

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

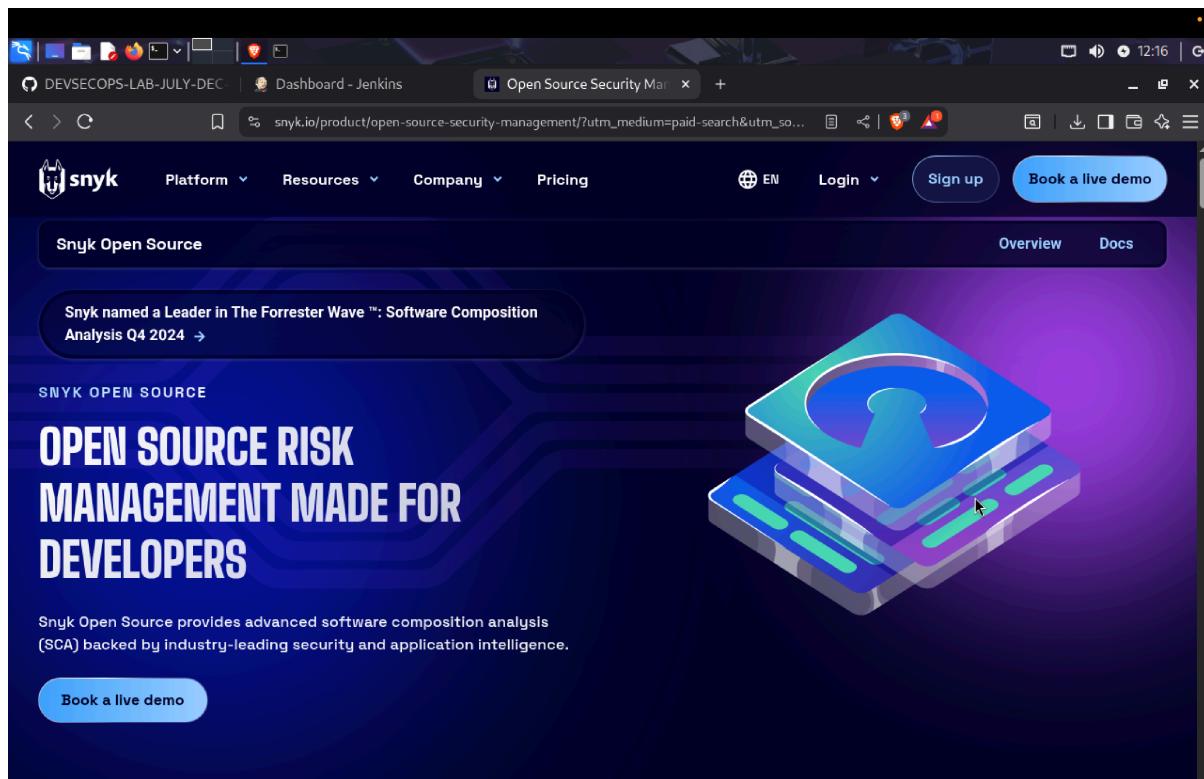
Prerequisites: None

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



2. Navigate to **Integrations** and select **Jenkins**

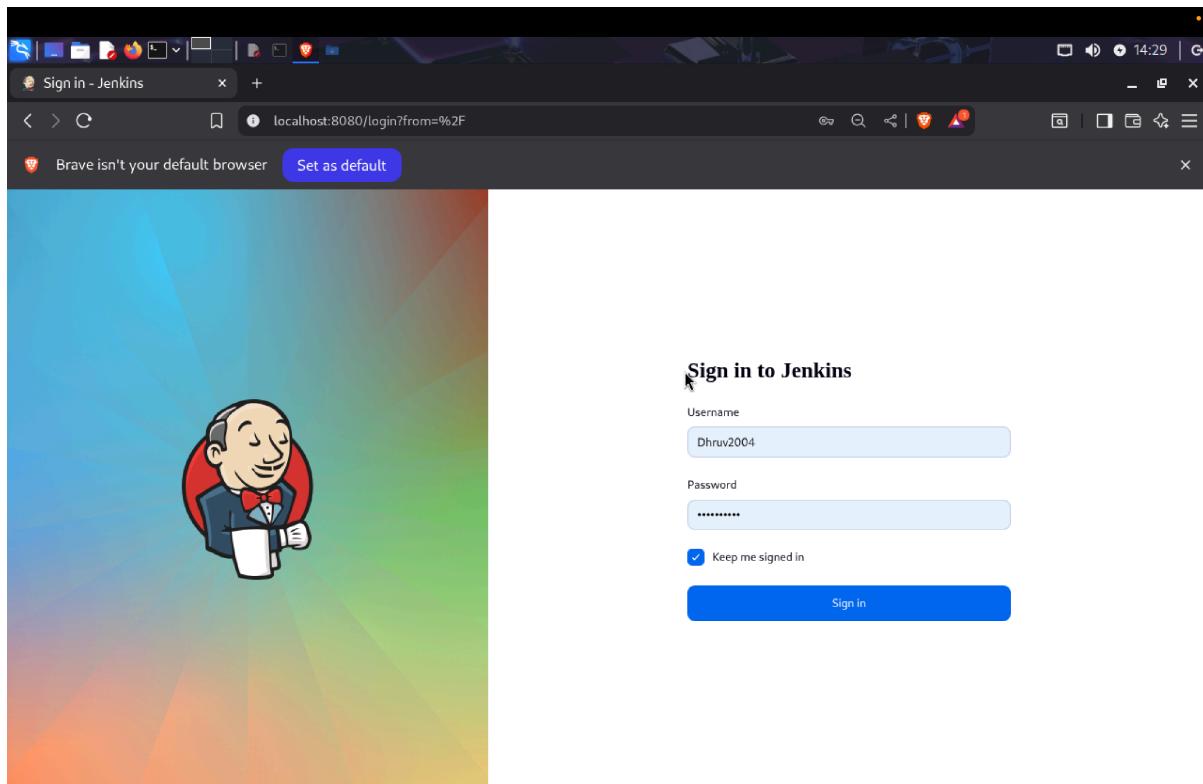
The screenshot shows the Snyk web interface. On the left, there's a sidebar with options like Organization, Dashboard, Projects, Integrations (which is selected and highlighted in purple), Members, and Settings. The main content area has a search bar with 'jen' typed in. Below it, a section titled 'Continuous integration' is shown, featuring a card for 'Jenkins' with a Jenkins logo icon and a 'View Documentation' button.

The screenshot shows the 'Jenkins plugin integration with Snyk' page from the Snyk User Docs. The page title is 'Jenkins plugin integration with Snyk'. It contains a brief introduction: 'Snyk offers a native plugin for Jenkins that is based on the [Snyk CLI](#), to test and monitor Projects for vulnerabilities in your pipelines.' A note states: 'The Snyk Jenkins plugin supports Snyk Open Source. If you plan to include Snyk Code, Snyk Container, and Snyk IaC scans in your pipeline, use the generic [Snyk CLI](#).' Below this, there's a list of steps to use the plugin: 1. Install the Snyk Security Jenkins Plugin, 2. Configure a Snyk installation, 3. Configure a Snyk API token credential, 4. Add Snyk Security to your Project, and 5. View your Snyk Security Report. To the right, there's a sidebar with links for installing the Jenkins plugin, configuring Snyk, adding Snyk security to a project, and viewing reports. At the bottom, there's a cookie consent banner.

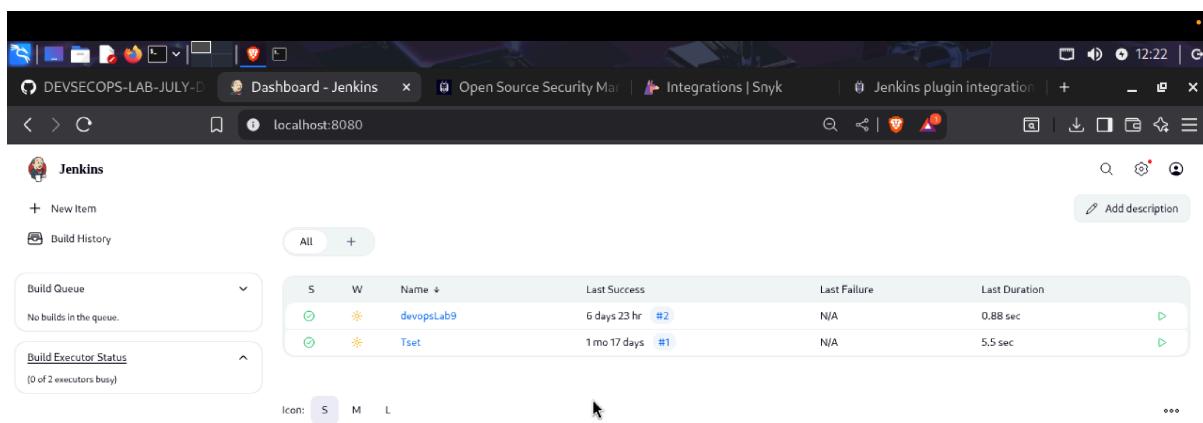
This will direct you to the documentation for integrating Snyk with Jenkins.

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

1. Open Jenkins and log in to the Jenkins account:



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



The screenshot shows the Jenkins plugin manager interface. The left sidebar has tabs for 'Updates' (47), 'Available plugins' (selected), 'Installed plugins', and 'Advanced settings'. A search bar at the top right contains the query 'sny'. The main table lists the 'Snyk Security' plugin version 5.0.1 by DevSecOps. It includes a brief description: 'Add the ability to test your code dependencies for vulnerabilities against Snyk database'. The table columns are 'Install', 'Name', 'Released', and 'Health'. The 'Released' column shows '3 mo 18 days ago' and the 'Health' column shows a green status icon.

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

The screenshot shows the 'Manage Jenkins' page. At the top, there is a message about a new Jenkins version (2.516.3) available for download ([changelog](#)). Below this, a red box highlights 'Warnings have been published for the following currently installed components:'

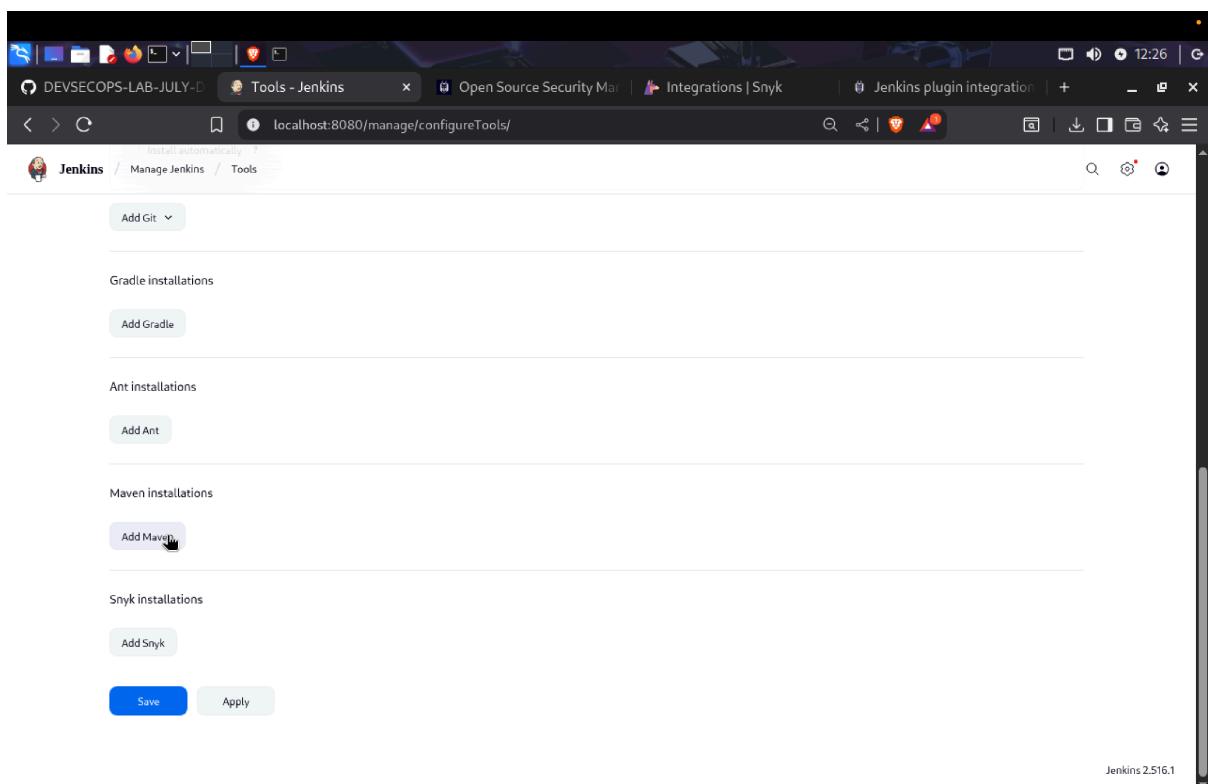
- Jenkins 2.516.1 core and libraries:
Multiple security vulnerabilities in Jenkins 2.527 and earlier, LTS 2.516.2 and earlier
A fix for this issue is available. Update Jenkins now.
- Jakarta Mail API 2.1.3-2:
SMTP command injection vulnerability
A fix for this issue is available. Go to the [plugin manager](#) to update the plugin.
- Git client plugin 6.3.0:
File system information disclosure vulnerability
A fix for this issue is available. Go to the [plugin manager](#) to update the plugin.

Buttons for 'Go to plugin manager' and 'Configure which of these warnings are shown' are present. The bottom section, 'System Configuration', contains several items with icons:

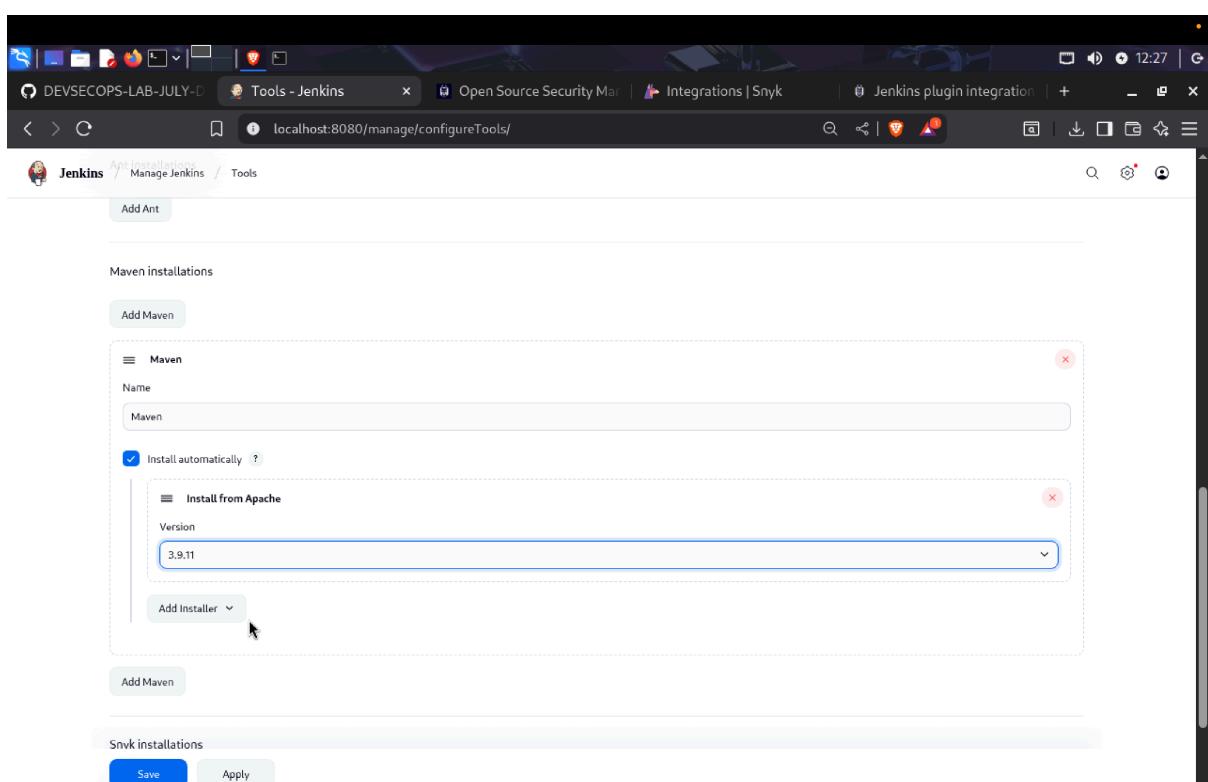
- System**: Configure global settings and paths.
- Tools**: Configure tools, their locations and automatic installers. (47)
- Plugins**: Add, remove, disable or enable plugins that can extend the functionality of Jenkins.
- Nodes**: Add, remove, control and monitor the various nodes that Jenkins runs jobs on.
- Clouds**: Add, remove, and configure cloud instances to provision agents on-demand.
- Appearance**: Configure the look and feel of Jenkins.

The URL in the address bar is 'localhost:8080/manage/configureTools'.

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

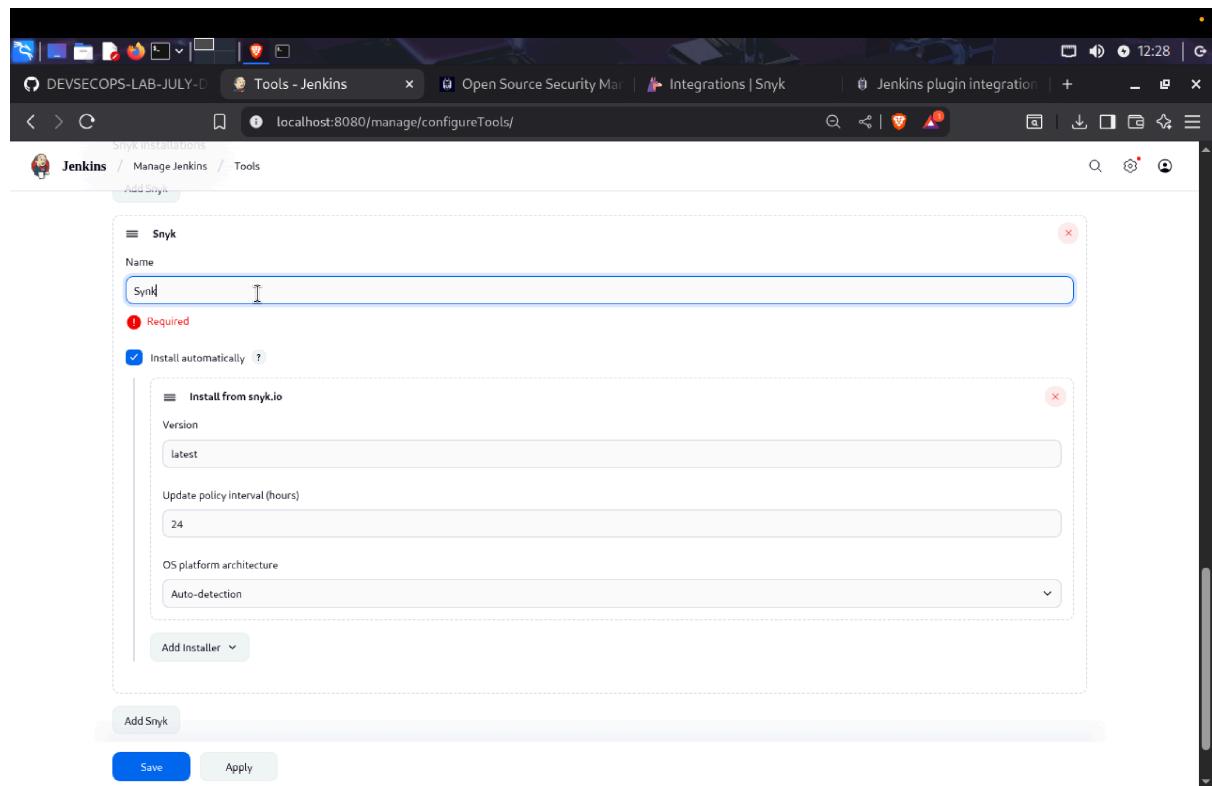


The screenshot shows the Jenkins 'Tools - Jenkins' configuration page. Under the 'Maven installations' section, there is a button labeled 'Add Maven'. A mouse cursor is hovering over this button, indicating it is the target of the next action.



The screenshot shows the Jenkins 'Tools - Jenkins' configuration page with the 'Add Maven' dialog open. The 'Name' field is filled with 'Maven'. Below it, the 'Install automatically' checkbox is checked. Under the 'Install from Apache' section, the 'Version' dropdown is set to '3.9.11'. A dropdown menu labeled 'Add Installer' is visible at the bottom of the dialog. The 'Save' and 'Apply' buttons are located at the bottom of the main configuration page.

4. 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

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 Snyk organization settings interface. On the left, there's a sidebar with 'Organization' and 'dhruv-cs2004' selected. The main area has tabs for 'General', 'Integrations', and 'Members'. Under 'General', there are fields for 'Organization name' (set to 'dhruv-cs2004') and 'Organization slug' (set to 'dhruv-cs2004'). Below that is an 'Organization API key' section with a 'Manage service accounts' button. At the bottom is an 'Organization ID' field. A sidebar on the left shows account details: 'Dhruvsapra448@gmail.com' and 'Dhruvsapra448@gmail.com'. Below the sidebar are links for 'Account settings', 'Notification preferences', 'Share with a friend', and 'Log out'. The URL in the address bar is 'https://app.snyk.io/account'.

The screenshot shows the Snyk account general settings page. The left sidebar has 'Account settings' with 'General' selected. The main area has sections for 'Auth Token', 'Authorized Applications', and 'Preferred Organization'. In the 'Auth Token' section, there's a 'KEY' field with 'click to show' and a 'CREATED' timestamp of '29 September 2025, 12:19:50'. A red 'Revoke & Regenerate' button is visible. The 'Authorized Applications' section says 'No applications'. The 'Preferred Organization' section shows 'dhruv-cs2004' selected. The URL in the address bar is 'https://app.snyk.io/account'.

The screenshot shows the 'General' tab of the Snyk account settings. It displays an 'Auth Token' section with a key value '20ba6466-238c-473c-ba86-429ef8011630' created on '29 September 2025, 12:19:50'. A red 'Revoke & Regenerate' button is visible. Below this is an 'Authorized Applications' section showing 'No applications'. At the bottom is a 'Preferred Organization' section with a dropdown set to 'dhruv-cs2004'.

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

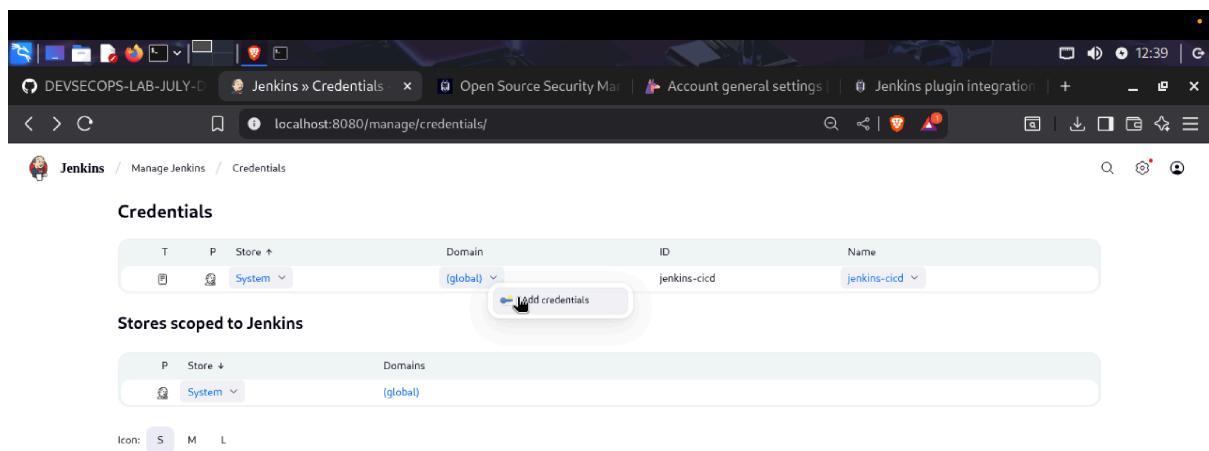
The screenshot shows the Jenkins dashboard at 'localhost:8080'. It features a 'Build Queue' section indicating 'No builds in the queue.' and a 'Build Executor Status' section showing '0 of 2 executors busy'. The main area displays a table of builds:

S	W	Name	Last Success	Last Failure	Last Duration
●	●	devopsLab9	6 days 23 hr #2	N/A	0.88 sec
●	●	Tset	1 mo 17 days #1	N/A	5.5 sec

The screenshot shows the Jenkins Manage Jenkins interface at localhost:8080/manage/. The main navigation bar includes links for Open Source Security, Account general settings, Jenkins plugin integration, and a search bar. The left sidebar has a 'Security' section with three items: 'Security' (selected), 'Credentials', and 'User'. Below this are sections for 'Status Information' (System Information, System Log, Load Statistics) and 'Troubleshooting' (Manage Old Data). The URL in the address bar is localhost:8080/manage/configureSecurity.

The screenshot shows the Jenkins Credentials management interface at localhost:8080/manage/credentials/. The main navigation bar includes links for Open Source Security, Account general settings, Jenkins plugin integration, and a search bar. The left sidebar has a 'Credentials' section with a table showing one entry: ID: jenkins-cicd, Name: jenkins-cicd, Store: System, Domain: (global). Below this is a section titled 'Stores scoped to Jenkins' with a table showing one entry: Store: System, Domain: (global). The URL in the address bar is localhost:8080/manage/credentials/.

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



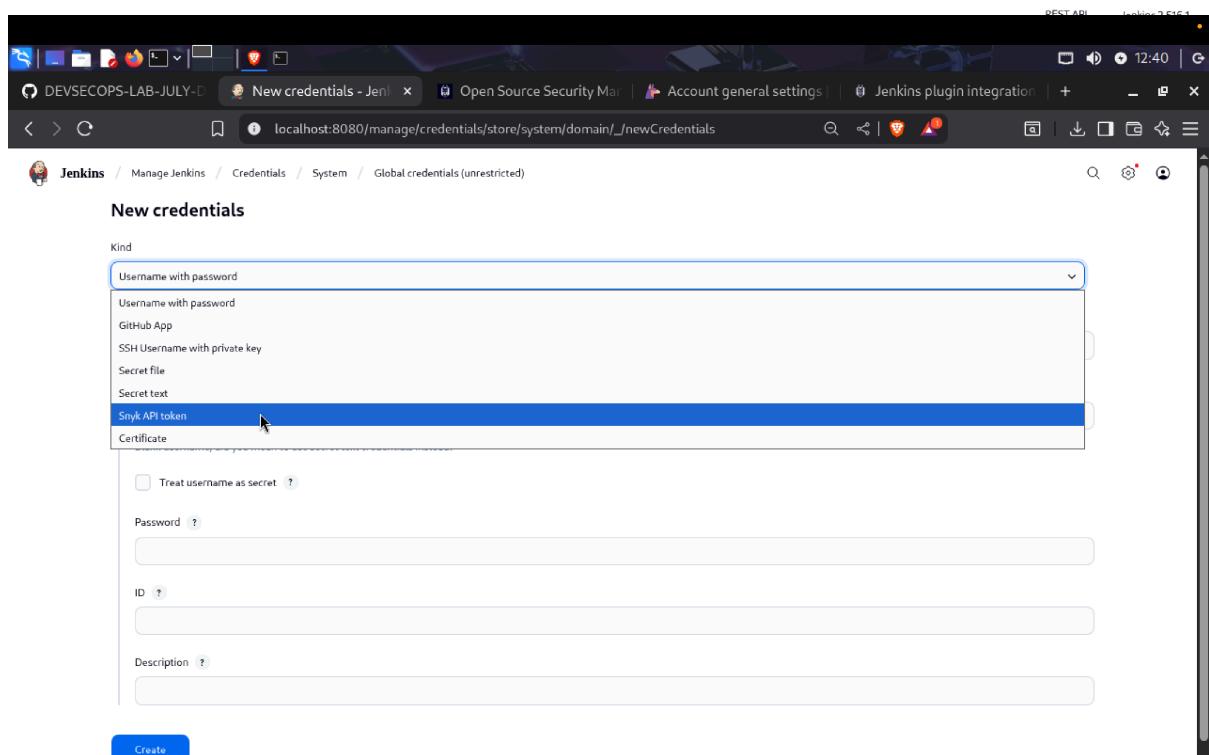
The screenshot shows the Jenkins 'Credentials' management page. A single credential entry is listed:

T	P	Store ↑	Domain	ID	Name
		System	(global)	jenkins-cicd	jenkins-cicd

Below this, there's a section titled 'Stores scoped to Jenkins' with a single entry:

P	Store ↓	Domains
	System	(global)

Icon options S, M, L are shown at the bottom.



The screenshot shows the 'New credentials' creation page. The 'Kind' dropdown menu is open, and the 'Snyk API token' option is selected, highlighted with a blue background.

Below the dropdown, there are several input fields and buttons:

- Treat username as secret ?
- Password ? (text input field)
- ID ? (text input field)
- Description ? (text input field)
- Create** button

The screenshot shows the Jenkins 'New credentials' creation 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 redacted value with an error message: 'Field is required'. The 'ID' field is empty. The 'Description' field contains 'SynkToken'. A blue 'Create' button is at the bottom.

New credentials

Kind

Snyk API token

Scope ?

Global (Jenkins, nodes, items, all child items, etc)

Token ?

.....
● Field is required

ID ?

Description ?

SynkToken

Create

REST API Jenkins 2.516.1

The screenshot shows the Jenkins 'Global credentials (unrestricted)' list page. It displays two entries:

ID	Name	Kind	Description
jenkins-cicd	jenkins-cicd	Secret text	
22f2e8fd-a07a-473e-b834-6e231b08ff63	SynkToken	Snyk API token	SynkToken

+ Add Credentials

Icon: S M L

REST API Jenkins 2.516.1

1. 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.

The screenshot shows the Jenkins configuration interface for a job named 'devopsLab9'. The left sidebar lists 'General', 'Triggers', 'Environment', 'Build Steps', and 'Post-build Actions'. The 'Source Code Management' section is selected and highlighted with a red box. It contains a 'Repository URL' field with the value 'https://github.com/hkshitesh/UPEs-DEVOPS-MAVEN-PROJECT-B3.git', a 'Credentials' dropdown set to '- none -', and an 'Advanced' button. Below these are 'Add Repository' and 'Branches to build' sections. The 'Branches to build' section has a 'Branch Specifier' field with the value '*/*master'. At the bottom are 'Save' and 'Apply' buttons.

This screenshot shows the same Jenkins configuration page after adding build steps. The 'Build Steps' section is now visible in the sidebar. The 'Source Code Management' section remains the same as in the previous screenshot. A new 'Build Steps' section is present, containing an 'Add Step' button with a dropdown menu showing 'Invoke Snyk Security task' and 'SnykToken'. Below this are 'Repository browser' and 'Additional Behaviours' sections. The 'Save' and 'Apply' buttons are at the bottom.

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

Status ✓ **devopsLab9**

Permalinks

- Last build (#3), 8.4 sec ago
- Last stable build (#3), 8.4 sec ago
- Last successful build (#3), 8.4 sec ago
- Last completed build (#3), 8.4 sec ago

Builds	...
Filter	Filter
Today	
✓ #3 12:47 PM	▼
September 22, 2025	
✓ #2 12:42 PM	▼
✓ #1 12:42PM	▼

REST API Jenkins 2.516.1

#3 (Sep 29, 2025, 12:47:29 PM)

Started by user **Dhruv Sapra** Started 29 sec ago
Took 4.4 sec

Console Output

git Revision: d91aaabe4a98717d34da3db1fcc9f0b271d6dc14
Repository: <https://github.com/hkshitesh/UPEs-DEVOPS-MAVEN-PROJECT-B3.git>

Timings

- 48 ms waiting;
- 4.4 sec build duration;
- 4.4 sec total from scheduled to completion.

Changes No changes.

REST API Jenkins 2.516.1

The screenshot shows the Jenkins console output for build #3. The output details the git fetch and checkout process from a GitHub repository. It starts by fetching changes from the remote Git repository, then checks out Revision d91aabe4a98717d34da3db1fcc9fb271d6dc14c (refs/remotes/origin/master). The commit message is "Update MyCalc.java". The build concludes with a "Finished: SUCCESS".

```

Started by user Dhruv Sapra
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/devopsLab9
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/devopsLab9/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/hkshitesh/UPEs-DEVOPS-MAVEN-PROJECT-B3.git # timeout=10
Fetching upstream changes from https://github.com/hkshitesh/UPEs-DEVOPS-MAVEN-PROJECT-B3.git
> git --version # timeout=10
> git -v version 2.47.2'
> git fetch --tags --force --progress -- https://github.com/hkshitesh/UPEs-DEVOPS-MAVEN-PROJECT-B3.git +refs/heads/*:refs/remotes/origin/*
# timeout=10
> git rev-parse refs/remotes/origin/master{commit} # timeout=10
Checking out Revision d91aabe4a98717d34da3db1fcc9fb271d6dc14c (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f d91aabe4a98717d34da3db1fcc9fb271d6dc14c # timeout=10
Commit message: "Update MyCalc.java"
> git rev-list --no-walk d91aabe4a98717d34da3db1fcc9fb271d6dc14c # timeout=10
Finished: SUCCESS

```

<https://github.com/hkshitesh/UPEs-DEVOPS-MAVEN-PROJECT-B3.git> REST API Jenkins 2.516.1

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

The screenshot shows the Snyk dashboard with a single project named "Hkshitesh/Secure-Coding". This project has one dependency listed: "demo.secure.code.db". The dependency was imported 8 minutes ago and tested 8 minutes ago. The status is "Imported".

4.

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.

