

Lab Exercise 19

Setting up Snyk for SAST in Jenkins

Objective: To demonstrate the setup of the Snyk plugin in Jenkins for Static Application Security Testing (SAST), to automatically detect vulnerabilities in their codebase during development, thereby enhancing application security before deployment

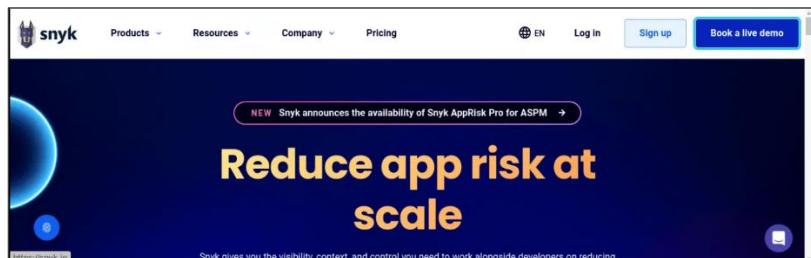
Tools required: Snyk

Steps to be followed:

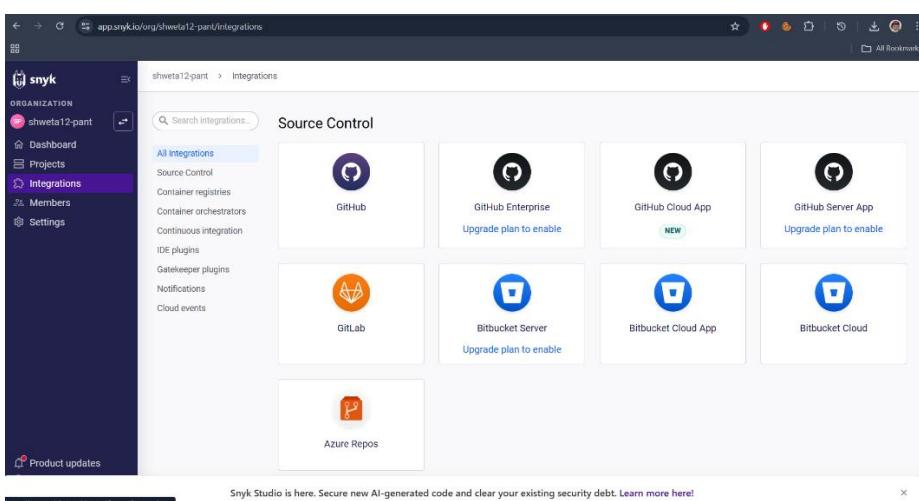
1. Configure Snyk as a SAST scan tool
2. Create and configure a Jenkins job for Snyk integration
3. Manage Snyk API and Jenkins credentials
4. Configure the Jenkins job for scanning

Step 1: Configure Snyk as a SAST scan tool

1.1 Visit <https://snyk.io/>, sign up for a new Snyk account, and log in



1.2 Navigate to Integrations and select Jenkins

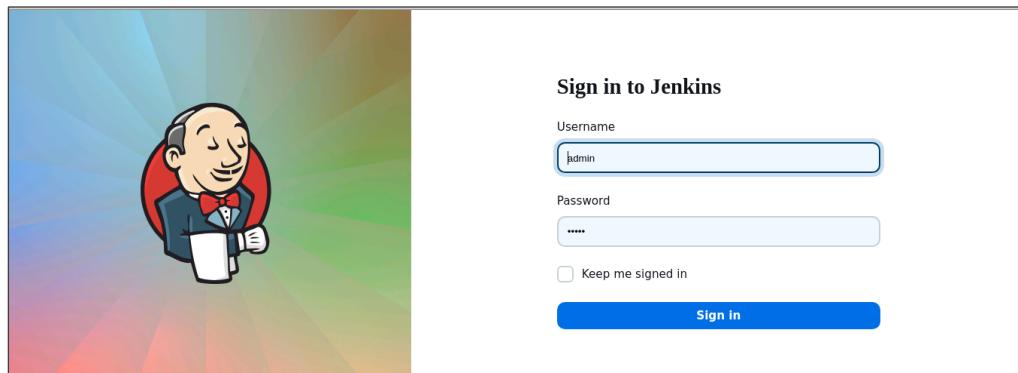


This will direct you to the documentation for integrating Snyk with Jenkins.

The screenshot shows a web browser displaying the Snyk User Docs website. The URL is docs.snyk.io/developer-tools/snyk-ci-cd-integrations/jenkins-plugin-integration-with-snyk. The page title is "Jenkins plugin integration with Snyk". The content includes sections on installing the plugin, configuring a Snyk installation, and adding Snyk Security to a project. A sidebar on the left provides links for scanning with Snyk, managing assets, and managing risk. A sidebar on the right provides links for installing the Jenkins plugin, configuring a Snyk installation, and troubleshooting the Jenkins plugin. A bottom banner says "Install the Snyk Security Jenkins Plugin" with a "Accept" button.

Step 2: Create and configure a Jenkins job for Snyk integration

2.1 Open Jenkins and log in to the Jenkins account:

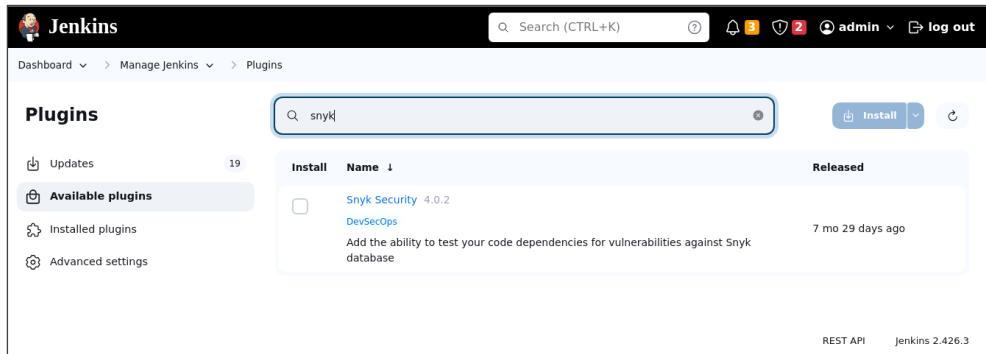


2.2 To install the Snyk plugin, navigate to **Manage Jenkins** and click **Available Plugins**, search for **Snyk Security** plugin, and then click **Install**

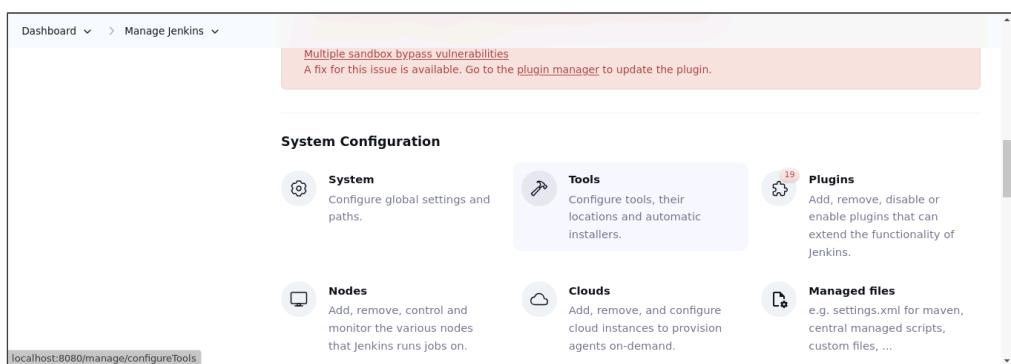
Note: The credentials for accessing Jenkins in the lab are Username: **admin** and Password: **admin**.

The screenshot shows the Jenkins Manage Jenkins dashboard under the "Available Plugins" tab. It lists several available plugins, including "Auto Trigger", "buildproject", "CodeScanSnyk", and "demo". Each plugin entry includes a status icon, name, last success, last failure, and last duration. The "Manage Jenkins" plugin is highlighted with a light gray background.

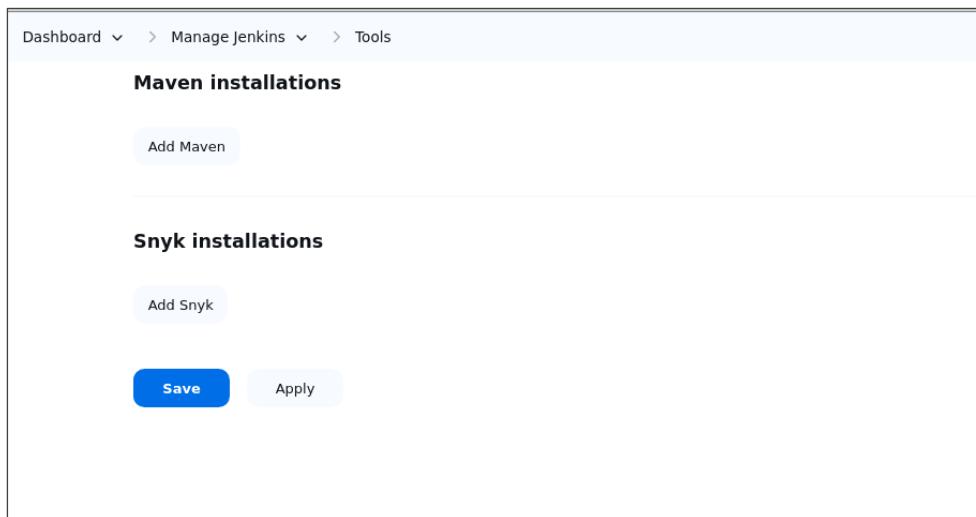
S	W	Name	Last Success	Last Failure	Last Duration
✓	☀️	Auto Trigger	20 days #2	N/A	1 sec
✓	☀️	buildproject	18 days #8	N/A	0.26 sec
✓	☀️	CodeScanSnyk	1 hr 26 min #2	N/A	15 sec
✓	☁️	demo	5 days 0 hr #7	5 days 0 hr #7	1.3 sec



2.3 To configure Maven and Snyk in the **Global Tool Configuration**, click on **Tools** inside **Manage Jenkins**

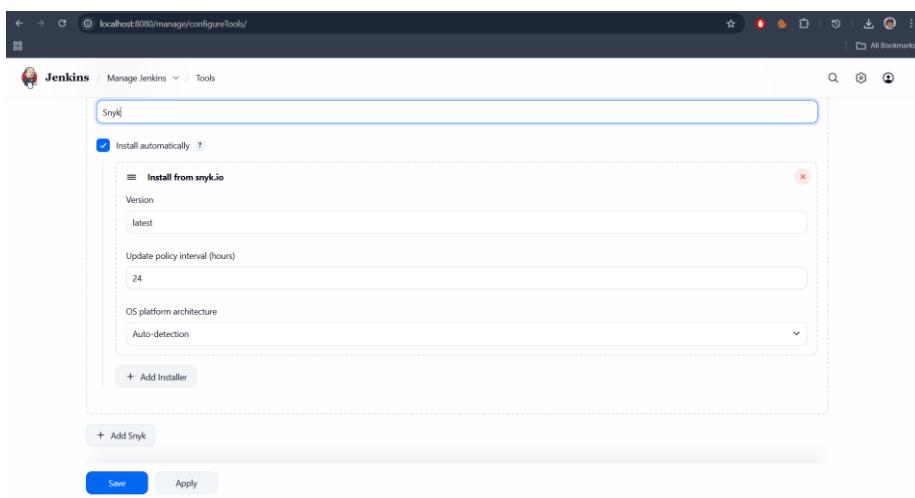
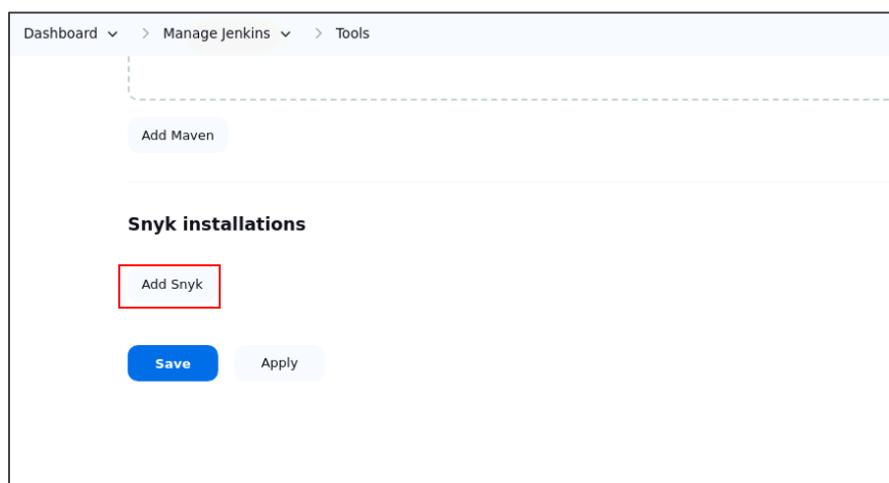


2.4 To add Maven, click on **Add Maven** under **Maven installations** and enter **Maven** as the **Name**





2.5 To add Snyk, click on **Add Snyk** under **Snyk Installations**, add **Name** as **Synk**, and click on the **Save** button



Step 3: Manage Snyk API and Jenkins credentials

3.1 To retrieve your Snyk API token, go to **Account Settings** in your Snyk account, click on **Click to show** under the Auth Token key field, and copy the token for further reference

The screenshot shows the 'General' tab of the Snyk Account settings. In the 'Auth Token' section, there is a text input field containing the placeholder 'click to show'. Below this field is a timestamp '25 September 2025, 12:03:53' and a red button labeled 'Revoke & Regenerate'. The 'Authorized Applications' section is empty, showing 'No applications'. The 'Preferred Organization' section shows 'shweta12-pant' selected. At the bottom of the page, a banner for 'Snyk Studio' is visible.

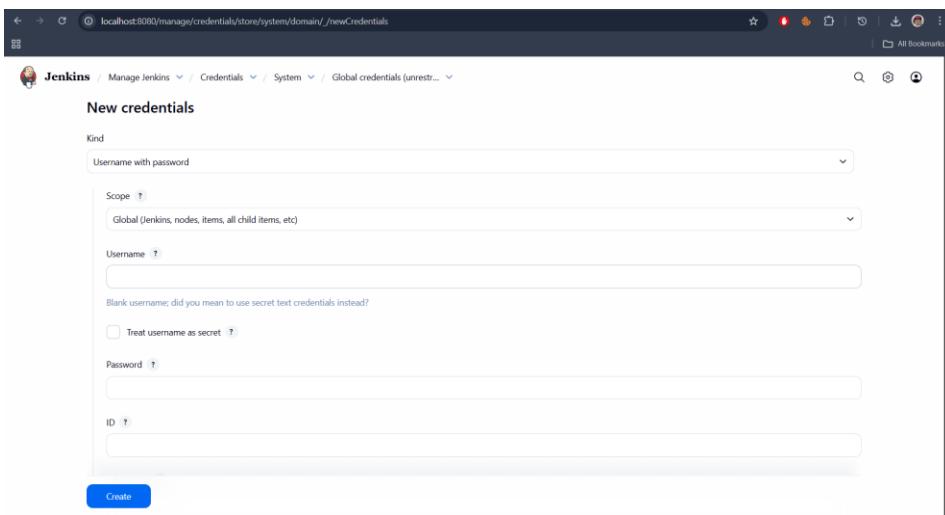
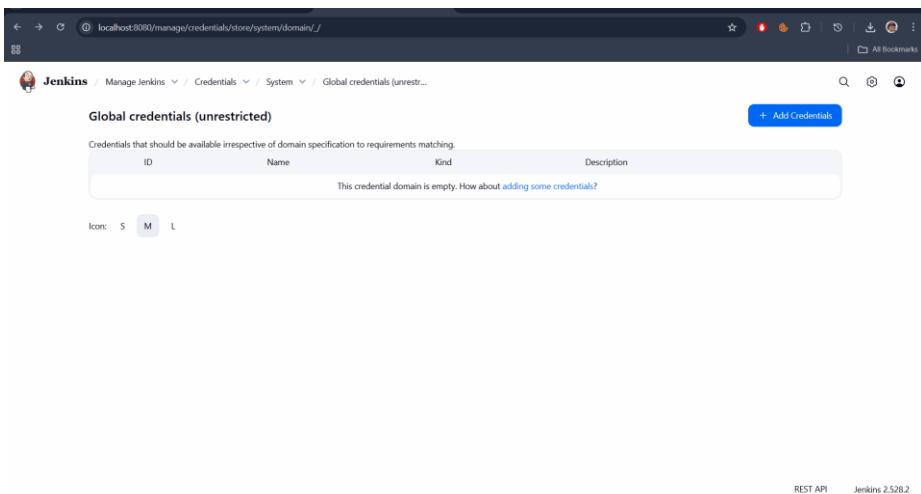
3.2 In the Jenkins interface, go to **Manage Jenkins**, select **Security**, then choose **Credentials** and select **global** to add global credentials

The screenshot shows the Jenkins Dashboard. On the left sidebar, the 'Manage Jenkins' link is highlighted. The main area displays a table of build items with columns: S, W, Name, Last Success, Last Failure, and Last Duration. The table includes entries for 'Auto Trigger', 'buildproject', 'CodeScanSnyk', and 'demo'.

The screenshot shows the 'Manage Jenkins' page under the 'Security' section. It features three main cards: 'Security' (Secure Jenkins), 'Credentials' (Configure credentials), and 'Credential Providers' (Configure the credential providers and types). Below these cards, there is a 'Users' section. The URL 'localhost:8080/manage' is visible at the bottom of the page.



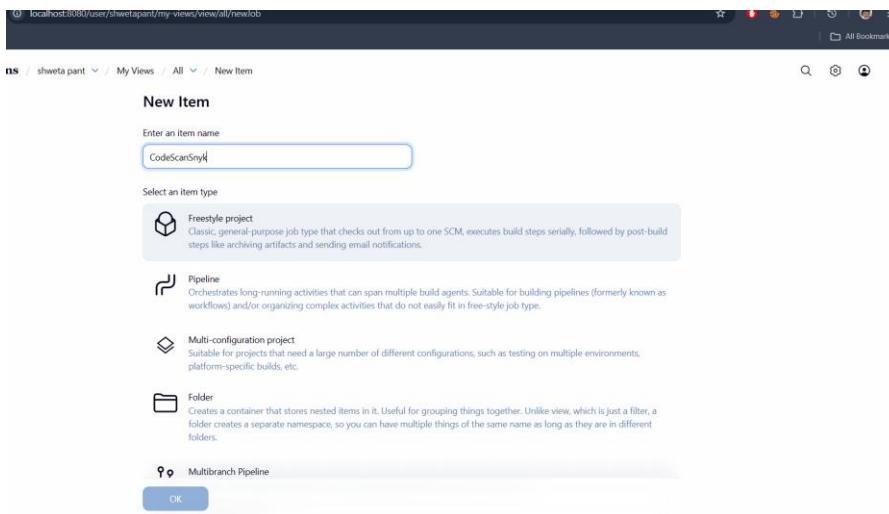
3.3 Click on **Add Credentials**, select the **Snyk API token** from the **Kind** field, paste the copied token from step 3.1 into the **Token** field, and then click the **Create** button



The screenshot shows the Jenkins 'New Credentials' configuration page. The 'Kind' dropdown is set to 'Snyk API token'. The 'Scope' dropdown is set to 'Global (Jenkins, nodes, items, all child items, etc.)'. The 'Token' field contains a masked value. The 'ID' field is filled with 'snyk token'. The 'Description' field contains 'Snyk API token for SAST scans'. A blue 'Create' button is at the bottom.

Step 4: Configure the Jenkins job for scanning

4.1 To create a new Jenkins job, click on **New Item**, enter the item name as **CodeScanSnyk**, select **Freestyle project**, and then click **OK**



4.2 After creating a job, go to **Source Code Management** and enter the GitHub repository URL. Then, under **Build Steps**, add the build step **Invoke Snyk Security task** with the name **SnykToken**. Finally, click the **Save** button to create the build.

Use GitHub Repo: <https://github.com/hkshitesh/Secure-Coding.git>

The image consists of three vertically stacked screenshots of the Jenkins configuration interface:

- Top Screenshot:** Shows the "Source Code Management" section where the "Repository URL" is set to <https://github.com/hkshitesh/Secure-Coding.git>.
- Middle Screenshot:** Shows the "Build Steps" section with a dropdown menu open, listing various build steps. The "Invoke Snyk Security task" option is highlighted.
- Bottom Screenshot:** Shows the "Build Steps" section with the "Invoke Snyk Security task" step added and configured with the name "SnykToken".

Note: For GitHub repository URL, use <https://github.com/hkshitesh/Secure-Coding.git>

4.3 To check the build status, click on the build link under **Permalinks**. After that, click on **Console Output**

The screenshot shows the Jenkins build status page for build #2. The top navigation bar includes 'Dashboard', 'CodeScanSnyk', '#2', and a dropdown menu. On the left, there's a sidebar with 'Status' (highlighted), 'Changes', 'Console Output' (selected), 'Edit Build Information', 'Delete build #2', and 'Git Build Data'. The main content area displays the build number (#2) and date (May 8, 2024, 9:24:10 AM). It shows 'Build Artifacts' (a file named 2024-05-08T09-24-17-848173830Z_snyk_report.html, 13.79 KB, with a 'view' link). Below that, it says 'No changes.' and 'Started by user admin (2 times)'. At the bottom, the URL is localhost:8080/job/CodeScanSnyk/lastBuild/console.

The screenshot shows the Jenkins build #2 console output. The log starts with '/var/lib/jenkins/workspace/CodeScanSnyk/2024-05-08T09-24-18-3/24082092_Snyk_report.json'. It then says 'Archiving artifacts' and 'Monitoring project... > /var/lib/jenkins/tools/io.snyk.jenkins.tools.SnykInstallation/Snyk/snyk-linux monitor --severity-threshold=low'. The next line is 'Monitoring /var/lib/jenkins/workspace/CodeScanSnyk (com.java.example:java-example)...'. It then provides a link to explore the snapshot at <https://app.snyk.io/org/palak.kharbanda/project/c08c922e-e55a-465f-91c7-f196291da77c/history/f6e31b9d-b848-43ef-bd73-e38bccf62ca4>. A note follows: 'Notifications about newly disclosed issues related to these dependencies will be emailed to you.' The log concludes with 'Finished: SUCCESS'.

4.4 To navigate to the Snyk tool to review code, scan reports under the **Projects** section

The screenshot shows the Snyk organization dashboard for 'palak.kharbanda'. The left sidebar has 'ORGANIZATION' (palak.kharbanda), 'Dashboard', 'Projects' (selected), 'Integrations', 'Members', and 'Settings'. The main content area shows 'All projects' with a 'Targets' section containing 'anujdevopslearn/MavenBuild'. There are buttons for 'Add projects' and 'View import log'. A message at the bottom says 'Ready to import another project?' with a link to 'Secure your entire stack with Snyk'.

By following the above steps, you have successfully demonstrated the setup of the Snyk plugin in Jenkins for static application security testing (SAST), to automatically

detect vulnerabilities in their codebase during development, thereby enhancing application security before deployment.