

Lab Exercise 19

Setting up Snyc for SAST in Jenkins

Objective: To demonstrate the setup of the Snyc 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: Snyc

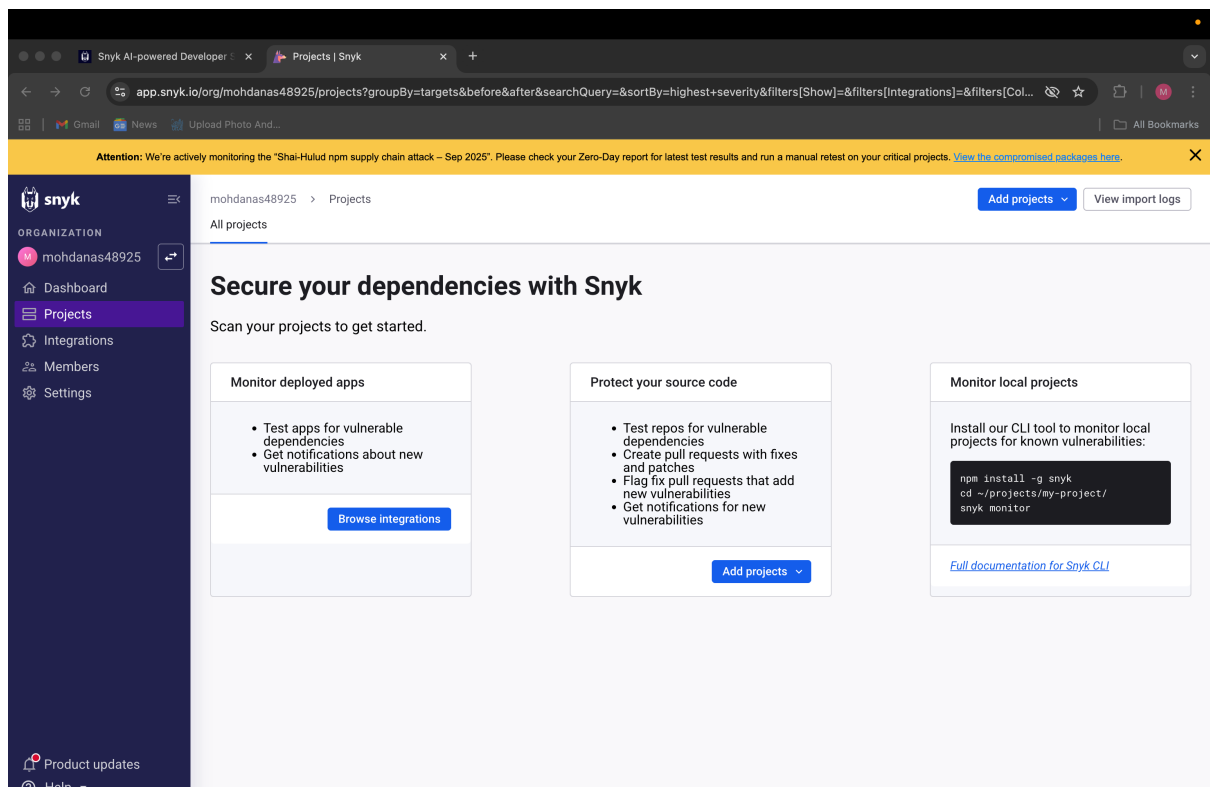
Prerequisites: None

Steps to be followed:

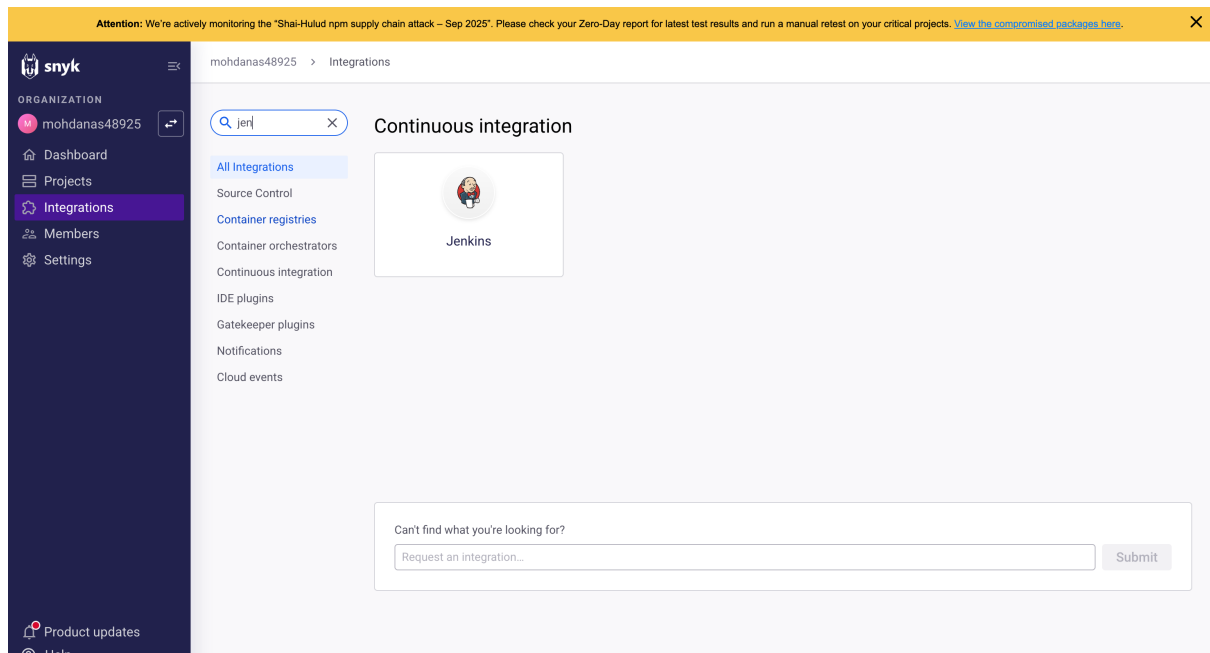
1. Configure Snyc as a SAST scan tool
2. Create and configure a Jenkins job for Snyc integration
3. Manage Snyc API and Jenkins credentials
4. Configure the Jenkins job for scanning

Step 1: Configure Snyc as a SAST scan tool

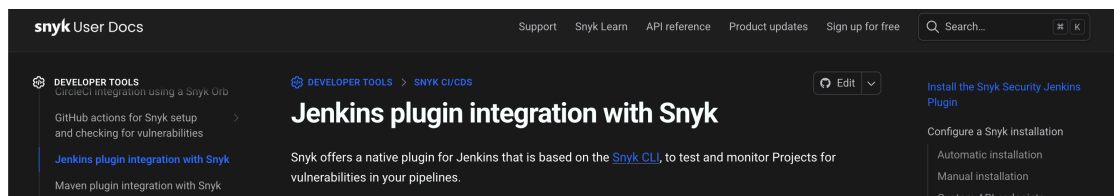
1. Visit <https://snyk.io/>, sign up for a new Snyc account, and log in



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

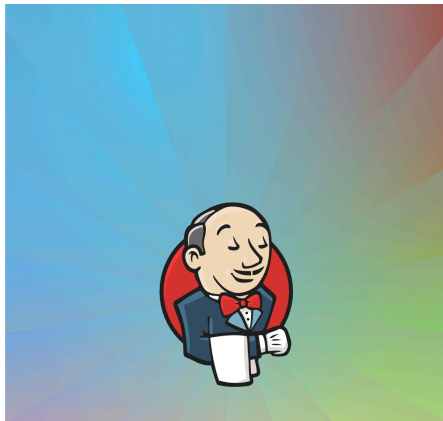


This will direct you to the documentation for integrating Snyk with Jenkins.



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

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



Sign in to Jenkins

Username

DakshDevarni


Password

.....

☐ Keep me signed in

Sign in

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

 **Jenkins**

Manage Jenkins / Plugins

Plugins

Updates 45

Available plugins

Installed plugins

Advanced settings

Q snyk

Install

7

Install	Name ↓	Released	Health
<input checked="" type="checkbox"/>	Snyk Security 5.0.1		
	DevSecOps	3 mo 13 days ago	93
Add the ability to test your code dependencies for vulnerabilities against Snyk database			

REST API

Jenkins 2.516.1

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

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

Add Maven

≡

Maven

×

Name

Maven

MAVEN_HOME

/opt/homebrew/Cellar/maven/3.9.11/libexec

☒ Install automatically ?

≡

Install from Apache

×

Version

3.9.11

▼

Add Installer

▼

Add Maven

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

Name

Snyk

☒ Install automatically ?

≡

Install from snyk.io

×

Version

latest

Update policy interval (hours)

24

OS platform architecture

Auto-detection

▼

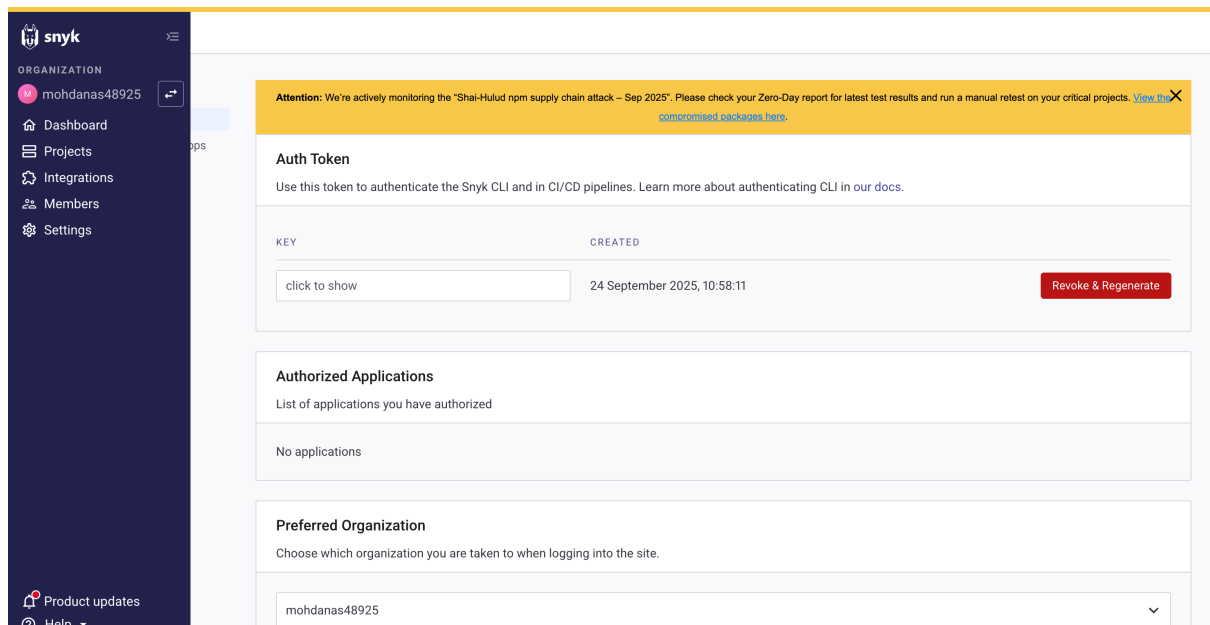
Add Installer

▼

Add Snyk

Step 3: Manage Snyc API and Jenkins credentials

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



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

Jenkins / Manage Jenkins / Credentials / System / Global credentials (unrestricted)

New credentials

Kind: Snyk API token

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

Token:
Field is required

ID: SNYK_DEVSECOPS_TOKEN

Description: SNYK_DEVSECOPS_TOKEN

Create

REST API Jenkins 2.516.1

Step 4: Configure the Jenkins job for scanning

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

New Item

Enter an item name

CodeScanSnyk

Select an item type



Freestyle project

Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.



Maven project

Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.



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.



Multi-configuration project

Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.



Folder

Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

OK

- 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>**

Jenkins / CodeScanSnyk / Configuration

Configure

- General
- Source Code Management
- Triggers
- Environment
- Build Steps
- Post-build Actions

☐ None

☒ Git ?

Repositories ?

Repository URL ?
https://github.com/hkshitesh/Secure-Coding.git

Credentials ?
- none -

+ Add

Advanced ▾

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?
*/master

Save Apply

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

Jenkins / CodeScanSnyk

Status

Changes

Workspace

Build Now

Configure

Delete Project

Rename

Credentials

CodeScanSnyk

Permalinks

- Last build (#9), 11 min ago
- Last failed build (#9), 11 min ago
- Last unsuccessful build (#9), 11 min ago
- Last completed build (#9), 11 min ago

Add description

Builds

Filter

Today

- #9 3:55 PM
- #8 3:49 PM
- #7 3:46 PM
- #6 3:37 PM
- #5 3:33 PM
- #4 3:25 PM
- #3 3:12 PM
- #2 3:11 PM


```
Started by user Mohd Anas
Running as SYSTEM
Building in workspace /Users/mohdanas/.jenkins/workspace/CodeScanSnyk
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /Users/mohdanas/.jenkins/workspace/CodeScanSnyk/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/hkshitesh/Secure-Coding.git # timeout=10
Fetching upstream changes from https://github.com/hkshitesh/Secure-Coding.git
> git --version # timeout=10
> git --version # 'git version 2.39.5 (Apple Git-154)'
> git fetch --tags --force --progress -- https://github.com/hkshitesh/Secure-Coding.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 5e3aaedae26e41b315263bf3151216fd7eb416b1 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 5e3aaedae26e41b315263bf3151216fd7eb416b1 # timeout=10
Commit message: "Add files via upload"
> git rev-list --no-walk 5e3aaedae26e41b315263bf3151216fd7eb416b1 # timeout=10
[CodeScanSnyk] $ /bin/bash /var/folders/0/_v_gbvylN4lq6pyx0dyp7yy9w0000gn/T/jenkins14985942942890390526.sh
--- Preparing environment for Snyk Scan ---
Verifying Maven installation...
Apache Maven 3.9.11 (3e54c93a704957b63ee3494413a2b544fd3d825b)
Maven home: /opt/homebrew/Cellar/maven/3.9.11/libexec
Java version: 23.0.2, vendor: Oracle Corporation, runtime: /Users/mohdanas/Library/Java/JavaVirtualMachines/openjdk-23.0.2/Contents/Home
Default locale: en_US, platform encoding: UTF-8
OS name: "mac os x", version: "15.5", arch: "aarch64", family: "mac"

Now redirecting you to our auth page, go ahead and log in,
and once the auth is complete, return to this prompt and you'll
be ready to start using snyk.

If you can't wait use this url:
```

```
Monitoring /Users/mohdanas/.jenkins/workspace/CodeScanSnyk
(demo.secure.code.db:demo.secure.code.db)...
```

Explore this snapshot at
<https://app.snyk.io/org/mohdanas48925/project/2563e6a3-d196-4f61-996a-eca0c6bdab07/history/3d94e363-249e-4000-aa5c-51dead5227f2>

Notifications about newly disclosed issues related to these dependencies
will be emailed to you.

```
Monitoring /Users/mohdanas/.jenkins/workspace/CodeScanSnyk
(com.basicsstrong:securecoding)...
```

Explore this snapshot at
<https://app.snyk.io/org/mohdanas48925/project/69242b35-9163-41a8-bd8b-5ff96ae7835a/history/f8a6a38a-a40c-4615-a45e-0403846c36fc>

Notifications about newly disclosed issues related to these dependencies
will be emailed to you.

```
Monitoring /Users/mohdanas/.jenkins/workspace/CodeScanSnyk
(demo.secure.code.db:demo.secure.code.db)...
```

Explore this snapshot at
<https://app.snyk.io/org/mohdanas48925/project/2563e6a3-d196-4f61-996a-eca0c6bdab07/history/9d3fcb15-36fe-4b99-bb2a-6d8ccaadb164>

Notifications about newly disclosed issues related to these dependencies
will be emailed to you.

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

The screenshot displays the Snyk web application interface. On the left, a dark sidebar contains the Snyk logo and navigation links: ORGANIZATION, mohdanas48925, Dashboard, Projects (highlighted), Integrations, Members, and Settings. The main content area is titled 'mohdanas48925 > Projects' and includes buttons for 'Add projects' and 'View import logs'. Below this, there's a section for 'All projects' with an 'Add filter' button and dropdowns for 'Group by targets' and 'Sort by highest severity'. The 'Targets' section shows a list of projects. The first project, 'hkshitesh/Secure-Coding', has a summary bar with 4 Critical (C), 23 High (H), 13 Medium (M), and 6 Low (L) issues. Below this, a table lists projects with columns for 'Project', 'Imported', 'Tested', and 'Issues'. The table shows two projects: 'com.basicsstrong:securecoding' and 'demo.secure.code.db:demo.secure.code.db'. At the bottom, a prompt 'Ready to import another project?' is displayed with the text 'Secure your entire stack with Snyk' and an 'Add projects' button.

Project	Imported	Tested	Issues
com.basicsstrong:securecoding	12 minutes ago	12 minutes ago	4 Critical, 22 High, 13 Medium, 6 Low
demo.secure.code.db:demo.secure.code.db	12 minutes ago	12 minutes ago	0 Critical, 1 High, 0 Medium, 0 Low

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.