

Lesson 09 Demo 01

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

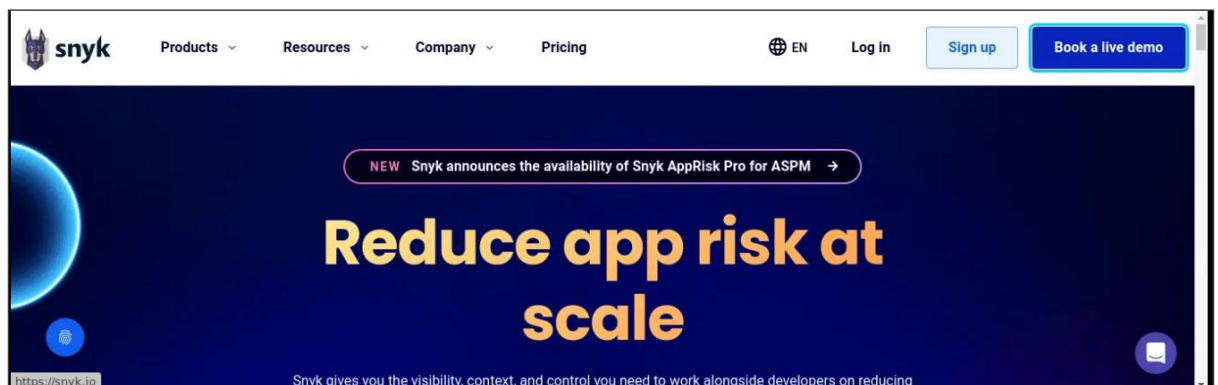
Prerequisites: None

Steps to be followed:

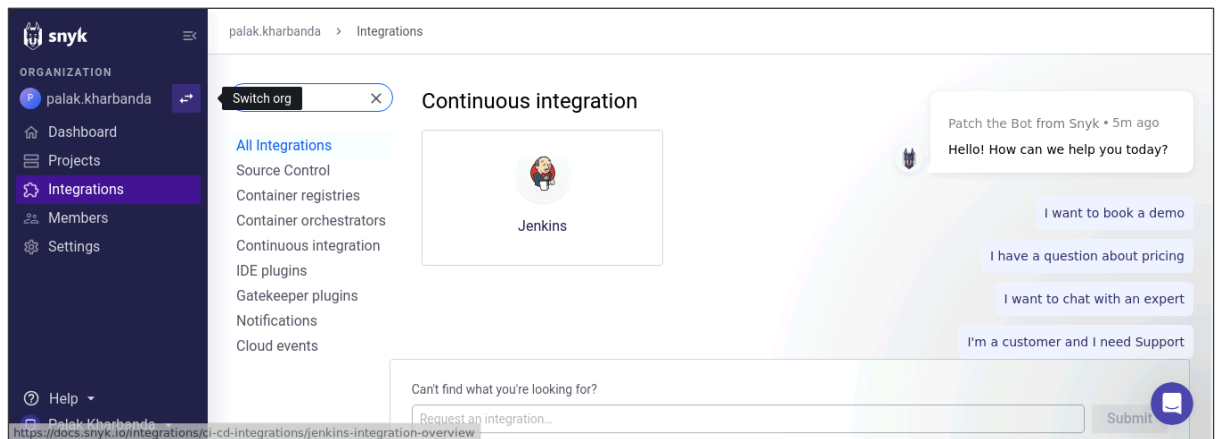
1. Configure Snyc as a SAST scan tool
2. Create and configure a Jenkins job for Snyc integration
3. Manage Snyc API and Jenkins credentials
4. Configure the Jenkins job for scanning

Step 1: Configure Snyc as a SAST scan tool

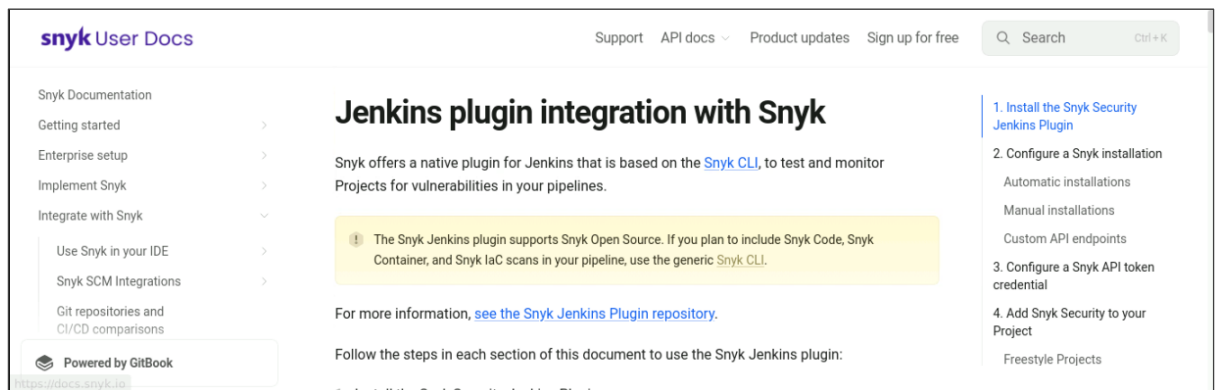
- 1.1 Visit <https://snyk.io/>, sign up for a new Snyc account, and log in



1.2 Navigate to **Integrations** and select **Jenkins**

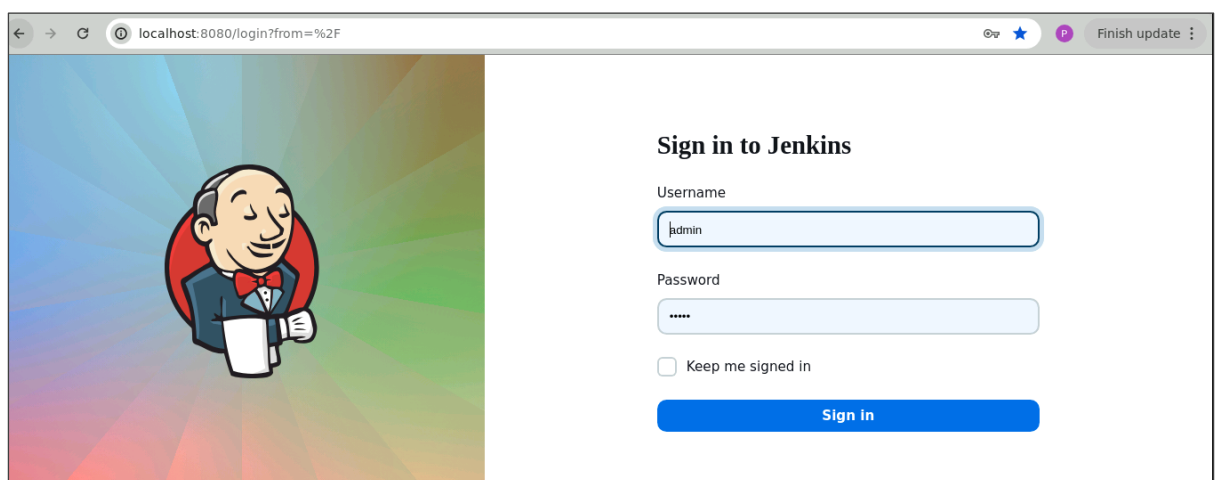


This will direct you to the documentation for integrating Snyk with Jenkins.



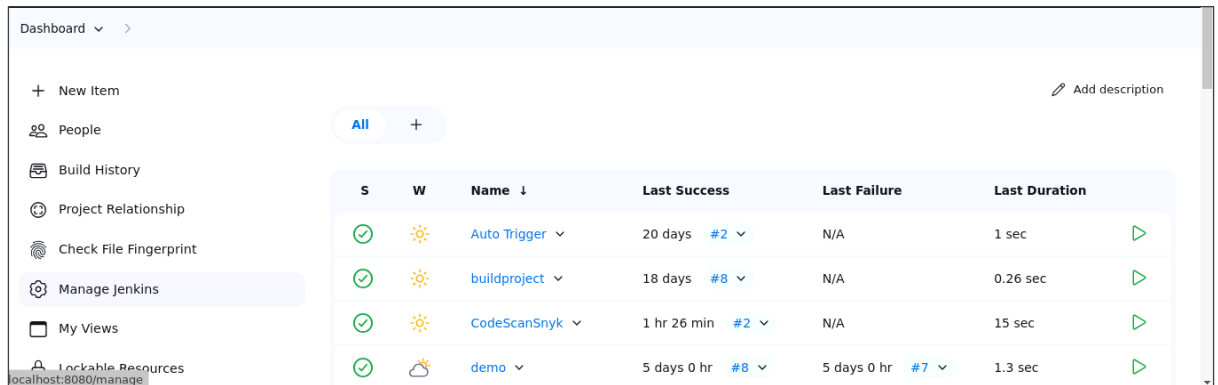
Step 2: Create and configure a Jenkins job for Snyk integration

2.1 Open Jenkins and log in to the Jenkins account:



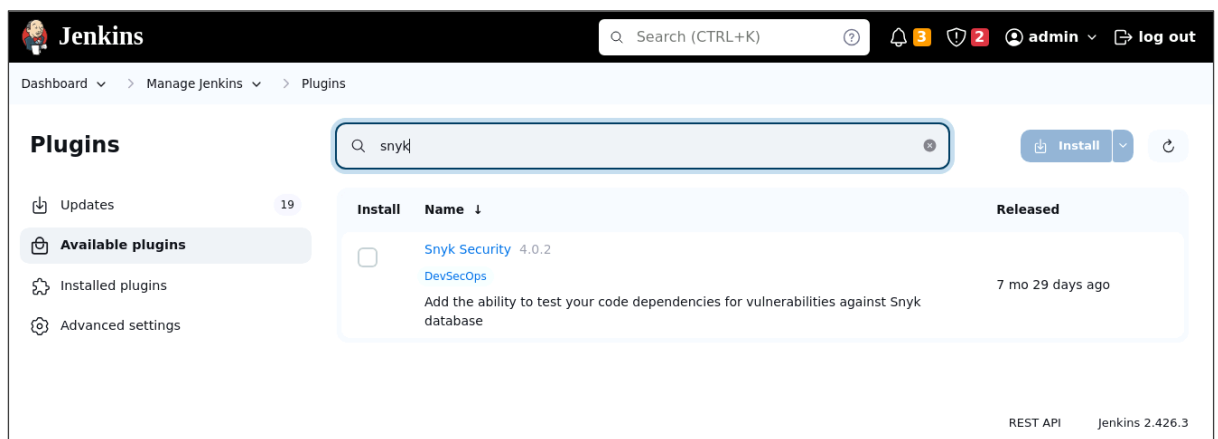
Note: The credentials for accessing Jenkins in the lab are Username: **admin** and Password: **Root123\$**.

2.2 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 'Manage Jenkins' page. On the left is a sidebar with navigation links: New Item, People, Build History, Project Relationship, Check File Fingerprint, Manage Jenkins (highlighted), My Views, and Lockable Resources. The main area displays a table of installed plugins. The table has columns: S (Status), W (Warning), Name, Last Success, Last Failure, and Last Duration. Four plugins are listed: Auto Trigger, buildproject, CodeScanSnyk, and demo.

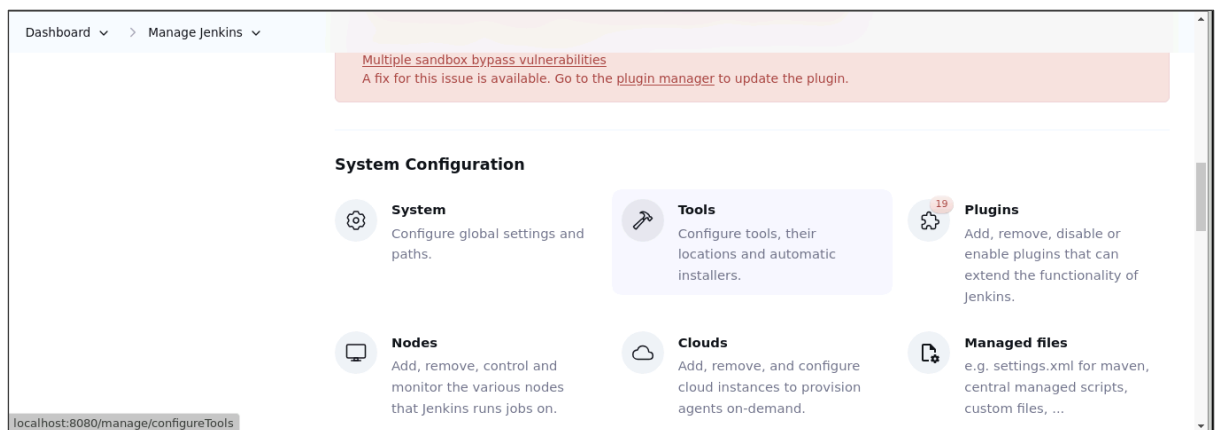
S	W	Name ↓	Last Success	Last Failure	Last Duration
✓	☀	Auto Trigger	20 days #2	N/A	1 sec
✓	☀	buildproject	18 days #8	N/A	0.26 sec
✓	☀	CodeScanSnyk	1 hr 26 min #2	N/A	15 sec
✓	☀	demo	5 days 0 hr #8	5 days 0 hr #7	1.3 sec



The screenshot shows the Jenkins 'Plugins' page. The top navigation bar includes the Jenkins logo, a search bar, and user information. The left sidebar has links for Updates, Available plugins (highlighted), Installed plugins, and Advanced settings. The main area shows a search for 'snyk' with an 'Install' button. Below the search bar is a table of available plugins.

Install	Name ↓	Released
<input type="checkbox"/>	Snyk Security 4.0.2 DevSecOps Add the ability to test your code dependencies for vulnerabilities against Snyk database	7 mo 29 days ago

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



The screenshot shows the Jenkins 'Global Tool Configuration' page. At the top, there is a red warning banner about sandbox vulnerabilities. Below this is the 'System Configuration' section with five cards: System, Tools (highlighted), Plugins, Nodes, and Clouds. Each card has a brief description of its function.

System Configuration

- System**: Configure global settings and paths.
- Tools**: Configure tools, their locations and automatic installers.
- 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.
- Managed files**: e.g. settings.xml for maven, central managed scripts, custom files, ...

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

Dashboard > Manage Jenkins > Tools

Maven installations

Add Maven

Snyk installations

Add Snyk

Save Apply

Dashboard > Manage Jenkins > Tools

Maven

Name

Required

☒ Install automatically ?

Install from Apache

Version

Save Apply

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

The screenshot shows the Jenkins 'Manage Jenkins' > 'Tools' page. At the top, there's a breadcrumb trail: 'Dashboard > Manage Jenkins > Tools'. Below this, there's a section for 'Add Maven'. Further down, the 'Snyk installations' section is visible. It contains a button labeled 'Add Snyk', which is highlighted with a red rectangle. Below this button are two other buttons: 'Save' (in blue) and 'Apply' (in light blue).

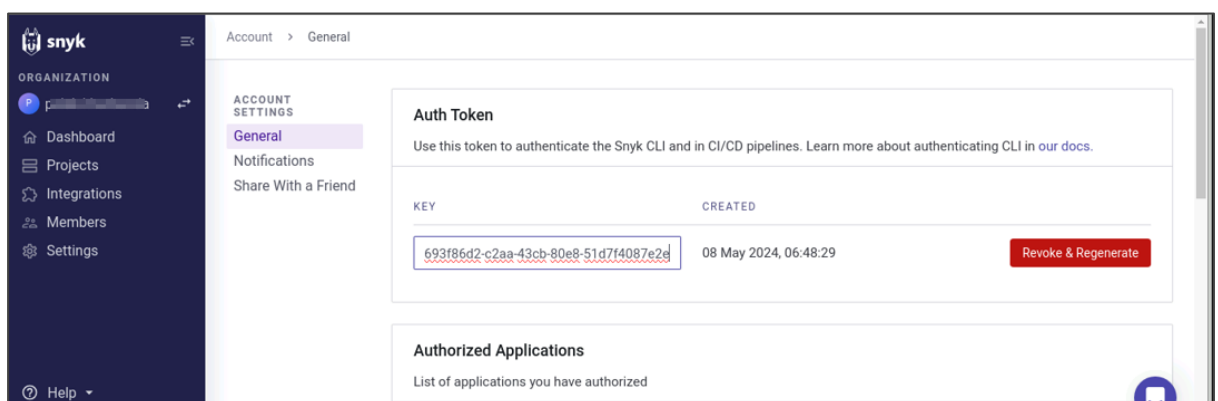
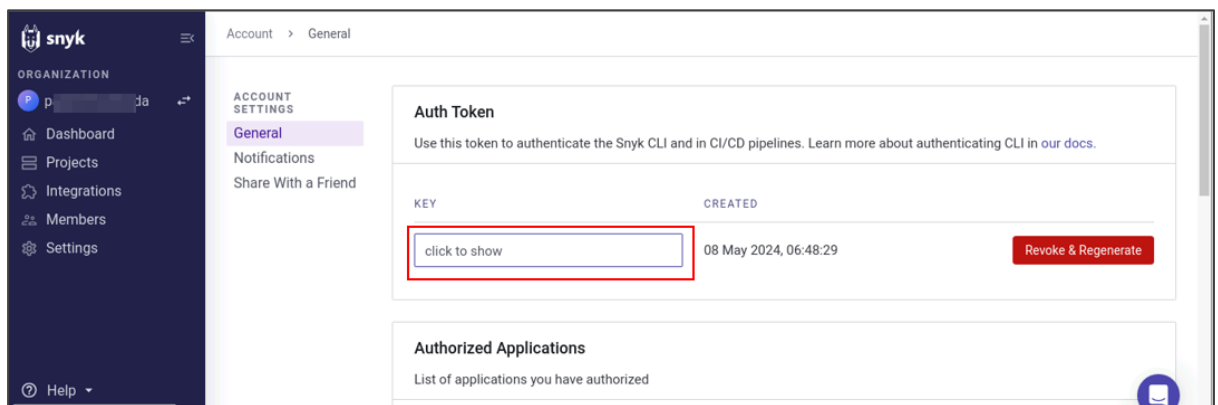
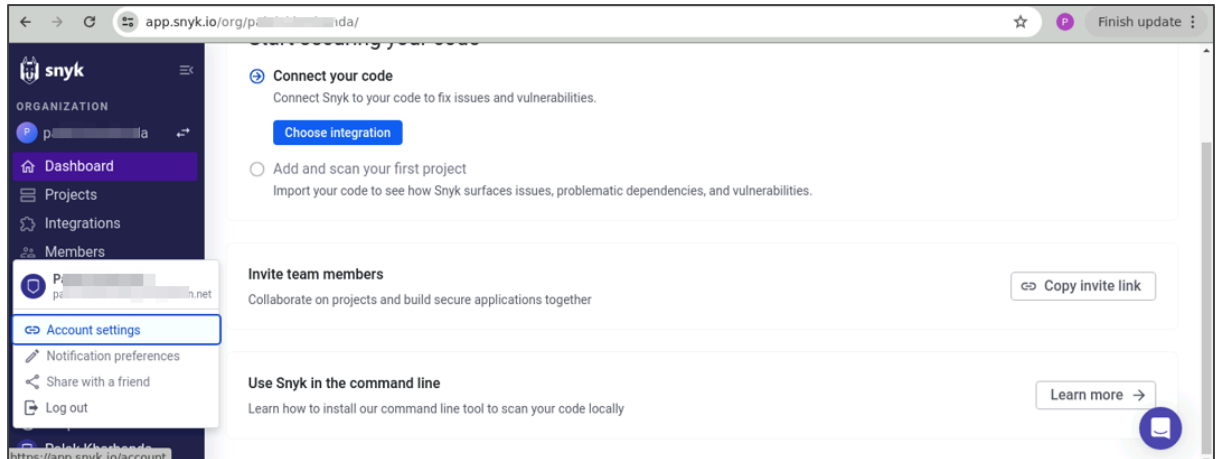
This screenshot shows the configuration form for adding a new Snyk installation. The form is titled 'Snyk' and has a close button (X) in the top right corner. It contains the following fields and options:

- Name:** A text input field containing the value 'Synk', highlighted with a red rectangle. Below the field is a red error message: 'Required'.
- Install automatically:** A checkbox that is checked, with a help icon (?) next to it.
- Install from snyk.io:** A sub-section with a close button (X) in its top right corner.
 - Version:** A text input field containing the value 'latest'.

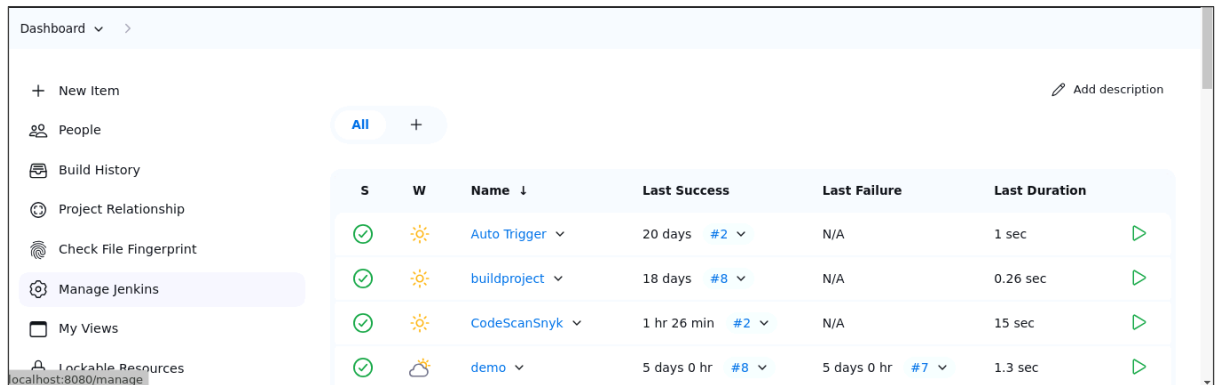
At the bottom of the form, there are two buttons: 'Save' (in blue, highlighted with a red rectangle) and 'Apply' (in light blue).

Step 3: Manage Snky API and Jenkins credentials

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



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



Dashboard >

+ New Item Add description

People All +

Build History

Project Relationship

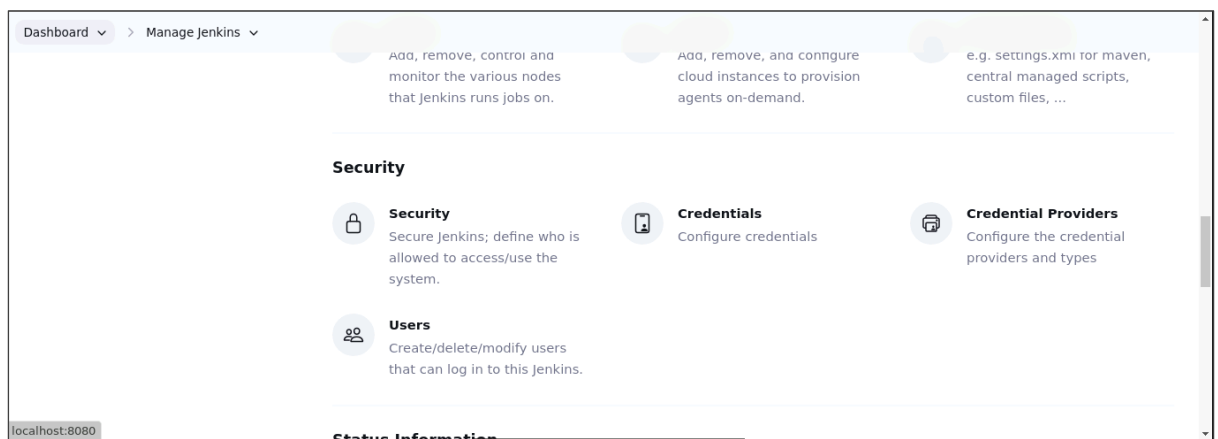
Check File Fingerprint

Manage Jenkins

My Views

Lockable Resources

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 #8 ↓	5 days 0 hr #7 ↓	1.3 sec ▶



Dashboard > Manage Jenkins >

Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

Add, remove, and configure cloud instances to provision agents on-demand.

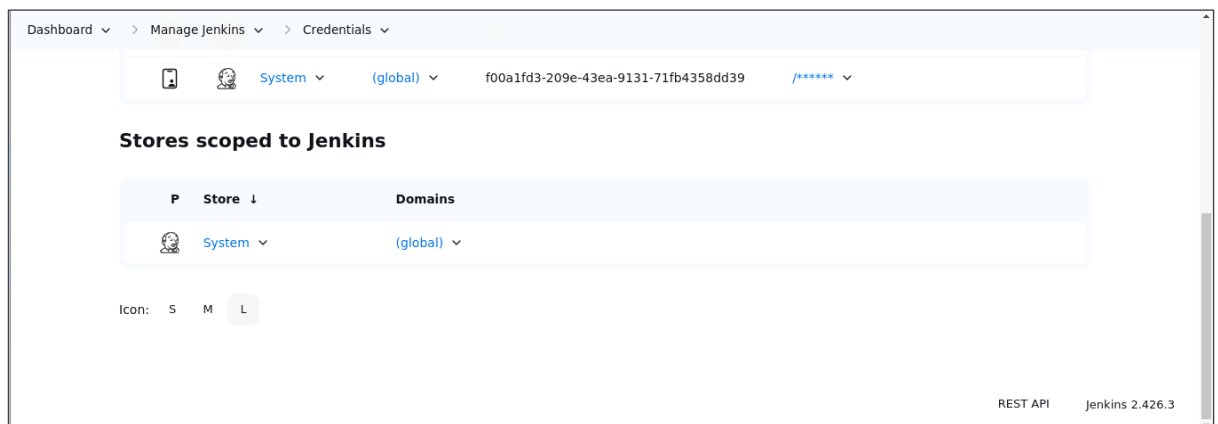
e.g. settings.xml for maven, central managed scripts, custom files, ...

Security

- Security**
Secure Jenkins; define who is allowed to access/use the system.
- Credentials**
Configure credentials
- Credential Providers**
Configure the credential providers and types
- Users**
Create/delete/modify users that can log in to this Jenkins.

localhost:8080

Status Information



Dashboard > Manage Jenkins > Credentials >

System (global) f00a1fd3-209e-43ea-9131-71fb4358dd39 /*****

Stores scoped to Jenkins

P	Store ↓	Domains
System	(global)	

Icon: S M L

REST API Jenkins 2.426.3

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

Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >

Global credentials (unrestricted)

+ Add Credentials

Credentials that should be available irrespective of domain specification to requirements matching.

ID	Name	Kind	Description
geeks	admin/***** (geeks)	Username with password	geeks
geeks01	admin/***** (geeks01)	Username with password	geeks01
Maven	GithubResources1/***** (Mavemgit)	Username with password	Mavemgit
localhost:8080/manage/credentials/store/system/domain/_/newCredentials		GithubResources1/*****	Username with

Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >

Kind

Snyk API token

Username with password

GitHub App

SSH Username with private key

Secret file

Secret text

Snyk API token

Certificate

Token ?

Field is required

ID ?

Create

Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >

Scope ?

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

Token ?

ID ?

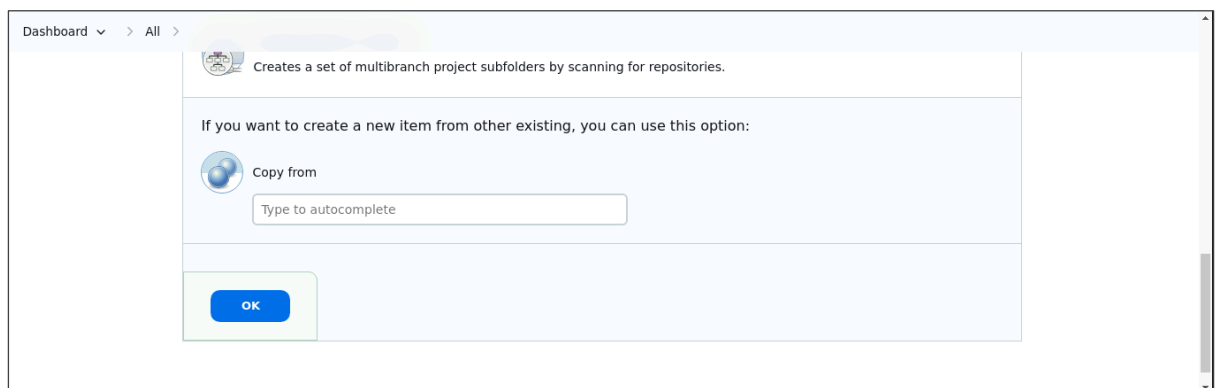
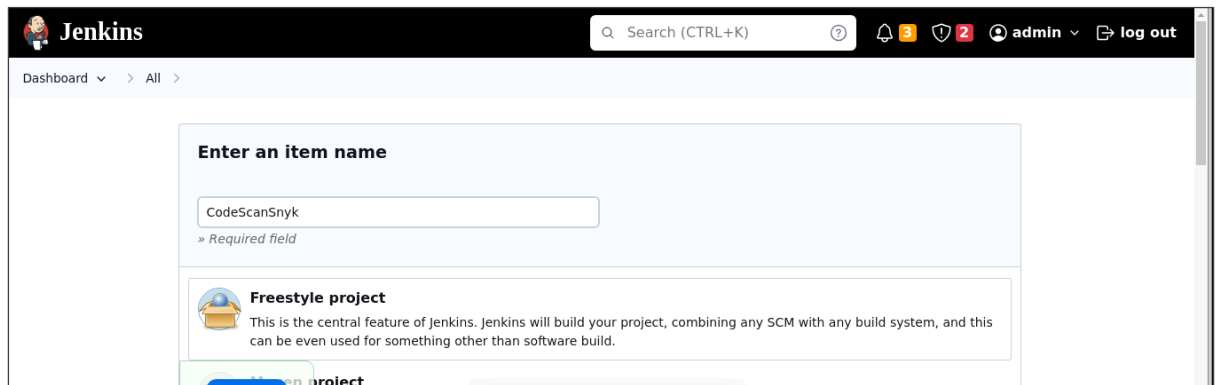
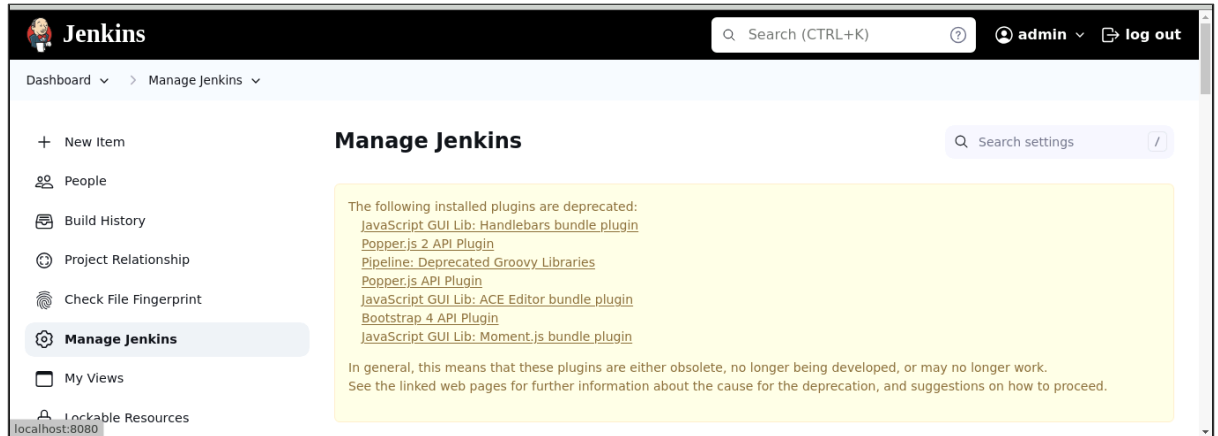
Description ?

SynkToken

Create

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 Snky Security task** with the name **SnykToken**. Finally, click the **Save** button to create the build.

Dashboard > CodeScanSnyk > Configuration

Configure

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

Repository URL ?

Please enter Git repository.

Credentials ?
- none -

+ Add

Advanced

Save Apply

Dashboard > CodeScanSnyk > Configuration

Configure

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

Build Steps

Add build step

Filter

- Execute Windows batch command
- Execute shell
- Invoke Ant
- Invoke Gradle script
- Invoke Snky Security task
- Invoke top-level Maven targets
- Provide Configuration files

Dashboard > CodeScanSnyk > Configuration

Configure

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

☒ Fail the build if errors occur ?

☒ Monitor project on build ?

Snyk API token ?

+ Add

Target file ?

Save Apply

Note: For GitHub repository URL, use <https://github.com/anujdevopslearn/MavenBuild>

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

Dashboard > CodeScanSnyk >

Changes

Workspace

Build Now

Configure

Delete Project

Rename

Build History trend

Filter builds...

Last Successful Artifacts

2024-05-08T09-24-17-848173830Z_snyk_report.html 13.79 KB view

Permalinks

- Last build (#2), 2 min 46 sec ago
- Last stable build (#2), 2 min 46 sec ago
- Last successful build (#2), 2 min 46 sec ago
- Last completed build (#2), 2 min 46 sec ago

Dashboard > CodeScanSnyk > #2

Status #2 (May 8, 2024, 9:24:10 AM) Keep this build forever

Changes

Console Output

Edit Build Information

Delete build '#2'

Git Build Data

Build Artifacts

2024-05-08T09-24-17-848173830Z_snyk_report.html 13.79 KB view

No changes.

Started by user admin (2 times)

Started 3 min 17 sec ago
Took 15 sec

Revision: 644ef597e4027355d708b2e254ace162844f

Jenkins

Search (CTRL+K)

admin log out

Dashboard > CodeScanSnyk > #2 > Console Output

Console Output

Started by user admin

Started by user admin

Running as SYSTEM

Building in workspace /var/lib/jenkins/workspace/CodeScanSnyk

The recommended git tool is: NONE

No credentials specified

> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/CodeScanSnyk/.git # timeout=10

Fetching changes from the remote Git repository

> git config remote.origin.url https://github.com/anujdevopslearn/MavenBuild # timeout=10

Fetching upstream changes from https://github.com/anujdevopslearn/MavenBuild

Dashboard > CodeScanSnyk > #2 > Console Output

```
/var/lib/jenkins/workspace/CodeScanSnyk/2024-05-08T09-24-17-848173830Z_snyk_report.json
Archiving artifacts
Monitoring project...
> /var/lib/jenkins/tools/io.snyk.jenkins.tools.SnykInstallation/Synk/snyk-linux monitor --severity-threshold=low

Monitoring /var/lib/jenkins/workspace/CodeScanSnyk (com.java.example:java-example)...

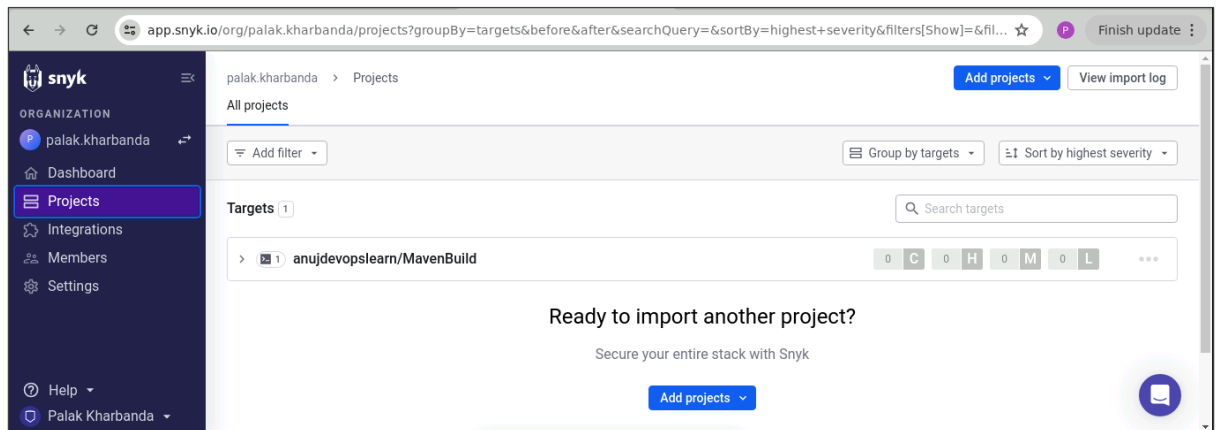
Explore this snapshot at https://app.snyk.io/org/palak.kharbanda/project/c88c922e-e55a-465f-91c7-f196291da77c/history/f6e31b9d-b848-43ef-bd73-e38bccf62ca4

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

Finished: SUCCESS
```

REST API Jenkins 2.426.3

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



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.