Lab Exercise 19

Setting up Snyk for SAST in Jenkins

Name: Viraj Bhidola
Sap id: 500121825
Batch 2 DevOps

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

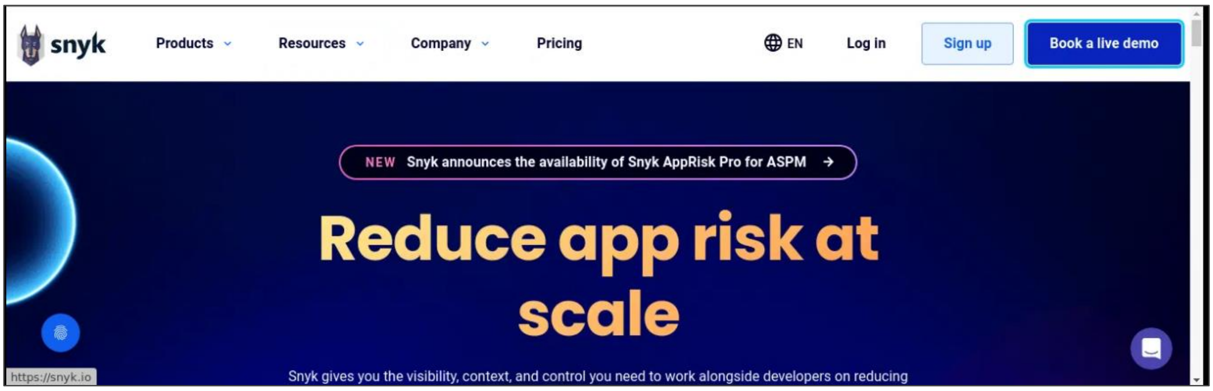
Prerequisites: None

Steps to be followed:

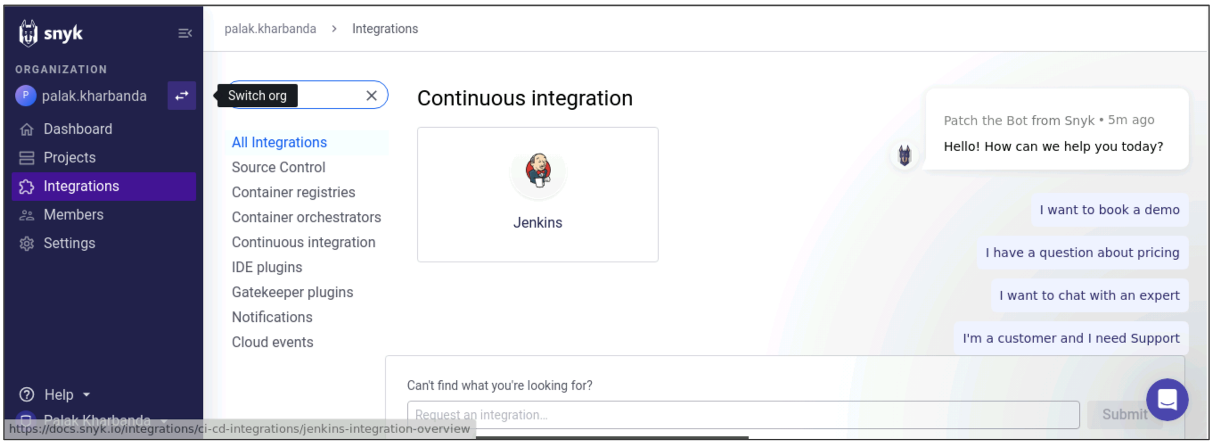
1. Configure Snyk as a SAST scan tool
2. Create and configure a Jenkins job for Snyk integration
3. Manage Snyk API and Jenkins credentials
4. Configure the Jenkins job for scanning

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

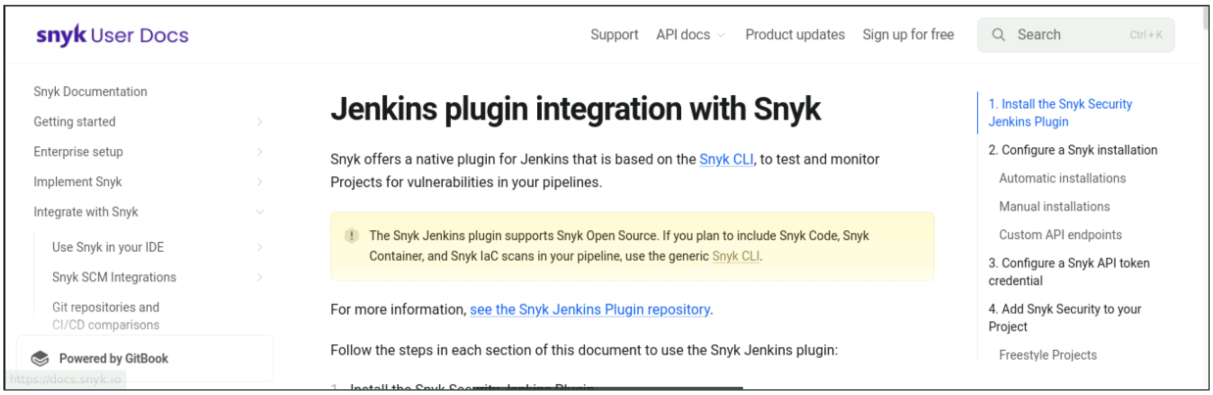
Can't find what you're looking for?

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

2. Configure a Snyk installation

Automatic installations

Manual installations

Custom API endpoints

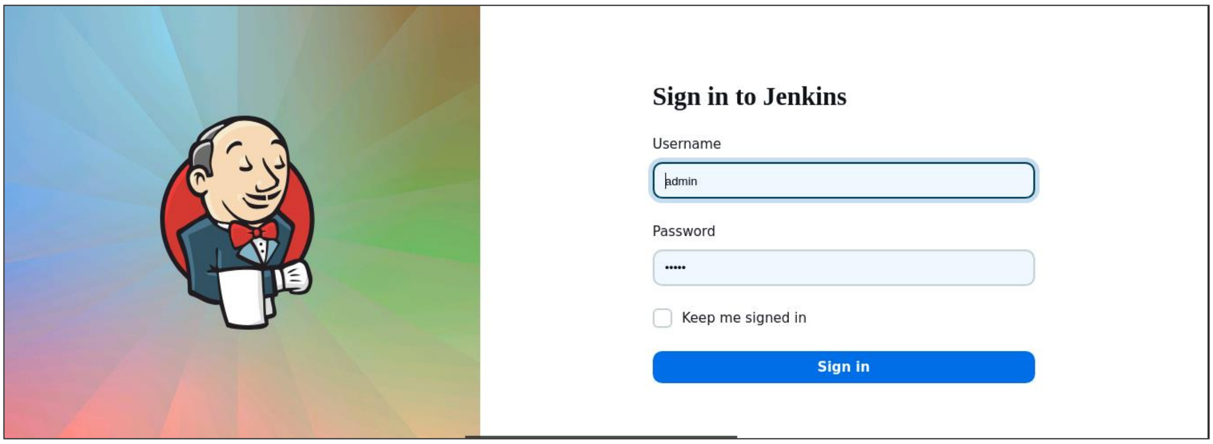
3. Configure a Snyk API token credential

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

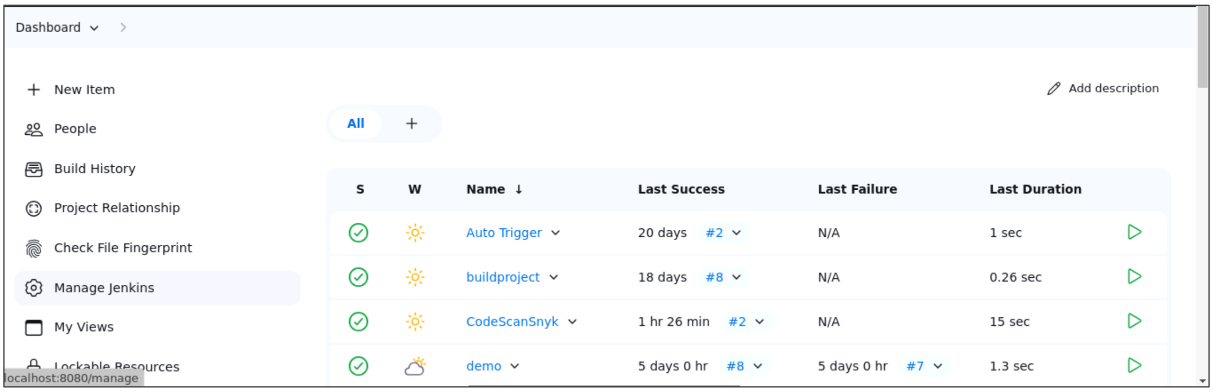
Username: admin

Password: •••••

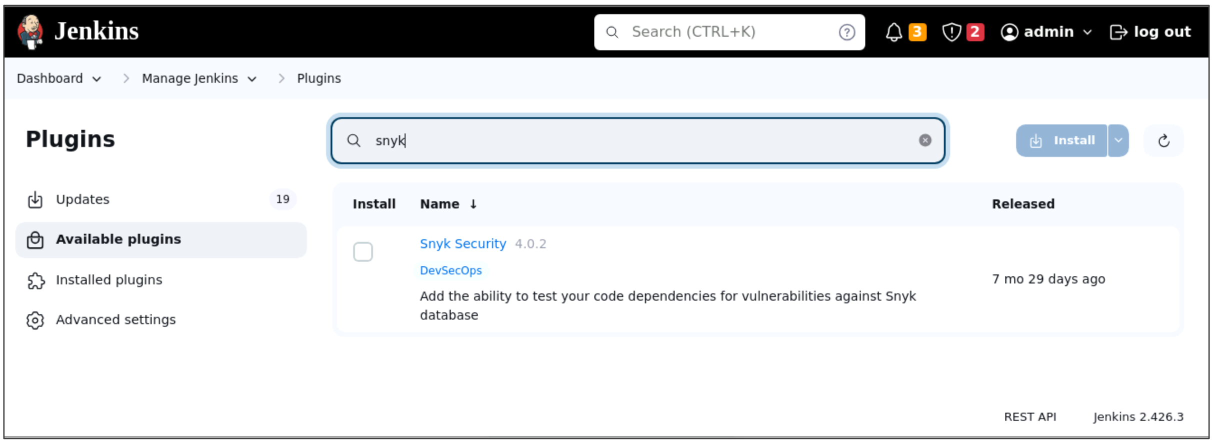
□ Keep me signed in

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

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



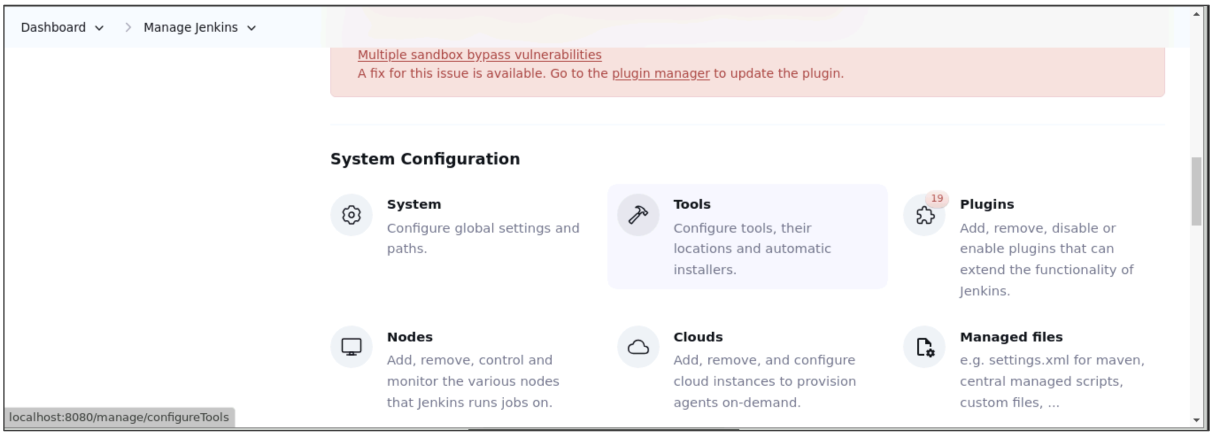
S	W	Name ↓	Last Success	Last Failure	Last Duration
✓	☀	Auto Trigger	20 days #2	N/A	1 sec
✓	☀	buildproject	18 days #8	N/A	0.26 sec
✓	☀	CodeScanSnyk	1 hr 26 min #2	N/A	15 sec
✓	☁	demo	5 days 0 hr #8	5 days 0 hr #7	1.3 sec



Search: snyk

Install	Name ↓	Released
□	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**



Dashboard > Manage Jenkins

Multiple sandbox bypass vulnerabilities
A fix for this issue is available. Go to the plugin manager to update the plugin.

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.

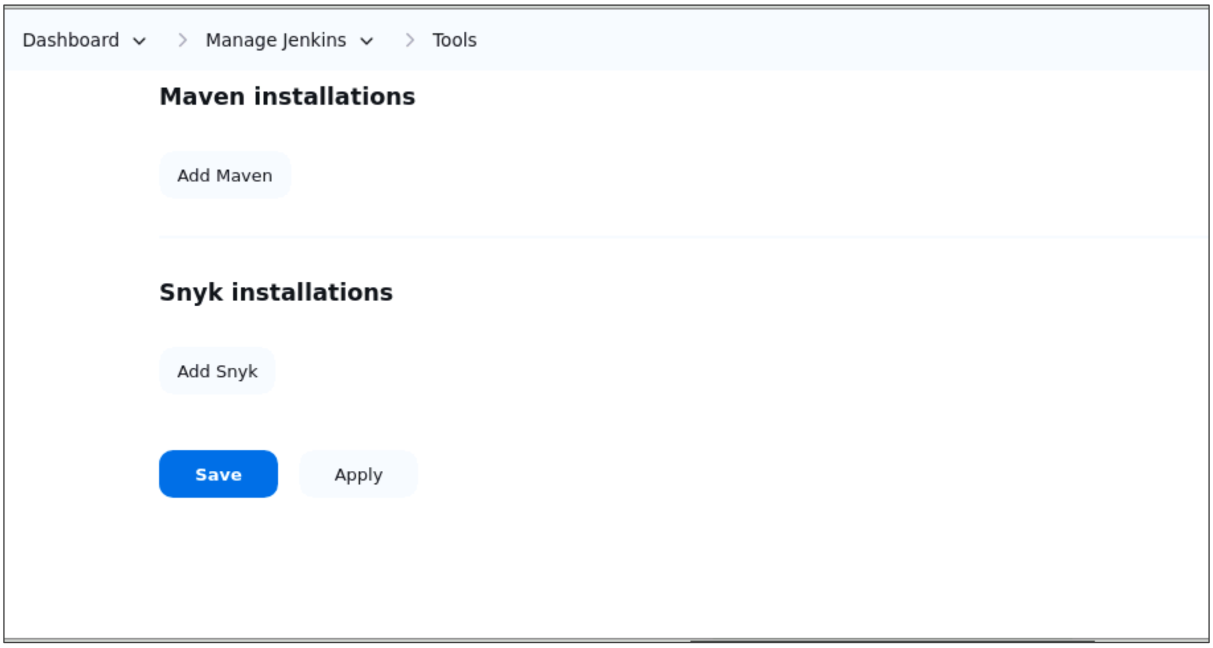
Nodes
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

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, ...

localhost:8080/manage/configureTools

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



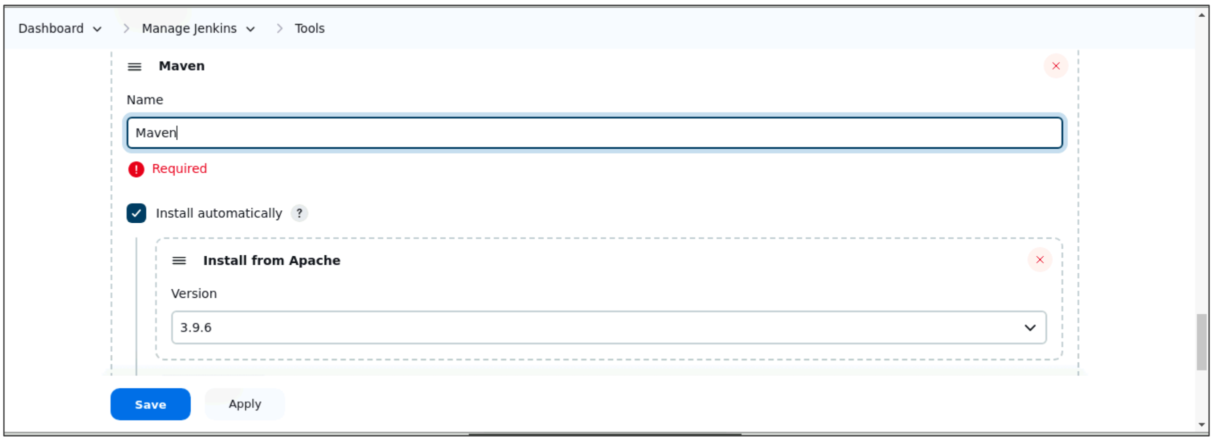
Dashboard > Manage Jenkins > Tools

Maven installations

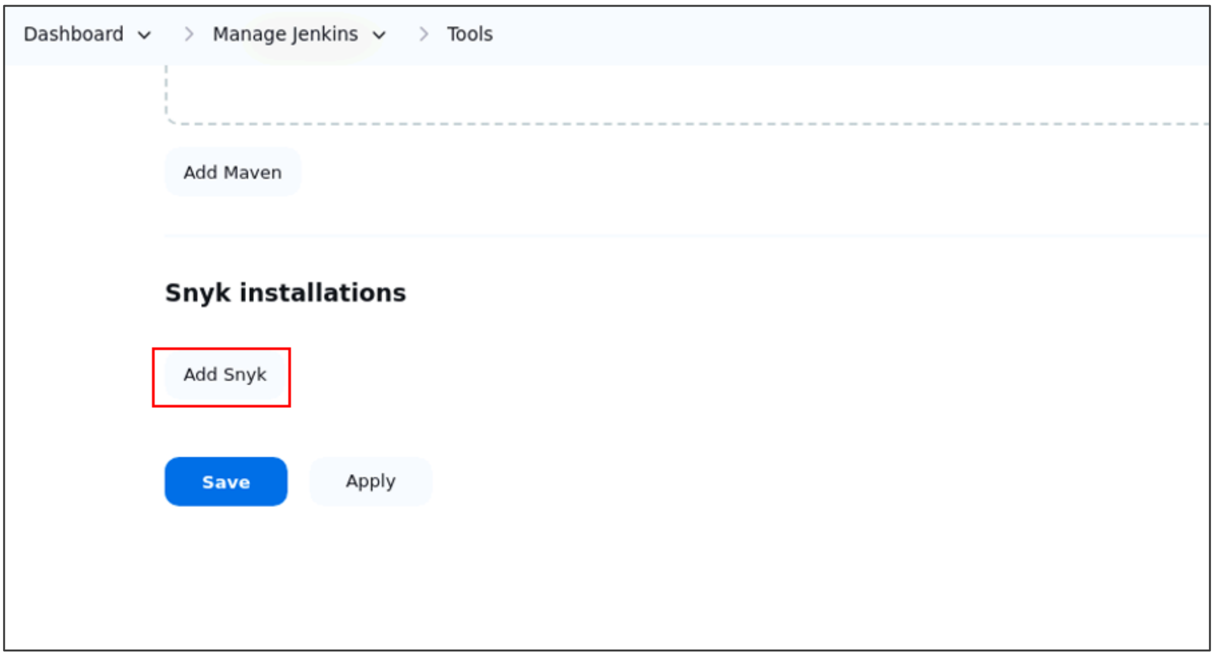
Add Maven

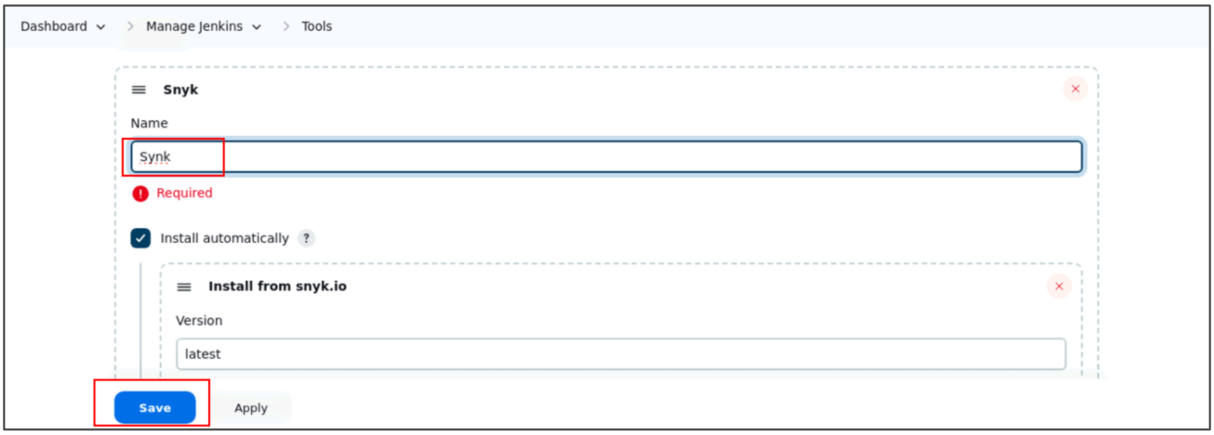
Snyk installations

Add Snyk



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





Name

Synk

Required

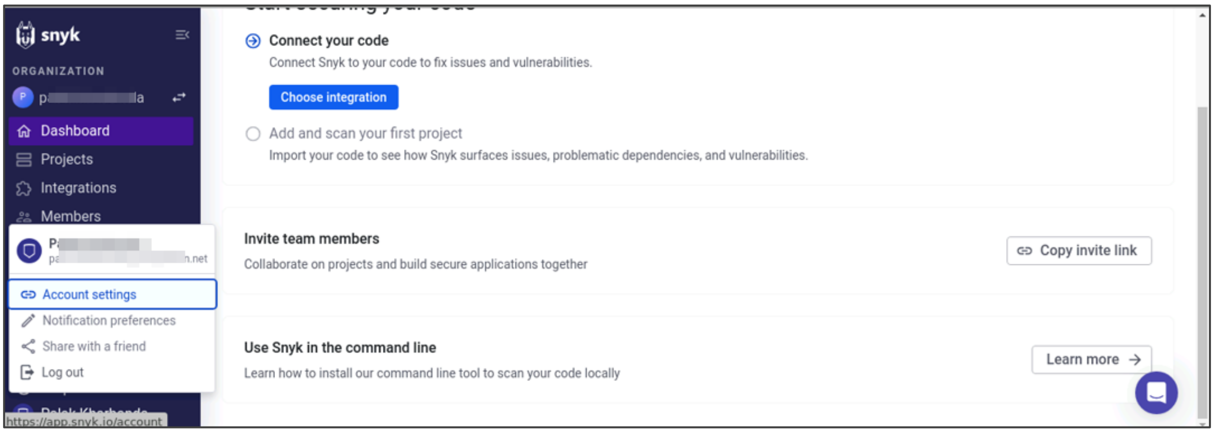
☑ Install automatically ?

Version

latest

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



Connect your code

Connect Snyk to your code to fix issues and vulnerabilities.

○ Add and scan your first project

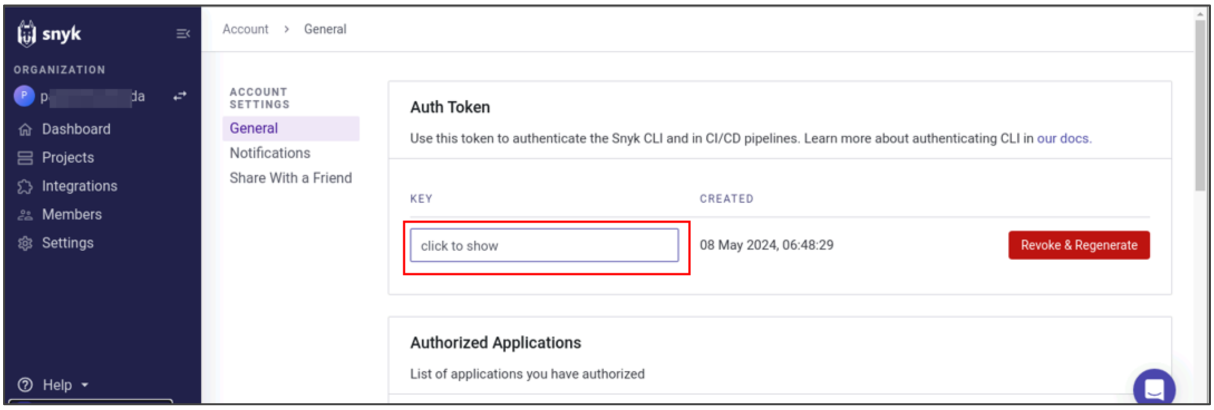
Import your code to see how Snyk surfaces issues, problematic dependencies, and vulnerabilities.

Invite team members

Collaborate on projects and build secure applications together

Use Snyk in the command line

Learn how to install our command line tool to scan your code locally



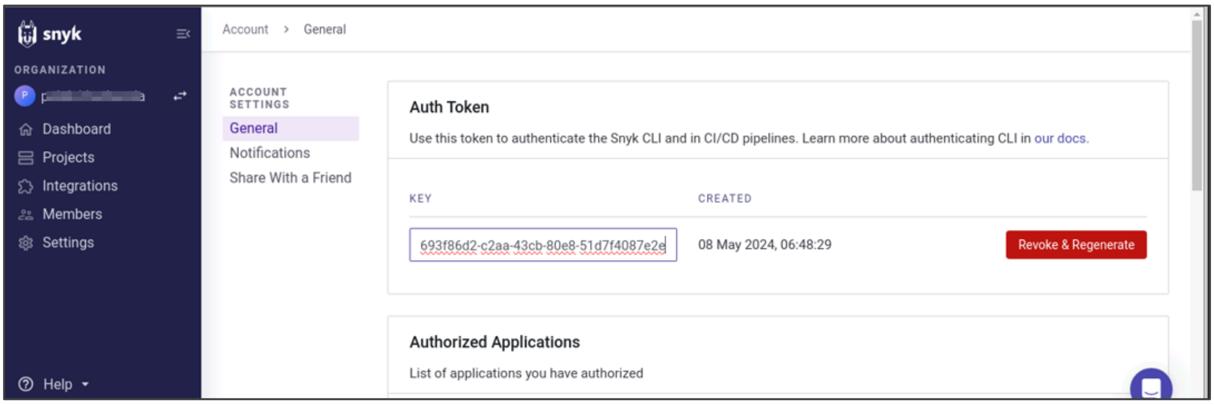
Auth Token

Use this token to authenticate the Snyk CLI and in CI/CD pipelines. Learn more about authenticating CLI in our docs.

KEY	CREATED
click to show	08 May 2024, 06:48:29

Authorized Applications

List of applications you have authorized



Auth Token

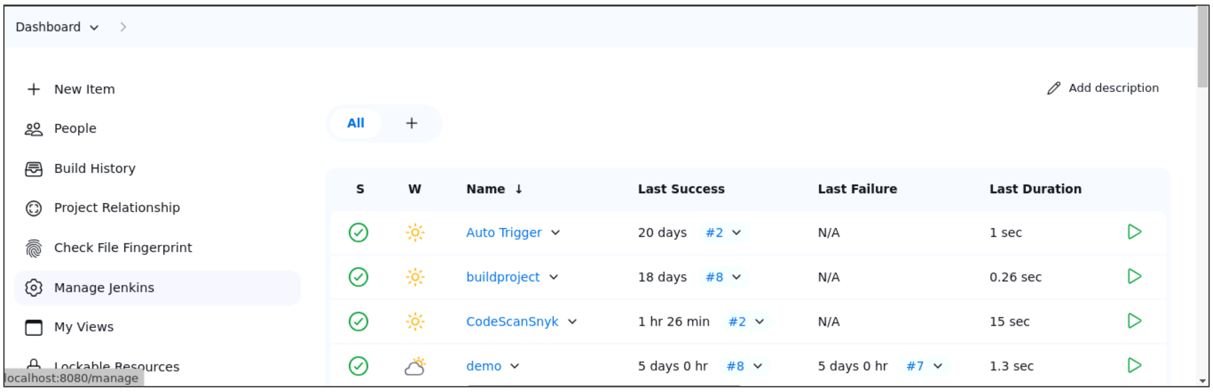
Use this token to authenticate the Snyk CLI and in CI/CD pipelines. Learn more about authenticating CLI in our docs.

KEY	CREATED
693f86d2-c2aa-43cb-80e8-51d7f4087e2e	08 May 2024, 06:48:29

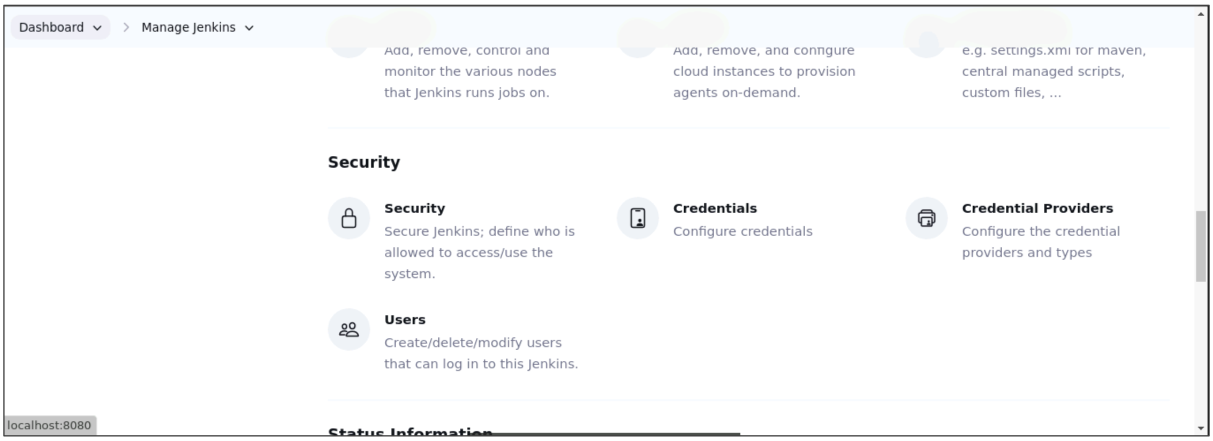
Authorized Applications

List of applications you have authorized

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



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.

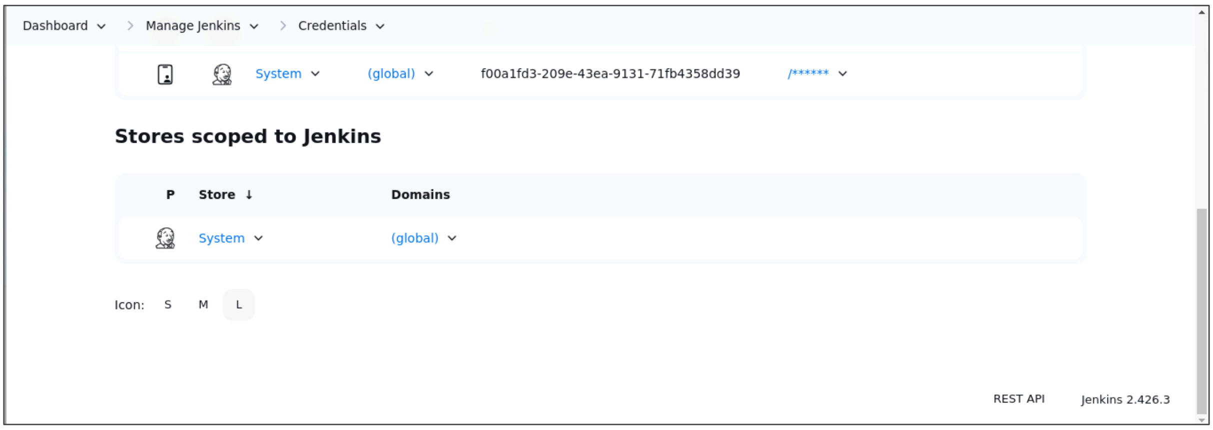
Credentials
Configure credentials

Credential Providers
Configure the credential providers and types

Users
Create/delete/modify users that can log in to this Jenkins.

Status Information

localhost:8080



Dashboard > Manage Jenkins > Credentials

		Store	Domain	ID	Name
		System	(global)	f00a1fd3-209e-43ea-9131-71fb4358dd39	/******

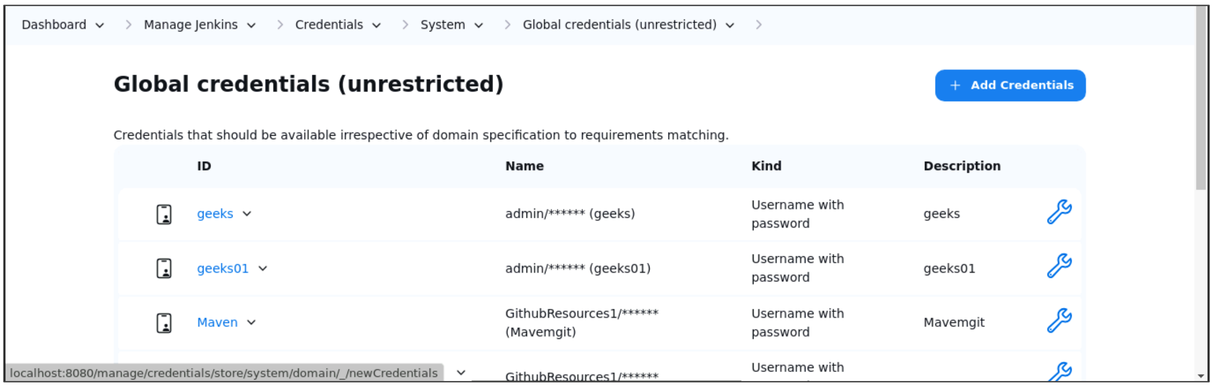
Stores scoped to Jenkins

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) >

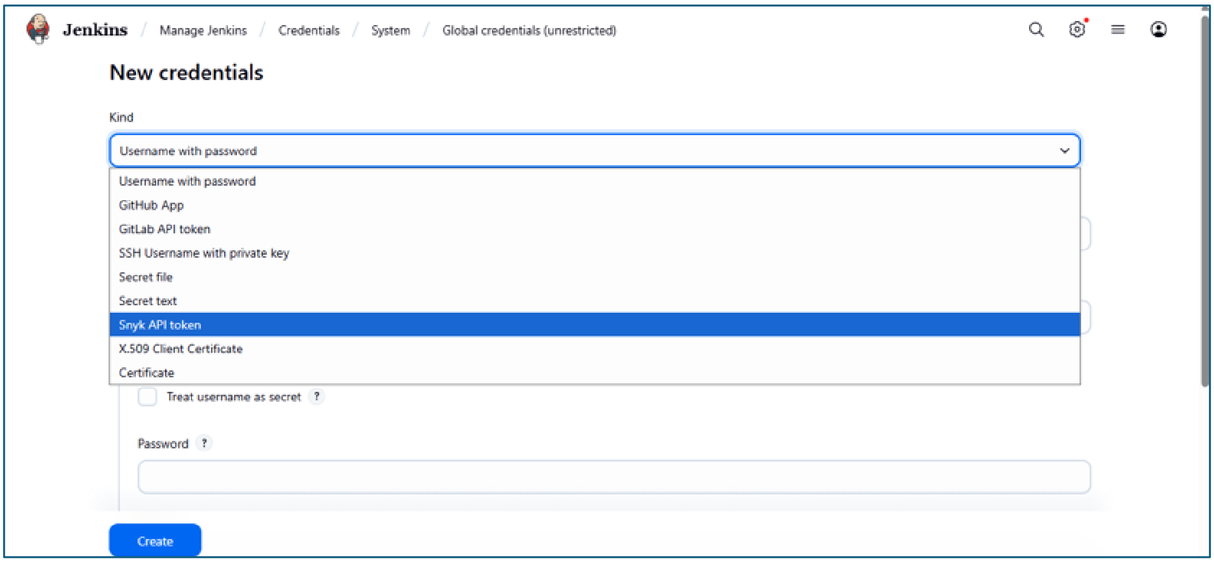
Global credentials (unrestricted)

+ 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	
		GithubResources1/******	Username with		

localhost:8080/manage/credentials/store/system/domain/_/newCredentials

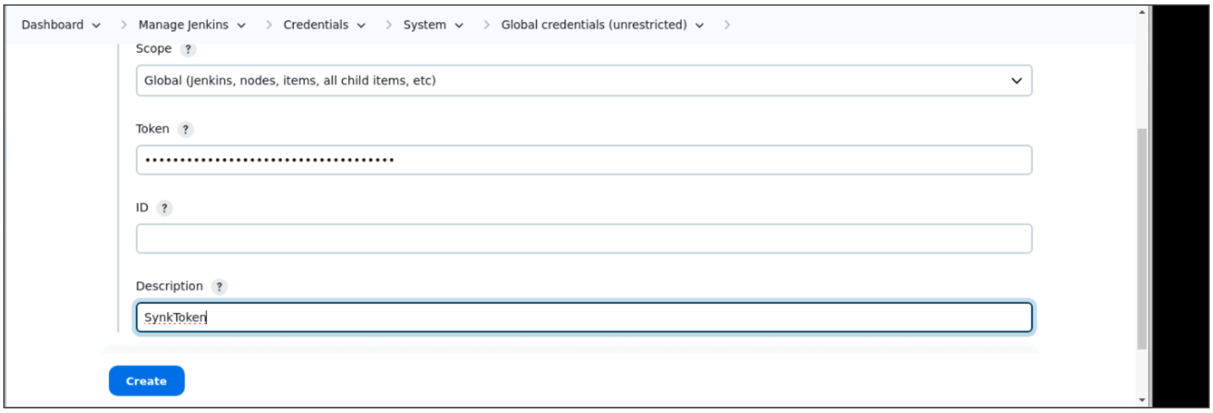


New credentials

Kind: Username with password

☐ Treat username as secret ?

Password ?



Dashboard › Manage Jenkins › Credentials › System › Global credentials (unrestricted) ›
Create

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

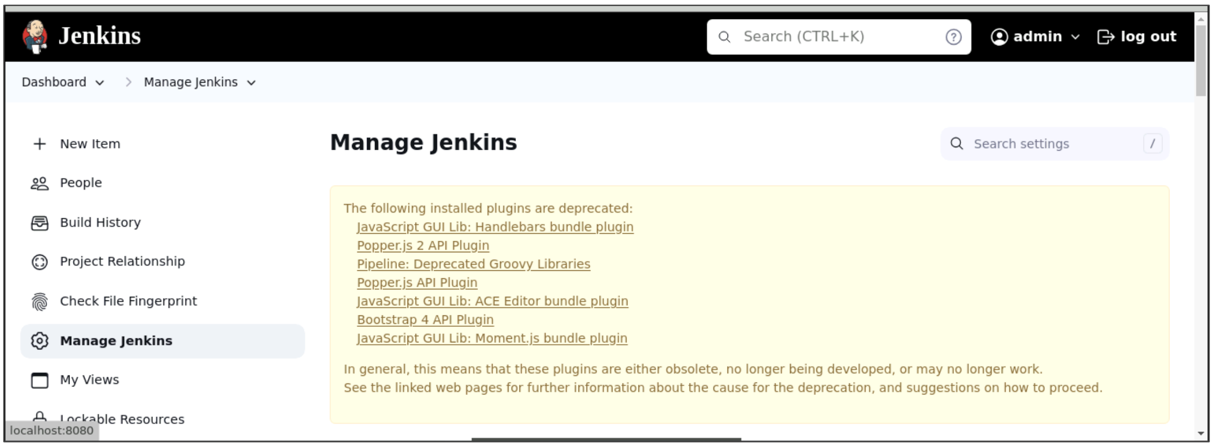
Token ?: ••••••••••••••••••••••••••••••••••••

ID ?:

Description ?: SynkToken

Step 4: Configure the Jenkins job for scanning

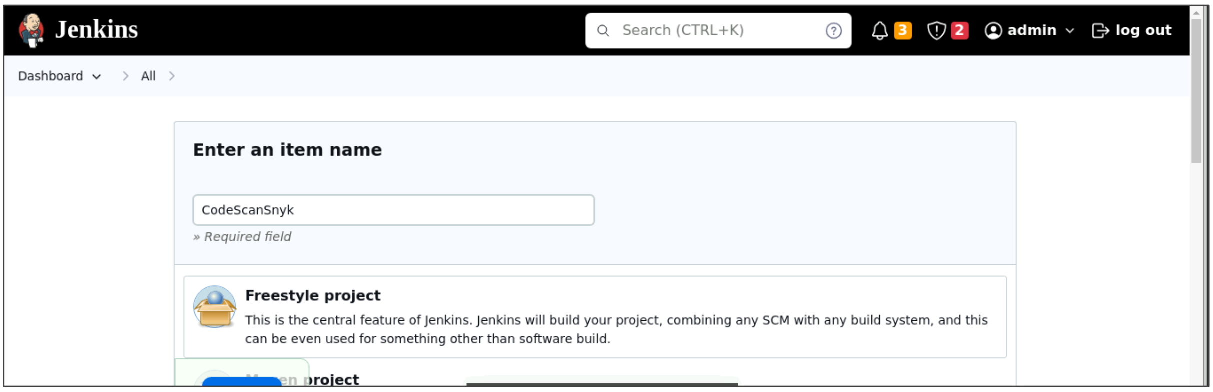
4.1 To create a new Jenkins job, click on **New Item**, enter the item name as **CodeScanSnyk**, select **Freestyle project**, and then click **OK**



Manage Jenkins

The following installed plugins are deprecated:
- JavaScript GUI Lib: Handlebars bundle plugin
- Popper.js 2 API Plugin
- Pipeline: Deprecated Groovy Libraries
- Popper.js API Plugin
- JavaScript GUI Lib: ACE Editor bundle plugin
- Bootstrap 4 API Plugin
- JavaScript GUI Lib: Moment.js bundle plugin

In general, this means that these plugins are either obsolete, no longer being developed, or may no longer work.
See the linked web pages for further information about the cause for the deprecation, and suggestions on how to proceed.



Enter an item name

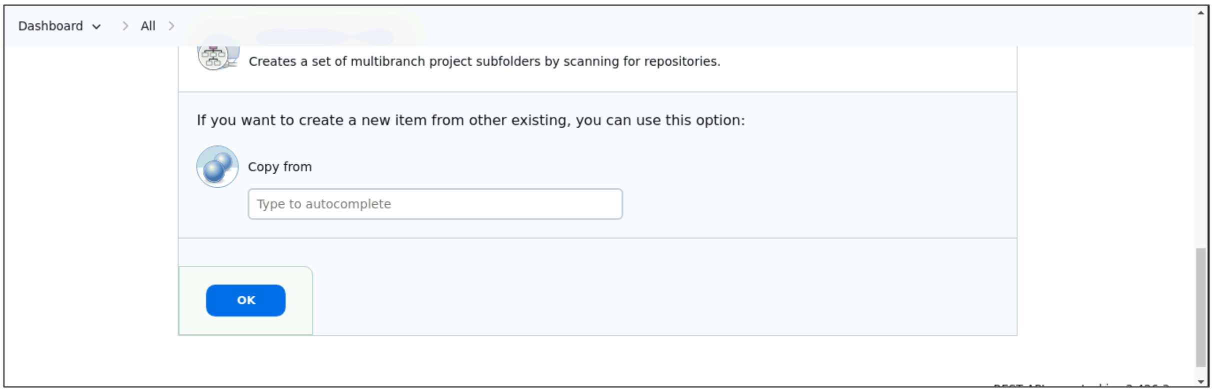
CodeScanSnyk

» Required field

Freestyle project

This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Maven project



Creates a set of multibranch project subfolders by scanning for repositories.

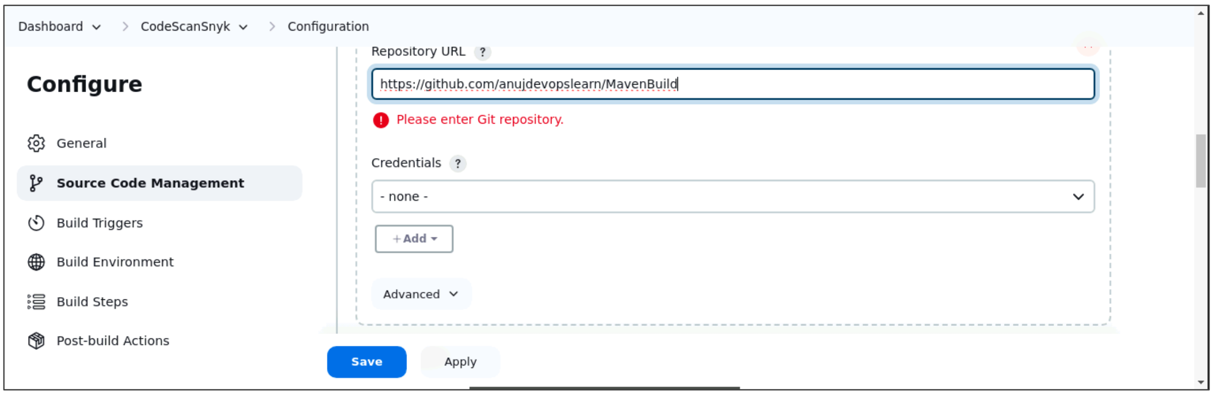
If you want to create a new item from other existing, you can use this option:

Copy from

Type to autocomplete

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**

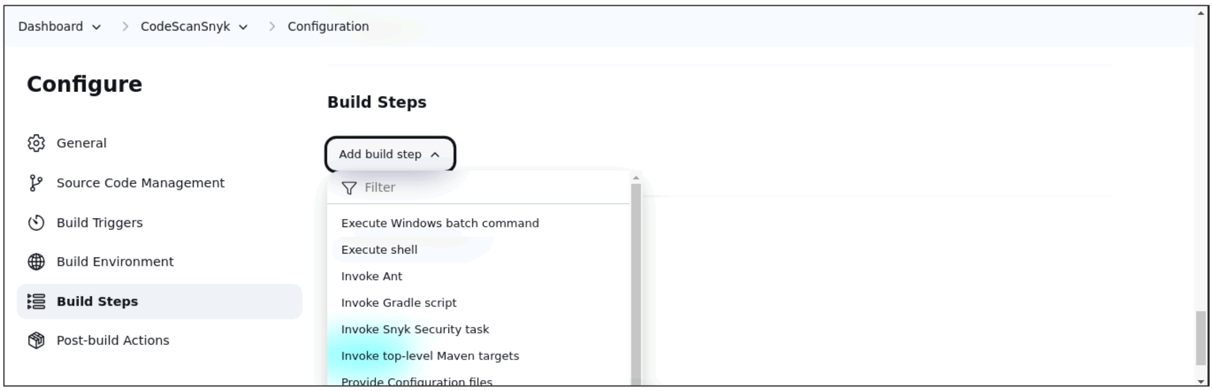


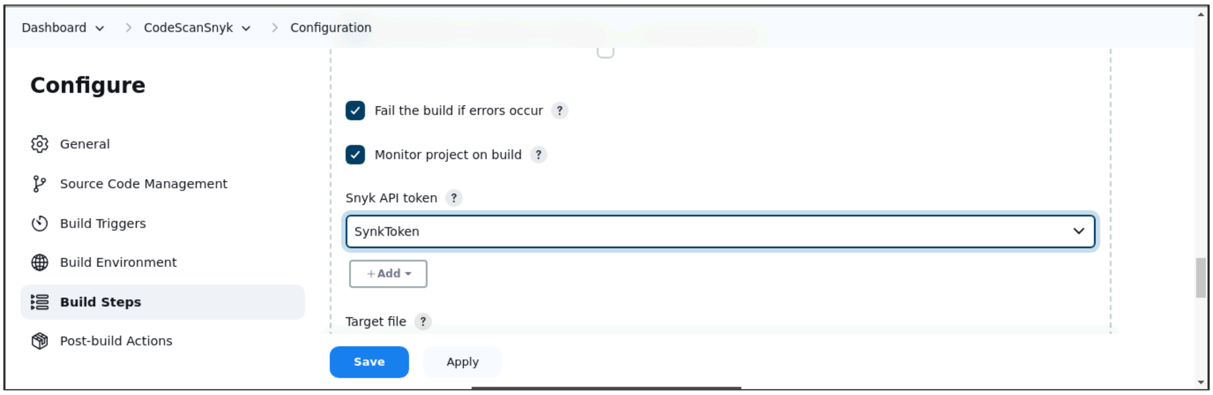
Repository URL

https://github.com/anujdevopslearn/MavenBuild

Please enter Git repository.

Credentials: - none -





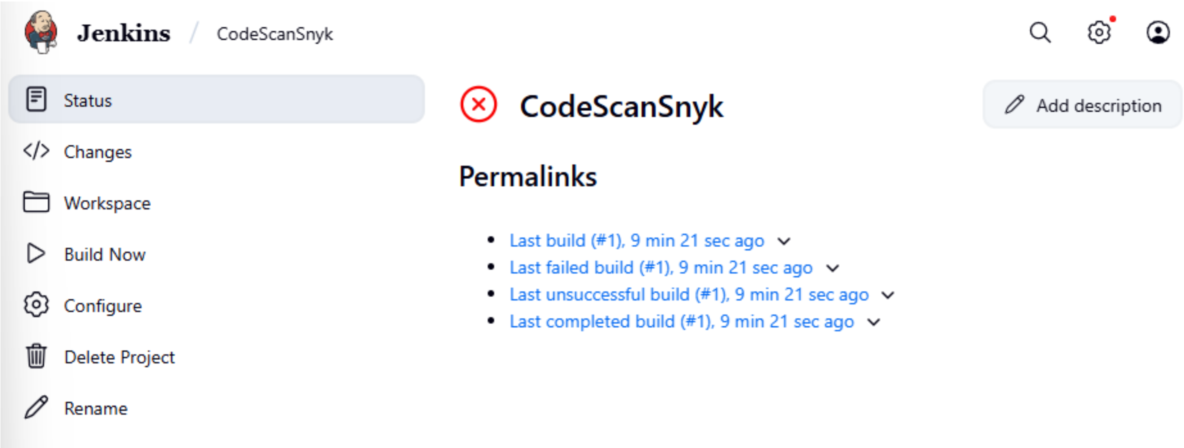
☑ Fail the build if errors occur

☑ Monitor project on build

Snyk API token: SynkToken

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

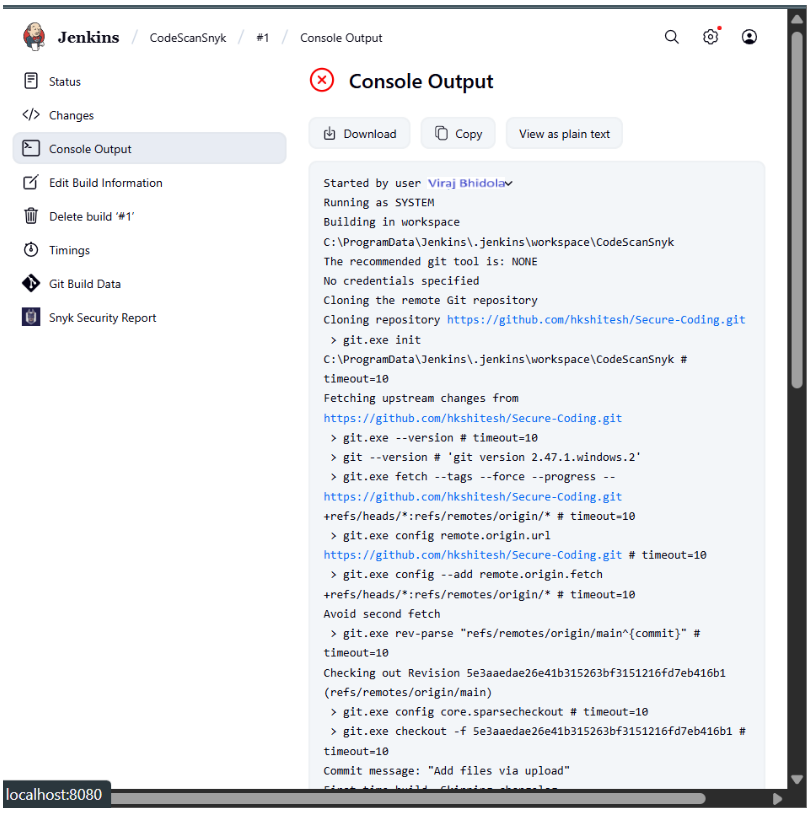
4.3 To check the build status, click on the build link under **Permalinks.** After that, click on **Console Output**



CodeScanSnyk

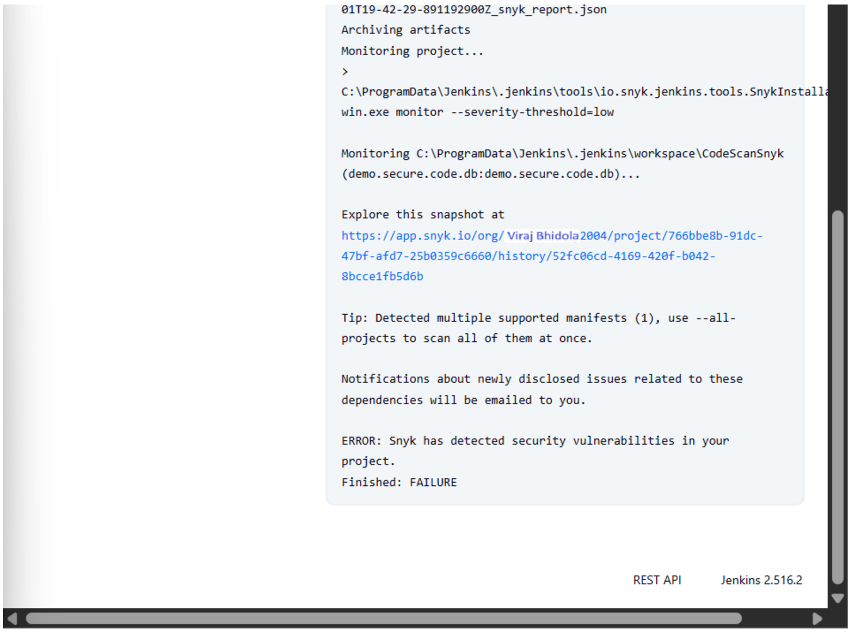
Permalinks

- Last build (#1), 9 min 21 sec ago
- Last failed build (#1), 9 min 21 sec ago
- Last unsuccessful build (#1), 9 min 21 sec ago
- Last completed build (#1), 9 min 21 sec ago



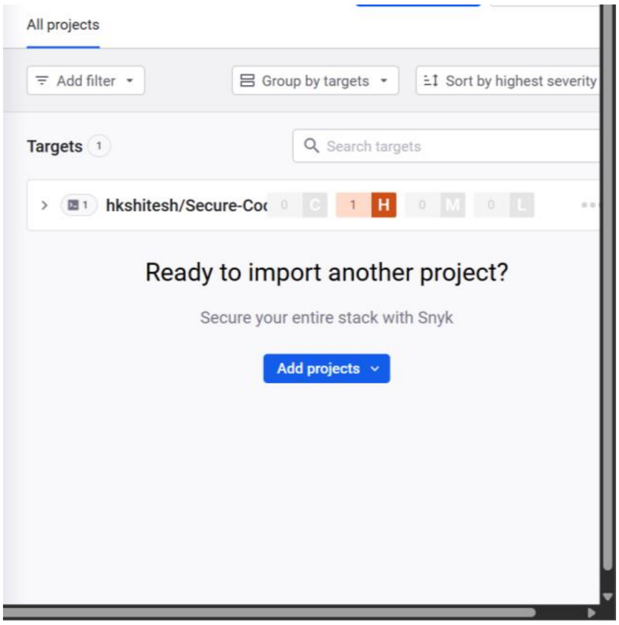
Console Output

```
Started by user Viraj Bhidola
Running as SYSTEM
Building in workspace
C:\ProgramData\Jenkins\.jenkins\workspace\CodeScanSnyk
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/hkshitesh/Secure-Coding.git
 > git.exe init
C:\ProgramData\Jenkins\.jenkins\workspace\CodeScanSnyk #
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 config remote.origin.url
https://github.com/hkshitesh/Secure-Coding.git # timeout=10
 > git.exe config --add remote.origin.fetch
+refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
 > 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"
```



```
01T19-42-29-891192900Z_snyk_report.json
Archiving artifacts
Monitoring project...
>
C:\ProgramData\Jenkins\.jenkins\tools\io.snyk.jenkins.tools.SnykInstalla
win.exe monitor --severity-threshold=low

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

Explore this snapshot at
https://app.snyk.io/org/ Viraj Bhidola2004/project/766bbe8b-91dc-
47bf-afd7-25b0359c6660/history/52fc06cd-4169-420f-b042-
8bcce1fb5d6b

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.

ERROR: Snyk has detected security vulnerabilities in your
project.
Finished: FAILURE
```

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.