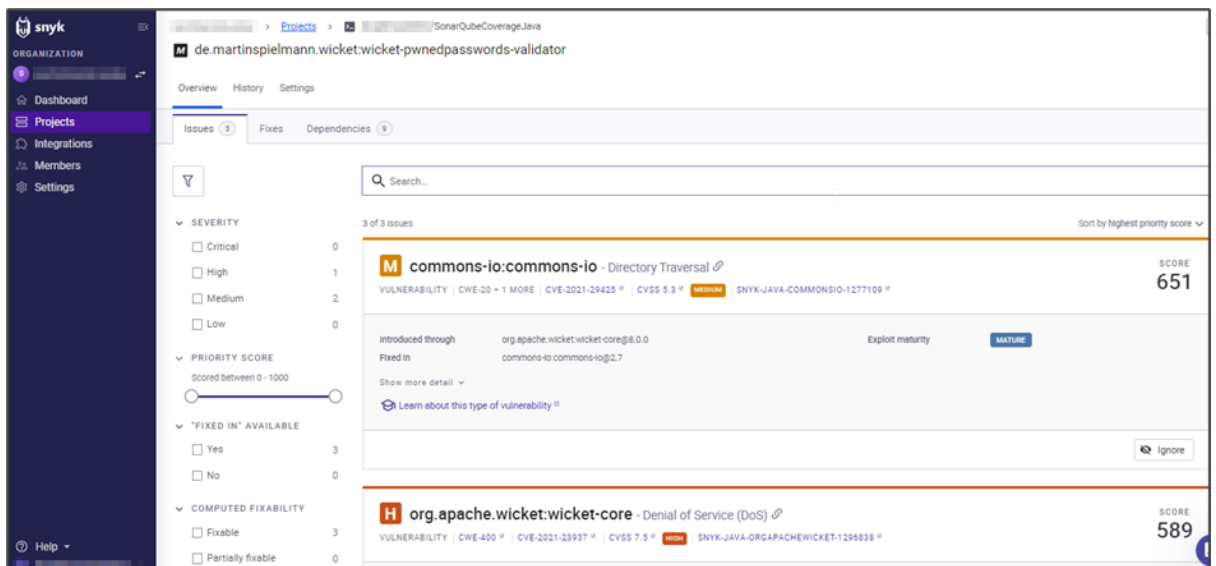
```
Dashboard > SCAPipeline > #20
#20
Vulnerabilities found!
Result: 3 known vulnerabilities | 9 dependencies
Generating report...
> /var/lib/jenkins/tools/io.snyk.jenkins.tools.SnykInstallation/Snyk/snyk-to-html-linux -i /var/lib/jenkins/workspace/SCAPipeline/2024-05-14T10-26-39-251053141Z_snyk_report.json
Archiving artifacts
Monitoring project...
> /var/lib/jenkins/tools/io.snyk.jenkins.tools.SnykInstallation/Snyk/snyk-linux monitor --severity-threshold=low
Monitoring /var/lib/jenkins/workspace/SCAPipeline (de.martinspielmann.wicket:wicket-pwnedpasswords-validator)...
Explore this snapshot at https://app.snyk.io/org/sachidanand.yadav/project/f0463d8f-24b9-4490-af3a-6fa5fa90e19c/history/8bd3b139-1b6c-4375-a401-c474b1a2218d
Notifications about newly disclosed issues related to these dependencies will be emailed to you.

[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
ERROR: Snyk has detected security vulnerabilities in your project.
```

### 3.5 Navigate to the Synk interface and click on the **Projects** tab



You can review code scan reports. In case of any vulnerabilities, it would be mentioned on the portal. You can validate the vulnerability report from Snyk to understand the security-related bugs.



By following these steps, you have effectively demonstrated how to automate SCA scans by integrating the Snyk plugin with Jenkins, enhancing the efficiency of vulnerability detection within Jenkins build jobs.