Name – Rachit Shrivastava

Sap Id – 500119571

Batch 3 – DevOps

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

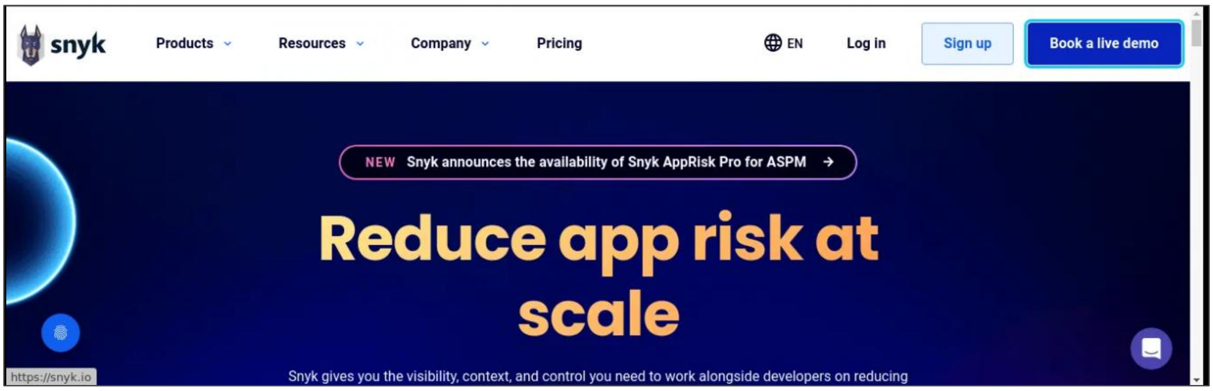
Prerequisites: None

Steps to be followed:

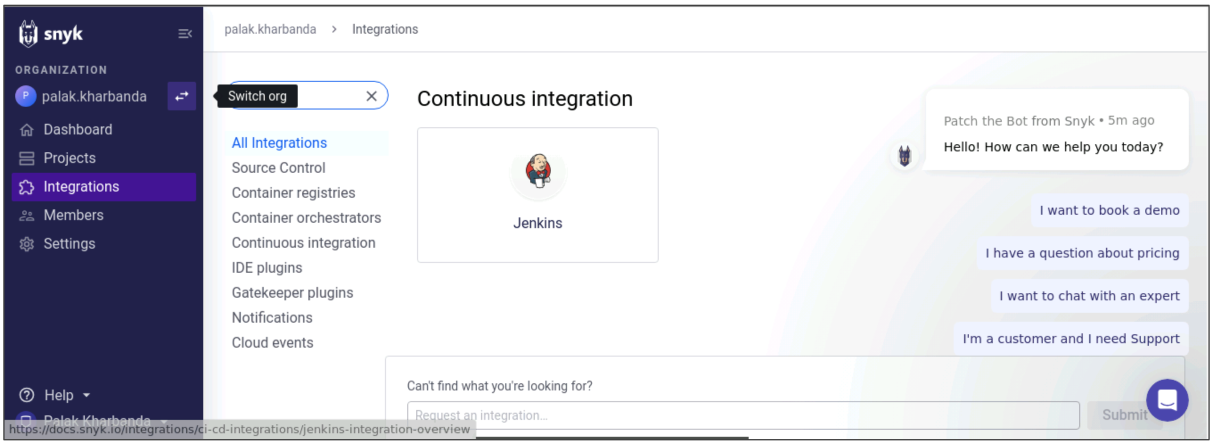
1. Configure Snyk as a SAST scan tool
2. Create and configure a Jenkins job for Snyk integration
3. Manage Snyk API and Jenkins credentials
4. Configure the Jenkins job for scanning

Step 1: Configure Snyk as a SAST scan tool

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


snyk
Products
Resources
Company
Pricing
EN
Log in
Sign up
Book a live demo
NEW Snyk announces the availability of Snyk AppRisk Pro for ASPM
Reduce app risk at scale
Snyk gives you the visibility, context, and control you need to work alongside developers on reducing
https://snyk.io

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



snyk

ORGANIZATION

palak.kharbanda

Switch org

Dashboard

Projects

Integrations

Members

Settings

Help

Palak Kharbanda

palak.kharbanda > Integrations

All Integrations

Source Control

Container registries

Container orchestrators

Continuous integration

IDE plugins

Gatekeeper plugins

Notifications

Cloud events

Continuous integration

Jenkins

Patch the Bot from Snyk • 5m ago

Hello! How can we help you today?

I want to book a demo

I have a question about pricing

I want to chat with an expert

I'm a customer and I need Support

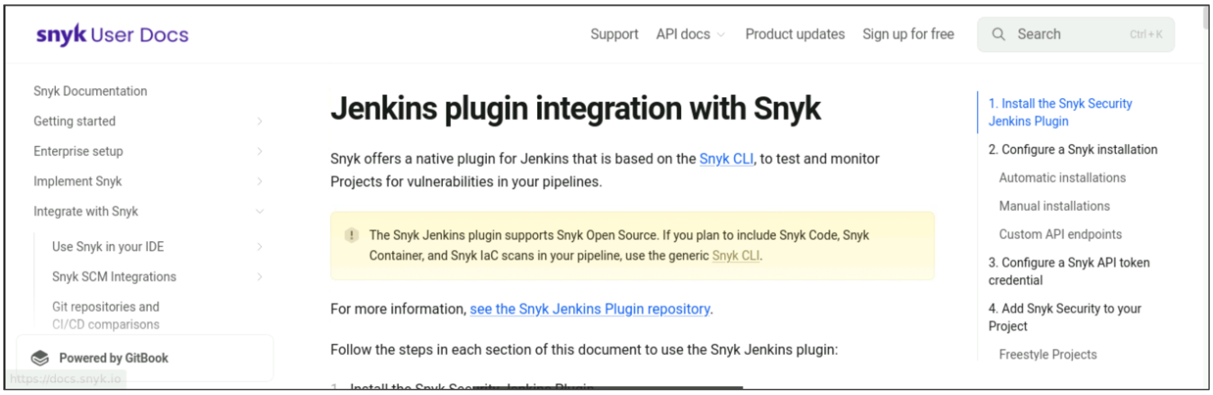
Can't find what you're looking for?

Request an integration...

Submit

https://docs.snyk.io/integrations/ci-cd-integrations/jenkins-integration-overview

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



snyk User Docs

Support API docs Product updates Sign up for free Search Ctrl + K

Snyk Documentation

Getting started

Enterprise setup

Implement Snyk

Integrate with Snyk

Use Snyk in your IDE

Snyk SCM Integrations

Git repositories and CI/CD comparisons

Powered by GitBook

https://docs.snyk.io

Jenkins plugin integration with Snyk

Snyk offers a native plugin for Jenkins that is based on the Snyk CLI, to test and monitor Projects for vulnerabilities in your pipelines.

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.

For more information, see the Snyk Jenkins Plugin repository.

Follow the steps in each section of this document to use the Snyk Jenkins plugin:

1. Install the Snyk Security Jenkins Plugin

1. Install the Snyk Security Jenkins Plugin

2. Configure a Snyk installation

Automatic installations

Manual installations

Custom API endpoints

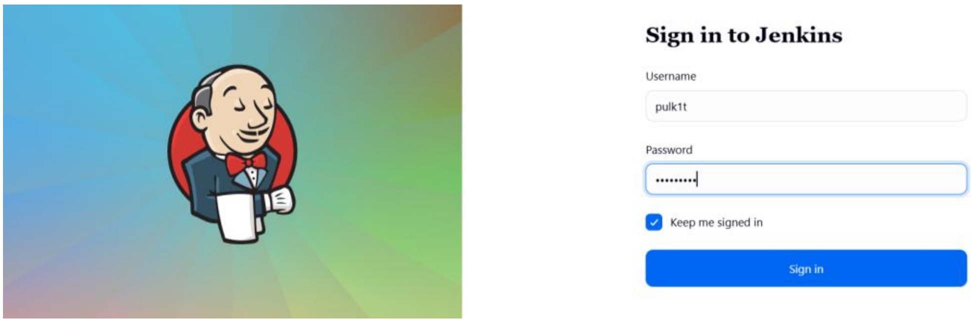
3. Configure a Snyk API token credential

4. Add Snyk Security to your Project

Freestyle Projects

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

2.1 Open Jenkins and log in to the Jenkins account:



Sign in to Jenkins

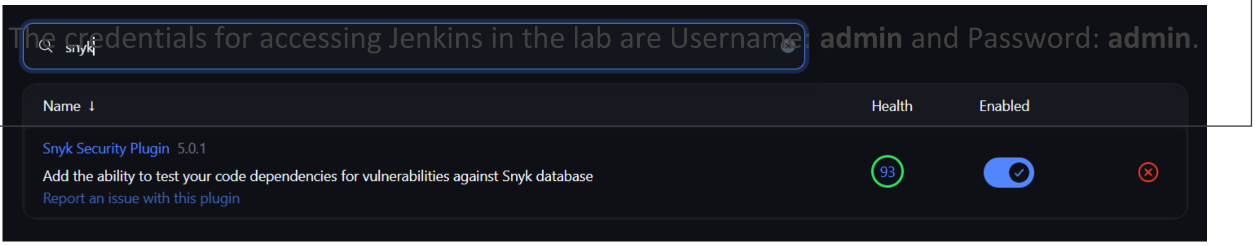
Username
pulk1t

Password
••••••••

☑ Keep me signed in

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

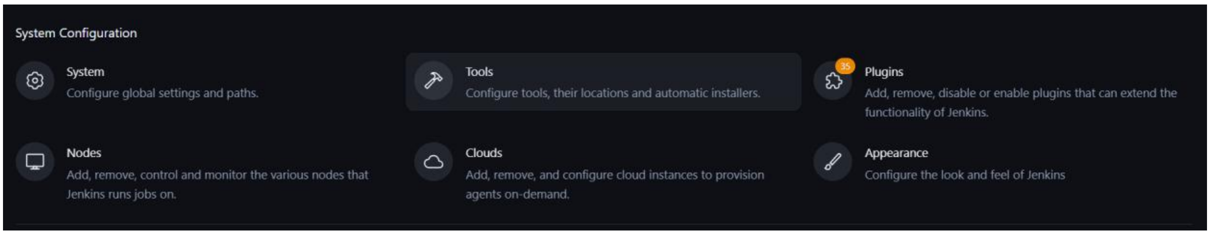
Note: The credentials for accessing Jenkins in the lab are Username: **admin** and Password: **admin**.



snyk

Name ↓	Health	Enabled
Snyk Security Plugin 5.0.1 Add the ability to test your code dependencies for vulnerabilities against Snyk database Report an issue with this plugin	93	✓

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



System Configuration

System
Configure global settings and paths.

Tools
Configure tools, their locations and automatic installers.

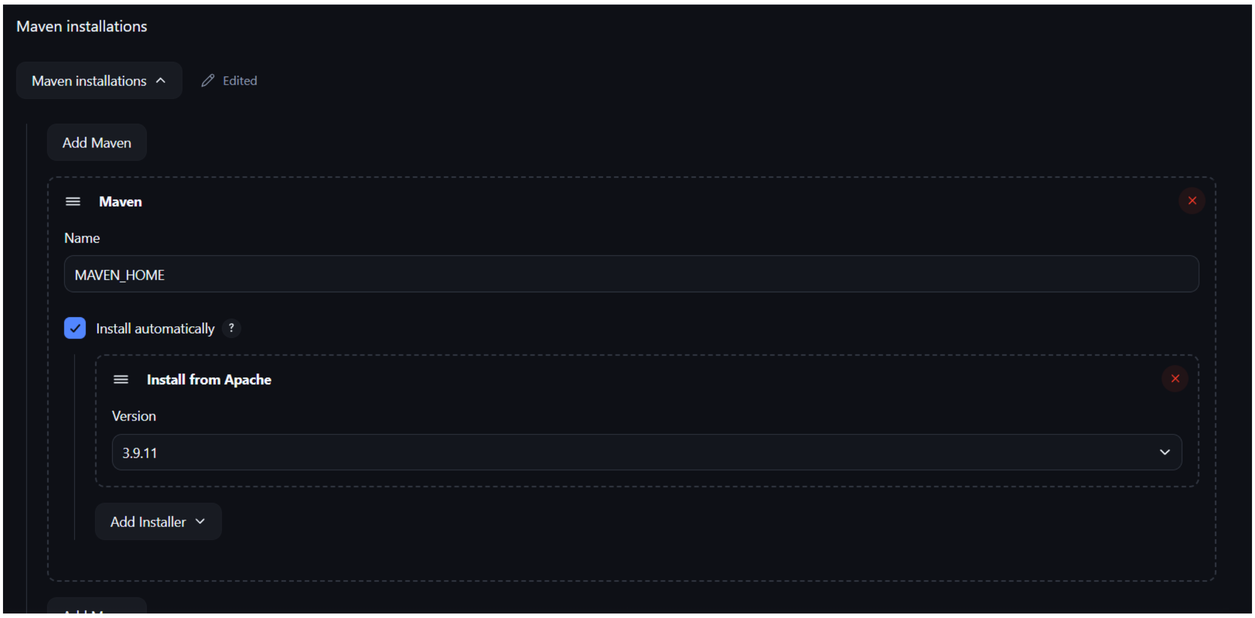
Plugins
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

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.

Appearance
Configure the look and feel of Jenkins

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



Maven installations

Name

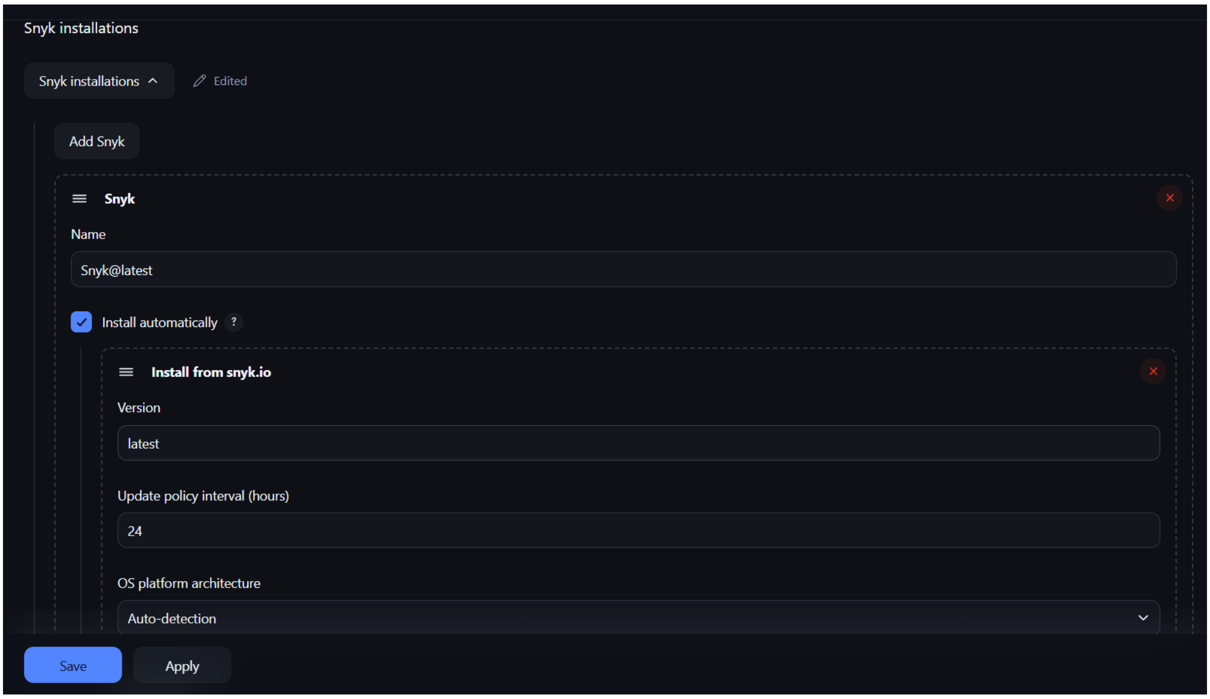
MAVEN_HOME

☑ Install automatically

Version

3.9.11

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



Snyk installations

Name

Snyk@latest

☑ Install automatically

Version

latest

Update policy interval (hours)

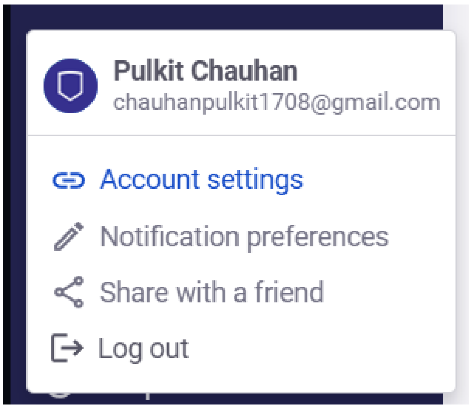
24

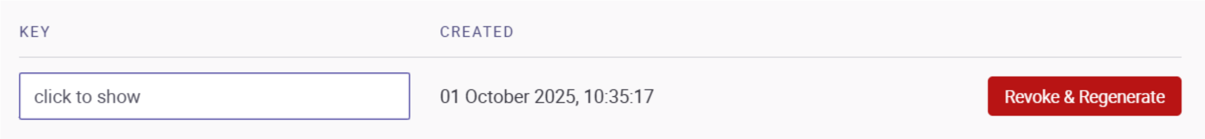
OS platform architecture

Auto-detection

Step 3: Manage Snyk API and Jenkins credentials

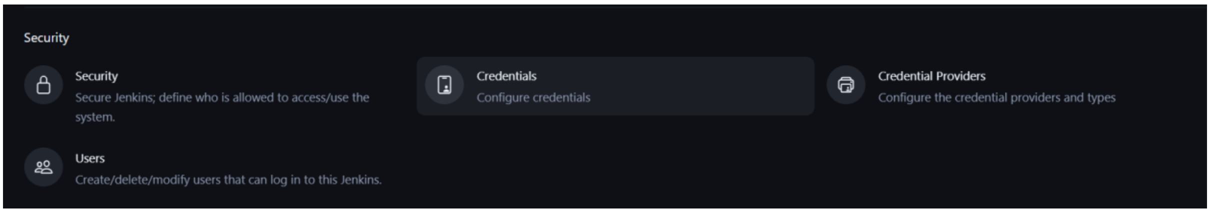
3.1 To retrieve your Snyk API token, go to **Account Settings** in your Snyk account, click on **Click to show** under the Auth Token key field, and copy the token for further reference





KEY	CREATED	
click to show	01 October 2025, 10:35:17	Revoke & Regenerate

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



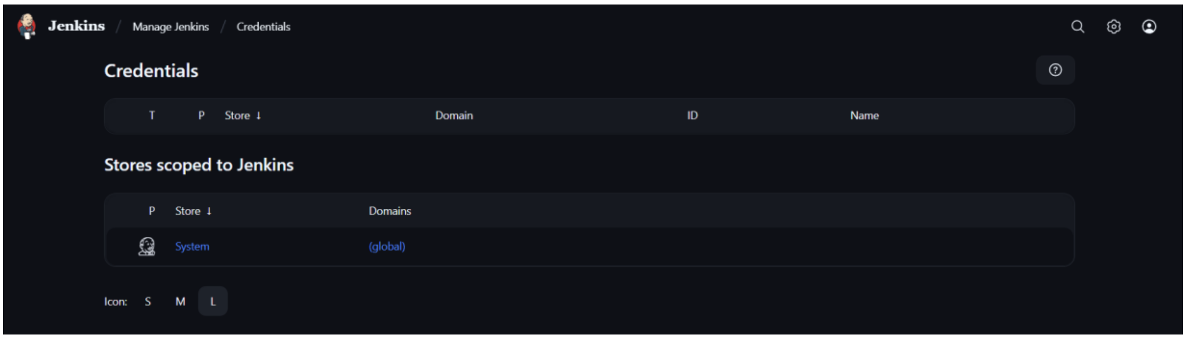
Security

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.



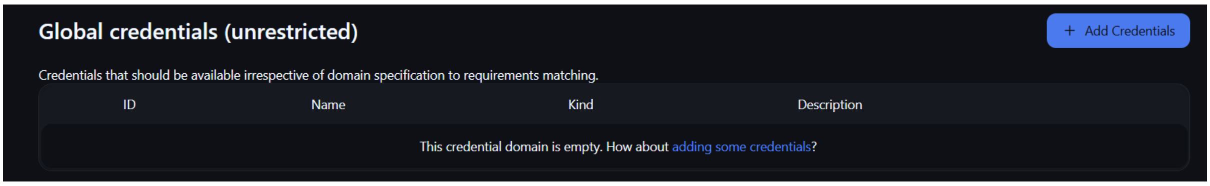
Credentials

T	P	Store ↓	Domain	ID	Name

Stores scoped to Jenkins

P	Store ↓	Domains
	System	(global)

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

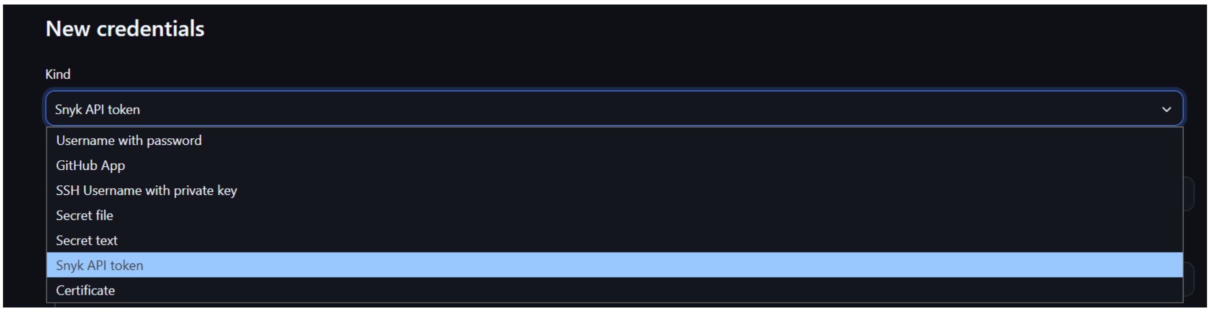


Global credentials (unrestricted)

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

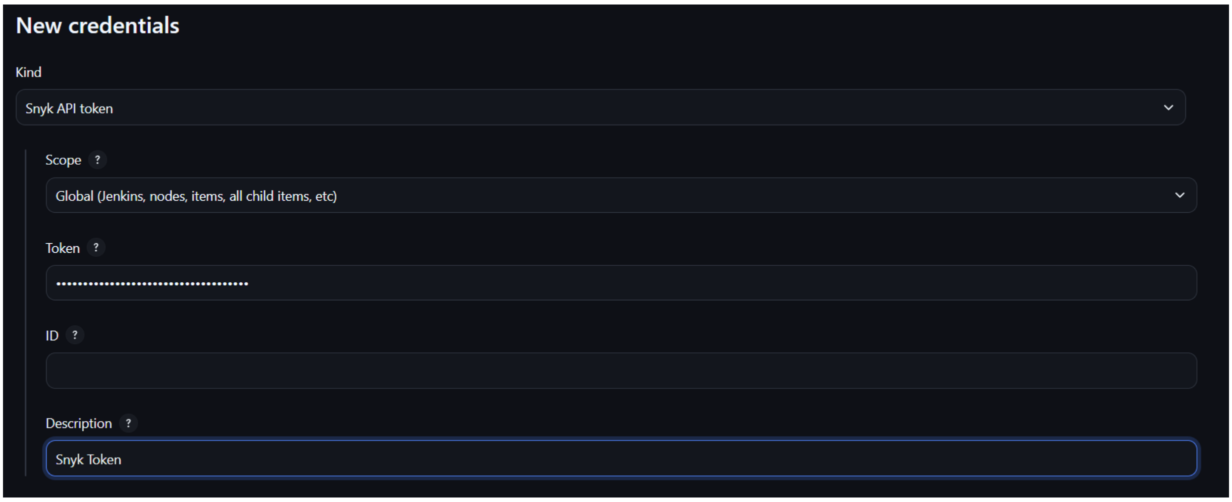
ID	Name	Kind	Description

This credential domain is empty. How about adding some credentials?



New credentials

Kind: Snyk API token



New credentials

Kind: Snyk API token

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

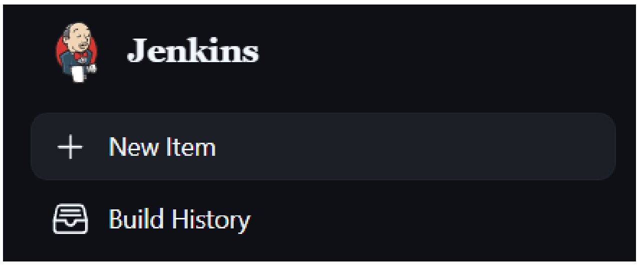
Token ?: ••••••••••••••••••••••••••••••••••••••

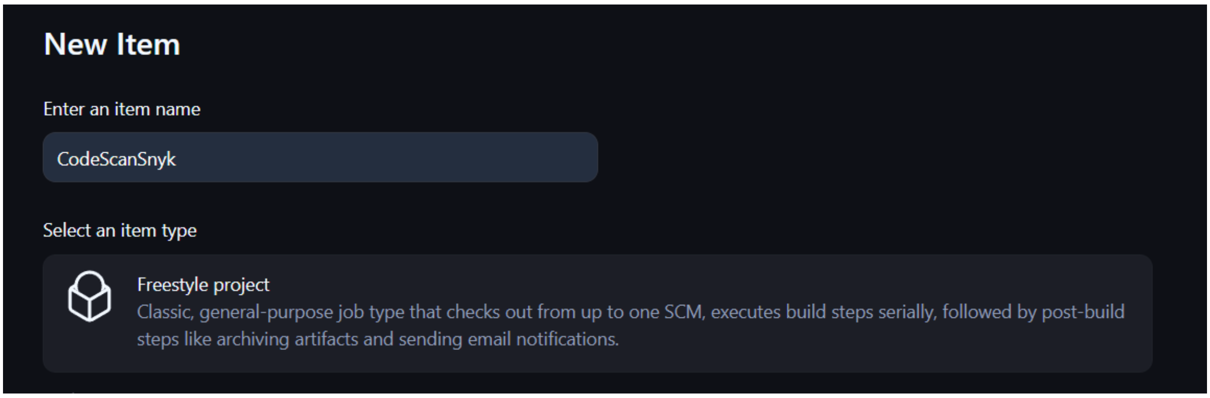
ID ?:

Description ?: Snyk Token

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**





New Item

Enter an item name

CodeScanSnyk

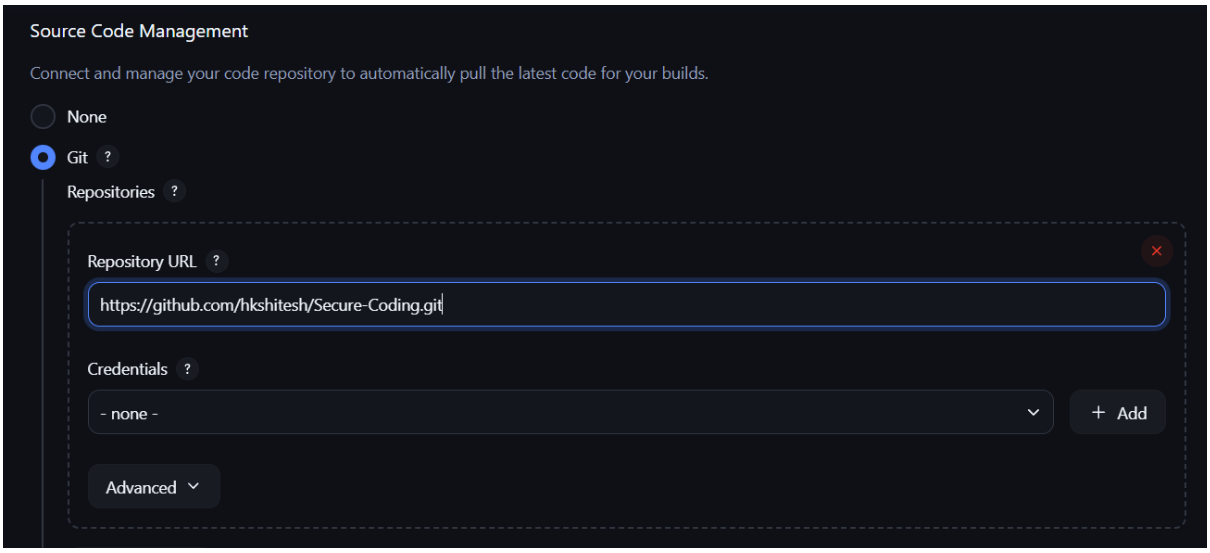
Select an item type

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.

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**



Source Code Management

Connect and manage your code repository to automatically pull the latest code for your builds.

○ None

● Git ?

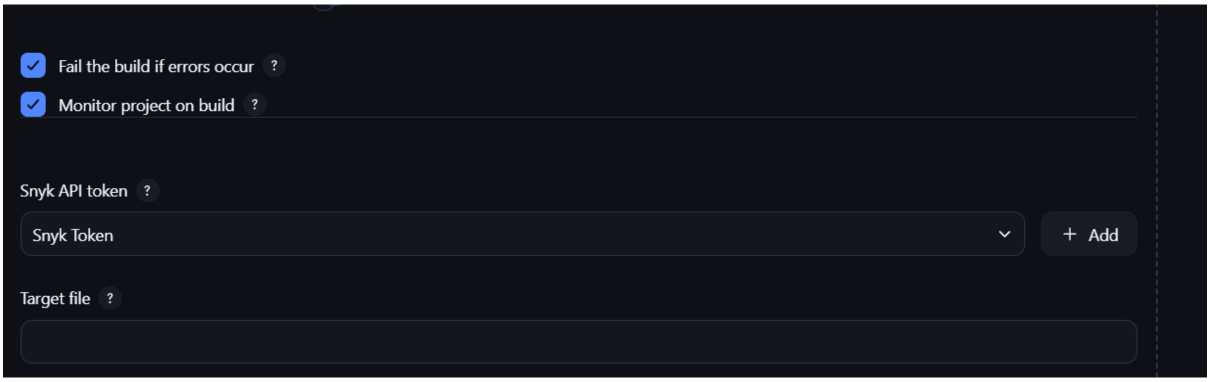
Repositories ?

Repository URL ?

https://github.com/hkshitesh/Secure-Coding.git

Credentials ?

- none -



☑ Fail the build if errors occur ?

☑ Monitor project on build ?

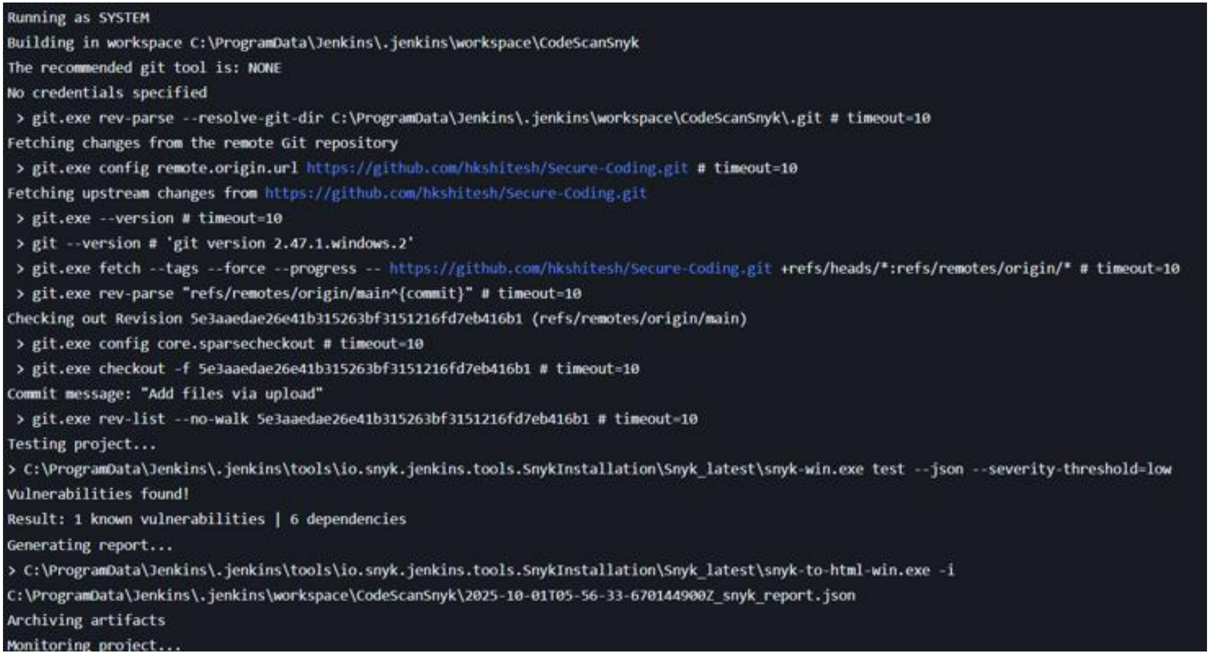
Snyk API token ?

Snyk Token

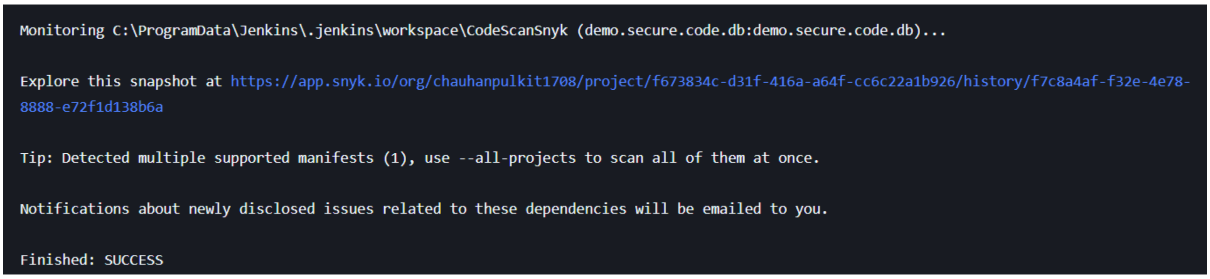
Target file ?

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**

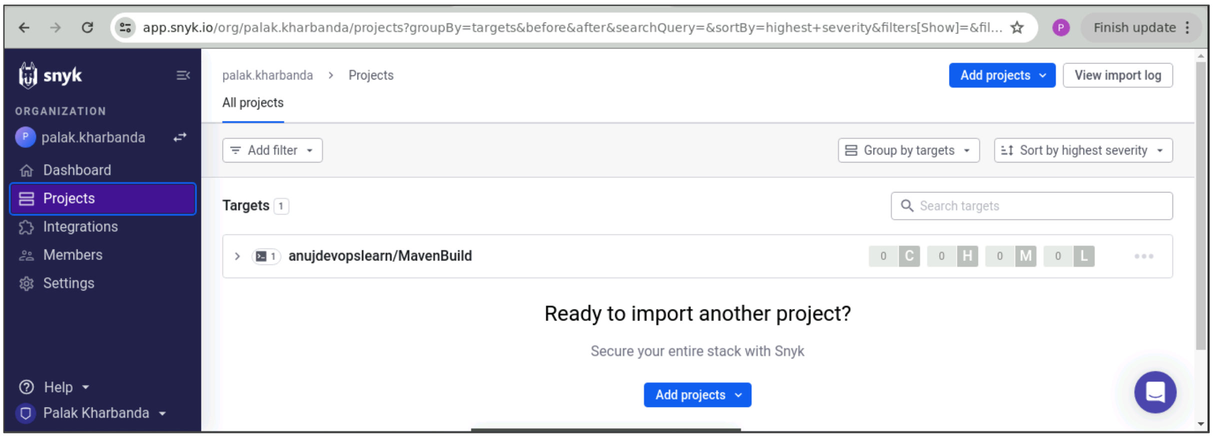


```
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\CodeScanSnyk
The recommended git tool is: NONE
No credentials specified
 > git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\CodeScanSnyk\.git # timeout=10
Fetching changes from the remote Git repository
 > git.exe 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.exe --version # timeout=10
 > git --version # 'git version 2.47.1.windows.2'
 > git.exe fetch --tags --force --progress -- https://github.com/hkshitesh/Secure-Coding.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git.exe rev-parse "refs/remotes/origin/main^{commit}" # timeout=10
Checking out Revision 5e3aaedae26e41b315263bf3151216fd7eb416b1 (refs/remotes/origin/main)
 > git.exe config core.sparsecheckout # timeout=10
 > git.exe checkout -f 5e3aaedae26e41b315263bf3151216fd7eb416b1 # timeout=10
Commit message: "Add files via upload"
 > git.exe rev-list --no-walk 5e3aaedae26e41b315263bf3151216fd7eb416b1 # timeout=10
Testing project...
> C:\ProgramData\Jenkins\.jenkins\tools\io.snyk.jenkins.tools.SnykInstallation\Snyk_latest\snyk-win.exe test --json --severity-threshold=low
Vulnerabilities found!
Result: 1 known vulnerabilities | 6 dependencies
Generating report...
> C:\ProgramData\Jenkins\.jenkins\tools\io.snyk.jenkins.tools.SnykInstallation\Snyk_latest\snyk-to-html-win.exe -i
C:\ProgramData\Jenkins\.jenkins\workspace\CodeScanSnyk\2025-10-01T05-56-33-670144900Z_snyk_report.json
Archiving artifacts
Monitoring project...
```



```
Monitoring C:\ProgramData\Jenkins\.jenkins\workspace\CodeScanSnyk (demo.secure.code.db:demo.secure.code.db)...

Explore this snapshot at https://app.snyk.io/org/chauhanpulkit1708/project/f673834c-d31f-416a-a64f-cc6c22a1b926/history/f7c8a4af-f32e-4e78-
8888-e72f1d138b6a

Tip: Detected multiple supported manifests (1), use --all-projects to scan all of them at once.

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

Finished: SUCCESS
```

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



app.snyk.io/org/palak.kharbanda/projects?groupBy=targets&before&after&searchQuery=&sortBy=highest+severity&filters[Show]=&fil...

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.