

Lab Exercise 19

Setting up Snyk for SAST in Jenkins

Name: Ayush Bhardwaj

Sap id: 500124917

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

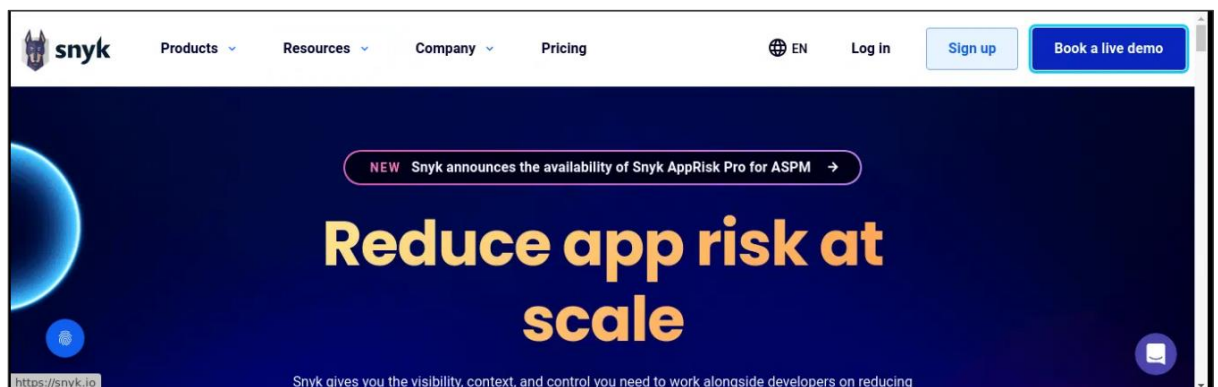
Prerequisites: None

Steps to be followed:

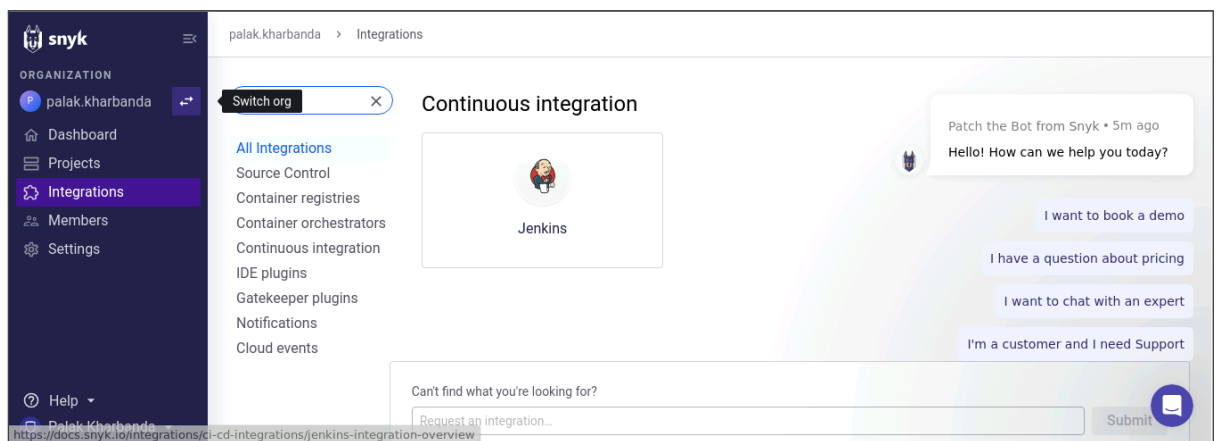
1. Configure Snyk as a SAST scan tool
2. Create and configure a Jenkins job for Snyk integration
3. Manage Snyk API and Jenkins credentials
4. Configure the Jenkins job for scanning

Step 1: Configure Snyk as a SAST scan tool

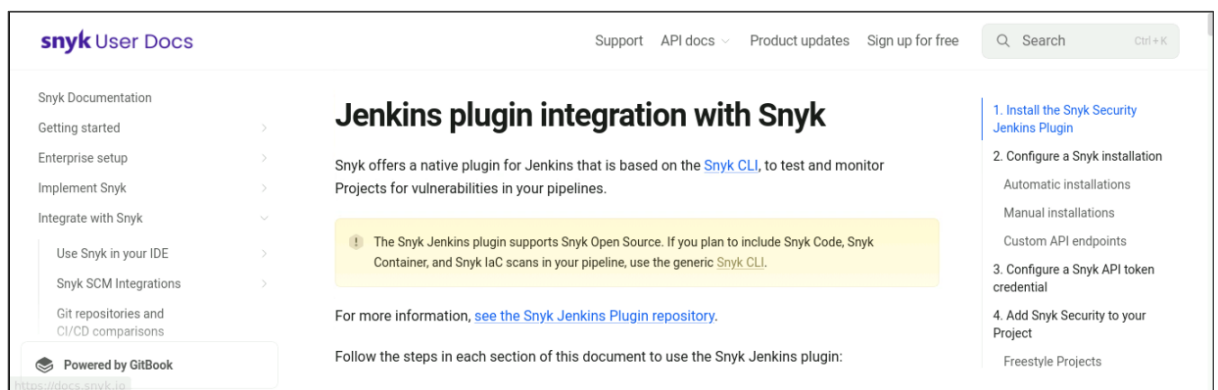
1.1 Visit <https://snyk.io/>, sign up for a new Snyk account, and log in



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:



Sign in to Jenkins

Username

Password

☐ Keep me signed in

[Sign 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**

Dashboard ▾ >

+ New Item

People

Build History

Project Relationship

Check File Fingerprint

Manage Jenkins

My Views

Lockable Resources

localhost:8080/manage

All +

Add description

| 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 ▶ |

Jenkins

Search (CTRL+K)

admin ▾ log out

Dashboard ▾ > Manage Jenkins ▾ > Plugins

Plugins

Updates 19

Available plugins

Installed plugins

Advanced settings

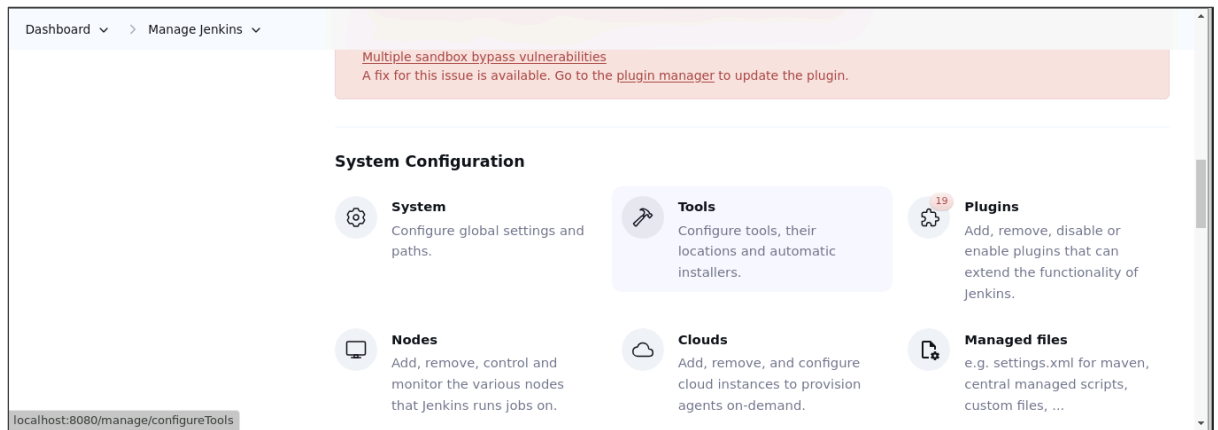
Q snyk

Install

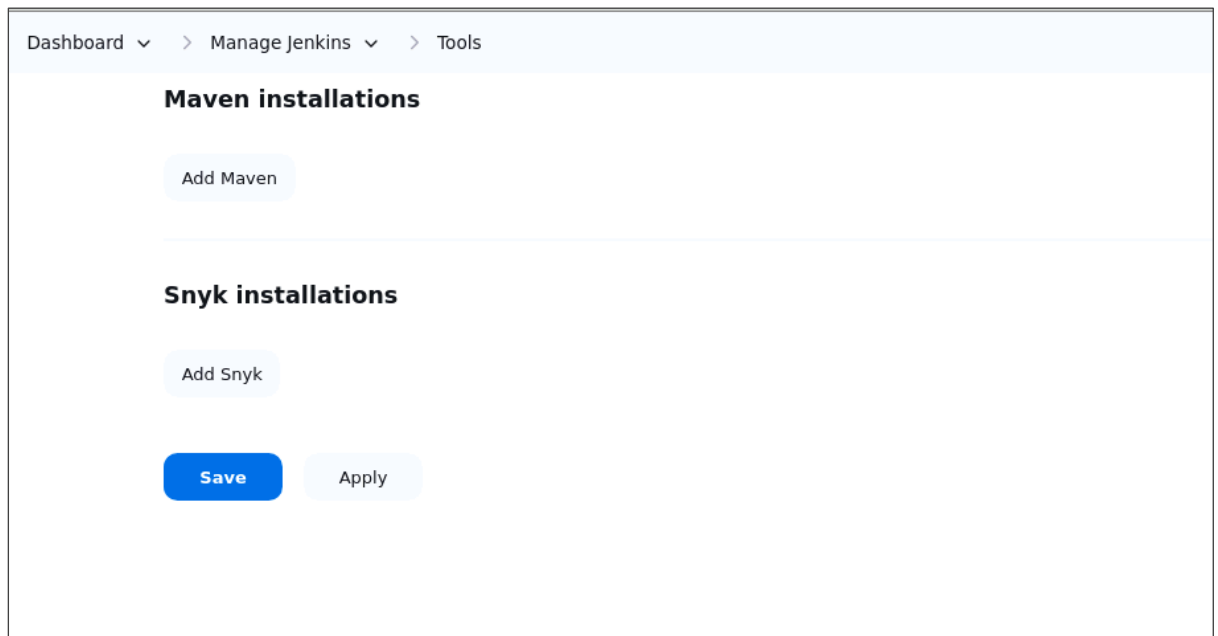
| Install | Name ↓ | Released |
|--------------------------|---|------------------|
| <input type="checkbox"/> | Snyk Security 4.0.2 DevSecOps Add the ability to test your code dependencies for vulnerabilities against Snyk database | 7 mo 29 days ago |

REST API Jenkins 2.426.3

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



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



Dashboard > Manage Jenkins > Tools

Maven

Name

Required

☒ Install automatically ?

Install from Apache

Version

Save Apply

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

Dashboard > Manage Jenkins > Tools

Add Maven

Snyk installations

Add Snyk

Save Apply

Dashboard > Manage Jenkins > Tools

Snyk

Name
Synk

Required

☒ Install automatically ?

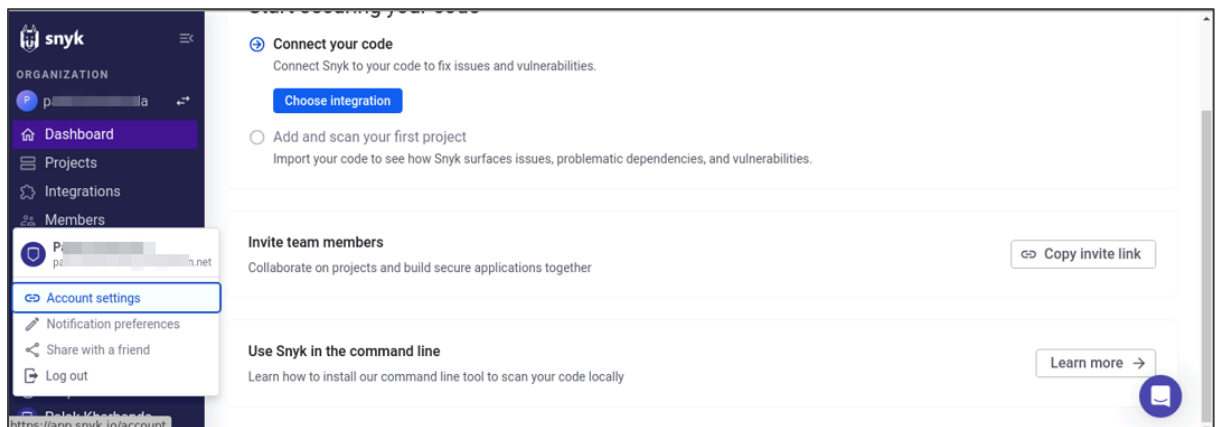
Install from snyk.io

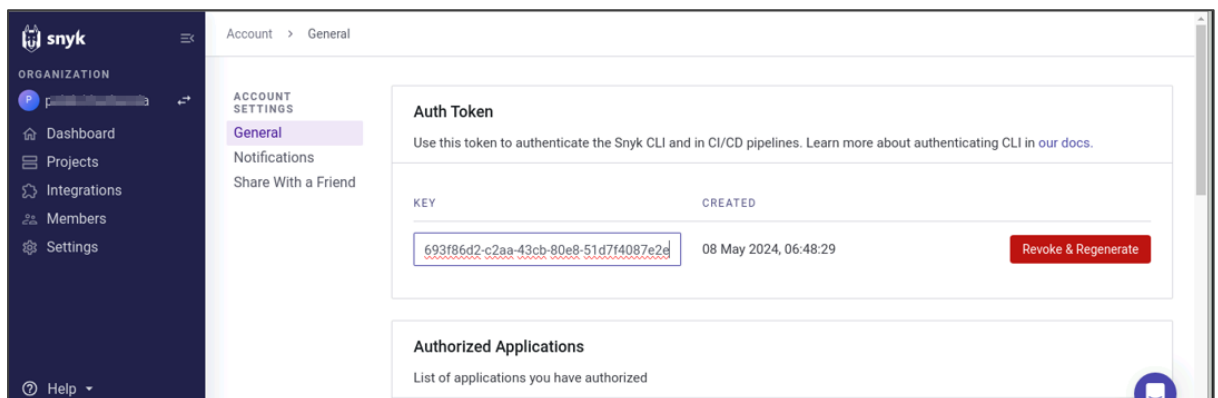
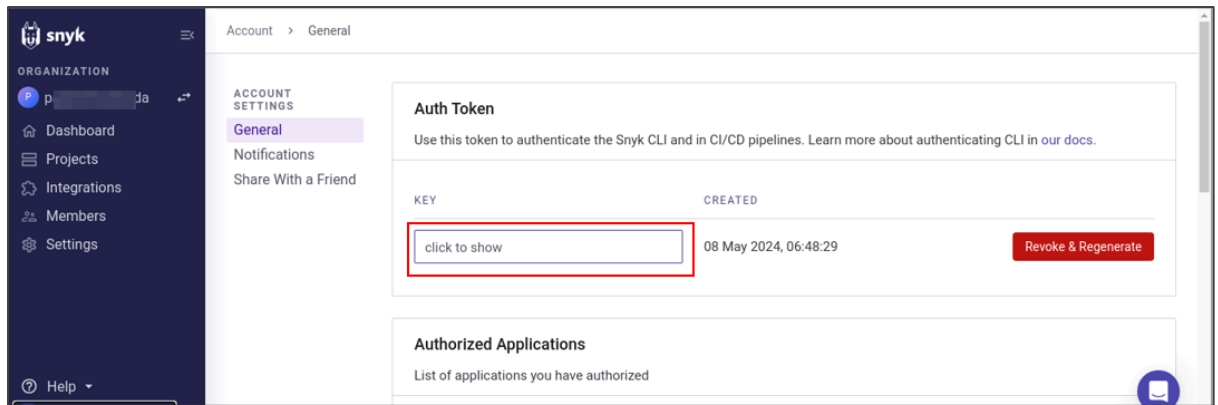
Version
latest

Save Apply

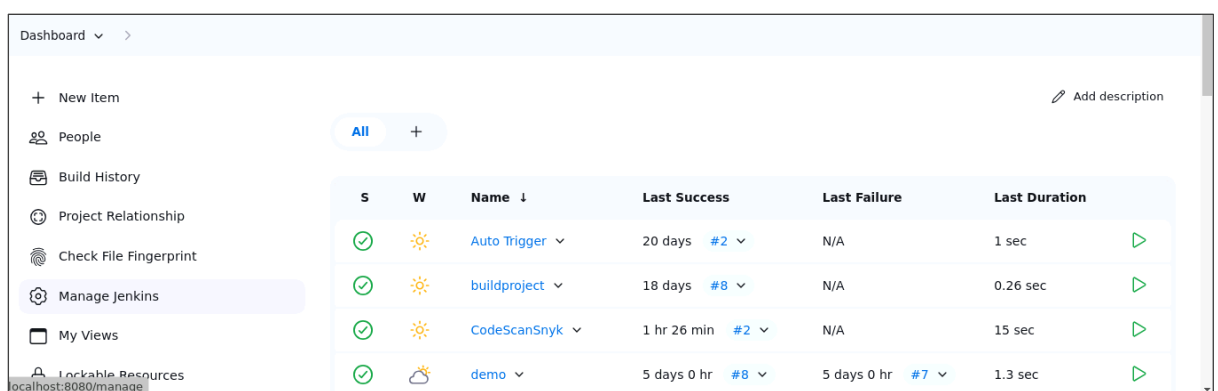
Step 3: Manage Snyc API and Jenkins credentials

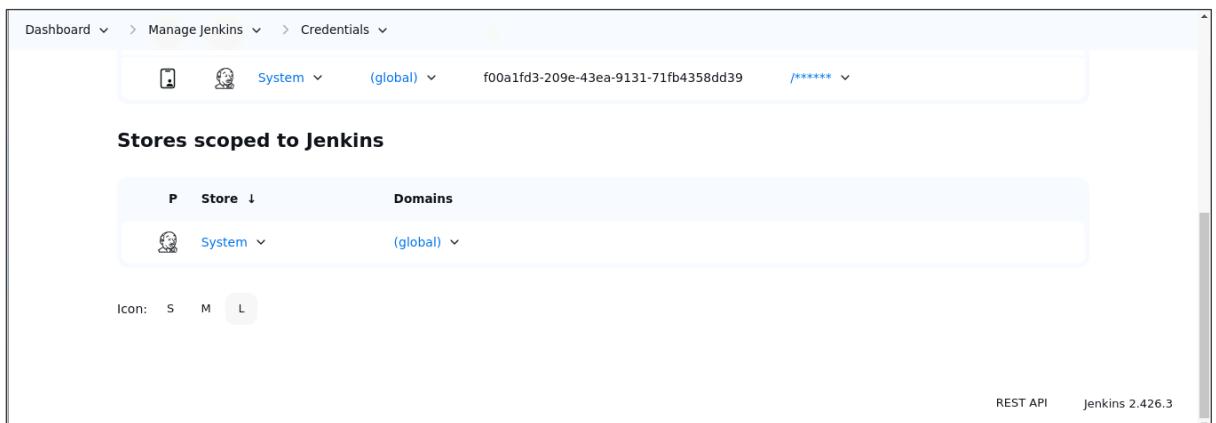
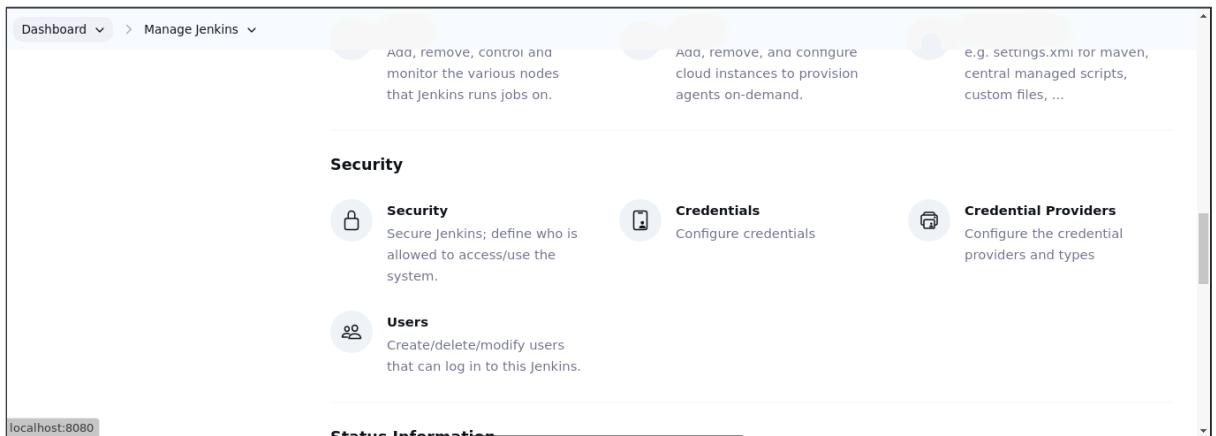
3.1 To retrieve your Snyc API token, go to **Account Settings** in your Snyc account, click on **Click to show** under the Auth Token key field, and copy the token for further reference



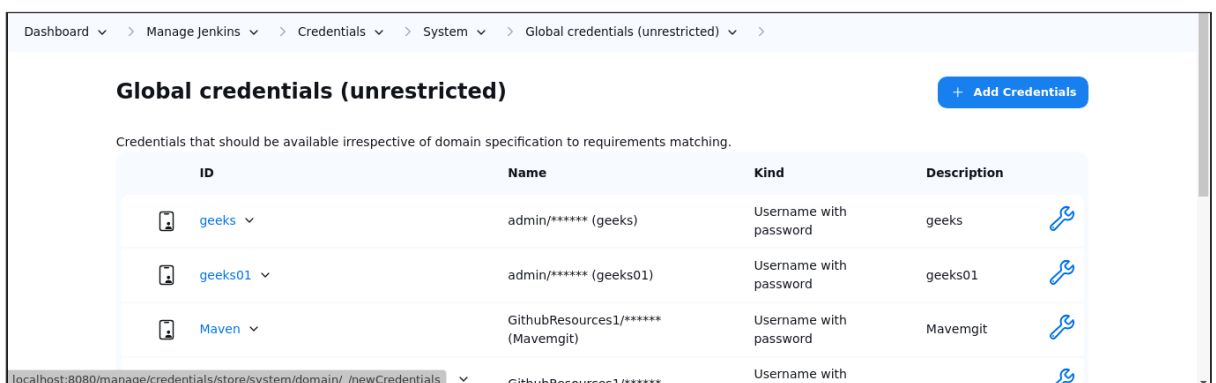


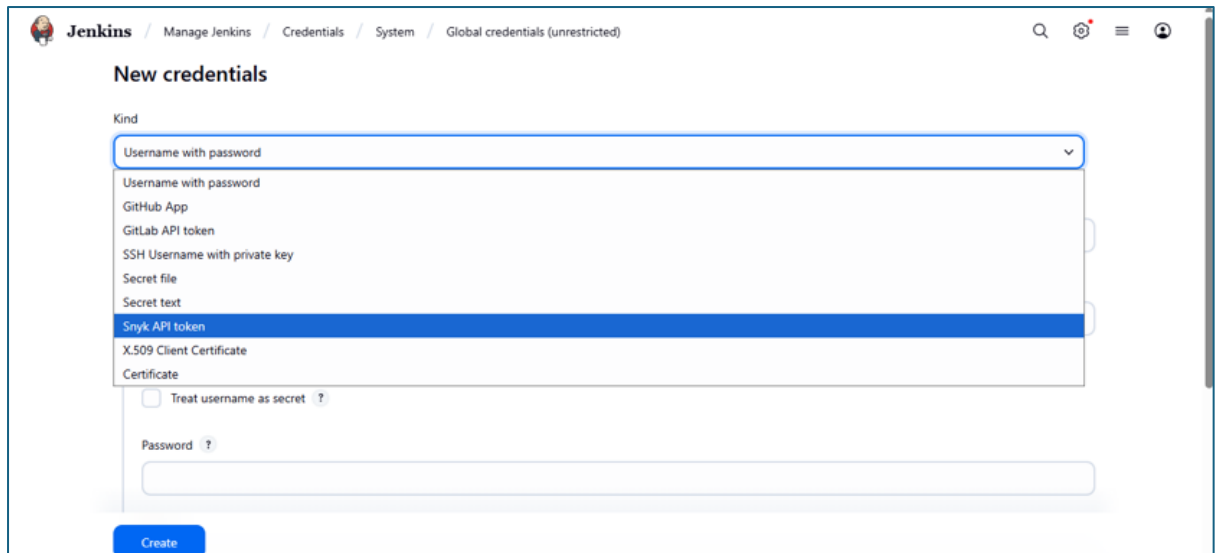
3.2 In the Jenkins interface, go to **Manage Jenkins**, select **Security**, then choose **Credentials** and select **global** to add global credentials



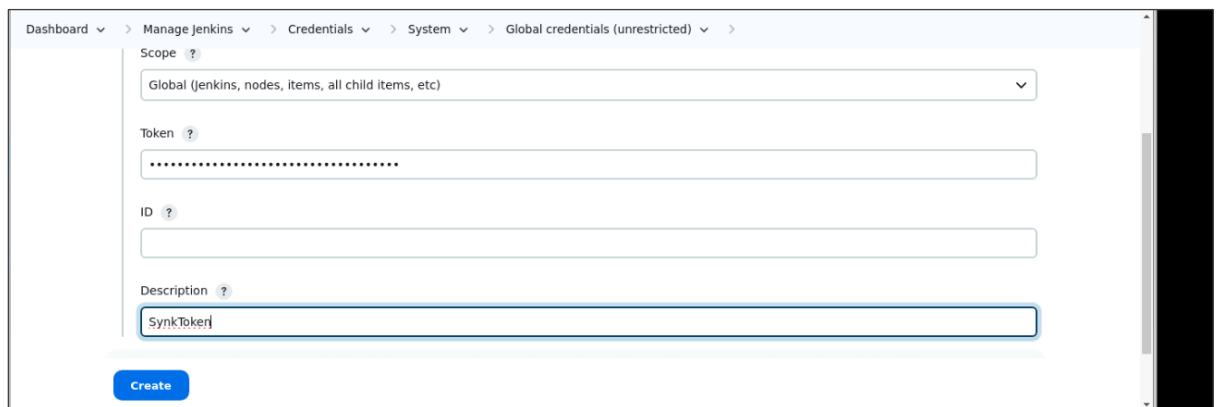


3.3 Click on **Add Credentials**, select the **Snyk API token** from the **Kind** field, paste the copied token from step 3.1 into the **Token** field, and then click the **Create** button





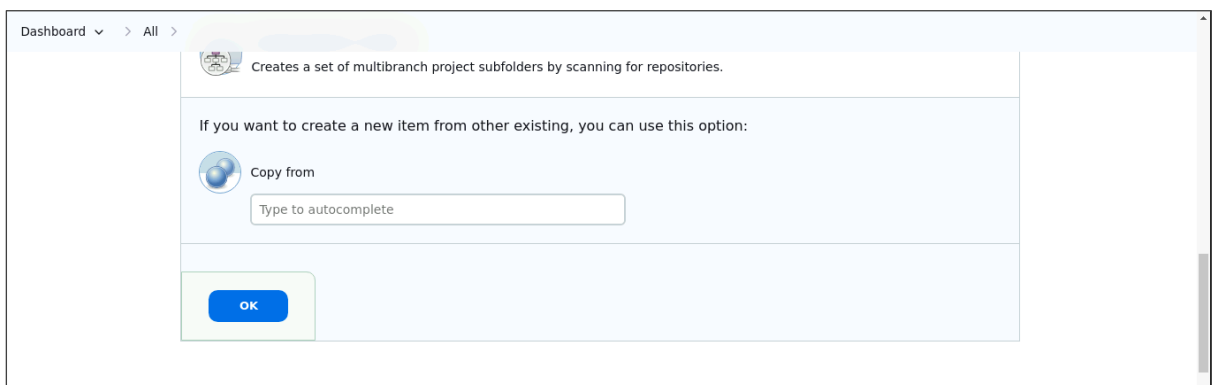
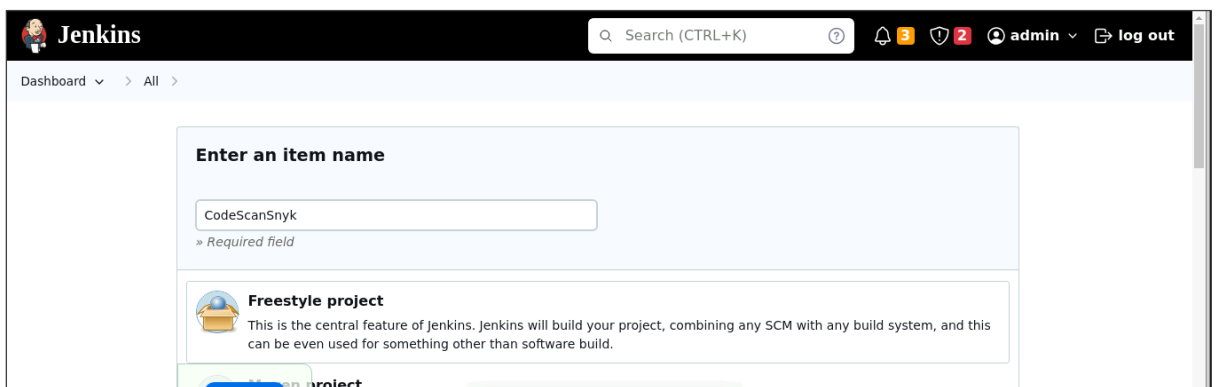
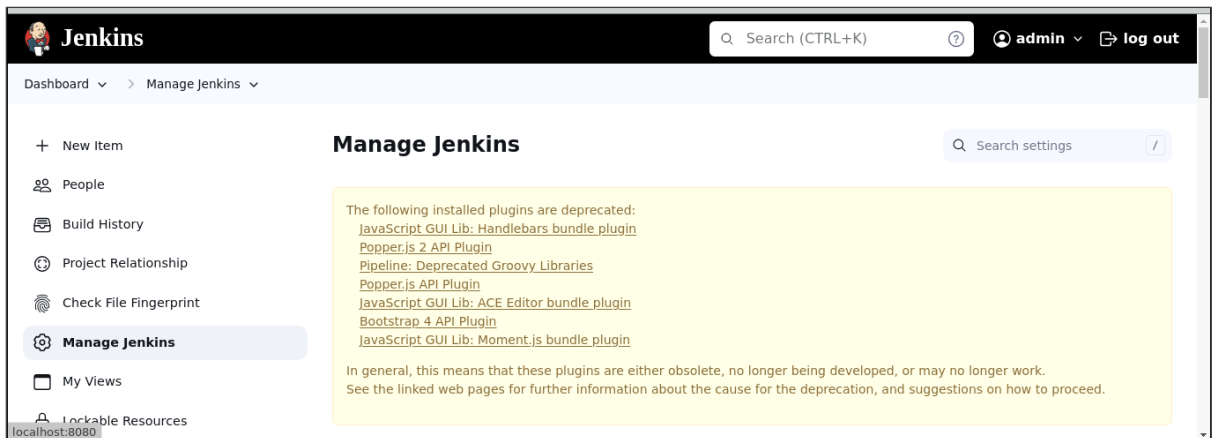
The image shows the 'New credentials' form in the Jenkins web interface. The breadcrumb trail at the top reads: 'Jenkins / Manage Jenkins / Credentials / System / Global credentials (unrestricted)'. The form title is 'New credentials'. Under the 'Kind' dropdown, a list of credential types is shown, with 'Snyk API token' selected and highlighted in blue. Other options include 'Username with password', 'GitHub App', 'GitLab API token', 'SSH Username with private key', 'Secret file', 'Secret text', 'X.509 Client Certificate', and 'Certificate'. Below the list, there is a checkbox labeled 'Treat username as secret' and a 'Password' input field. A blue 'Create' button is at the bottom.



The image shows the configuration form for a new credential in Jenkins. The breadcrumb trail is: 'Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted)'. The form fields are: 'Scope' (dropdown menu set to 'Global (jenkins, nodes, items, all child items, etc)'), 'Token' (password field filled with dots), 'ID' (text field), and 'Description' (text field containing 'SynkToken'). A blue 'Create' button is at the bottom.

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 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>

Dashboard > CodeScanSnyk > Configuration

Configure

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

Repository URL ?

`https://github.com/anujdevopslearn/MavenBuild`

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 Snyk 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 ?

SynkToken


+ Add




Target file ?


Save Apply


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**


 **Jenkins** / CodeScanSnyk





 Status


 Changes


 Workspace


 Build Now

 Configure

 Delete Project


 Rename




 **CodeScanSnyk**


 Add description


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](#) ▾


 **Jenkins** / CodeScanSnyk / #1 / Console Output





 Status


 Changes


 Console Output


 Edit Build Information


 Delete build '#1'


 Timings

 Git Build Data

 Snyk Security Report

 **Console Output**

 Download

 Copy

[View as plain text](#)

```
Started by user Ayush Bhardwaj
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"
Fetch first build -> Cloning changes
```

localhost:8080

```
01T19-42-29-891192900Z_snyk_report.json
Archiving artifacts
Monitoring project...
>
C:\ProgramData\Jenkins\.jenkins\tools\io.snyk.jenkins.tools.SnykInstall
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/AyushBhardwaj2004/project/766bbe8b-91dc-47bf-afd7-25b0359c6660/history/52fc06cd-4169-420f-b042-8bcce1fb5d6b

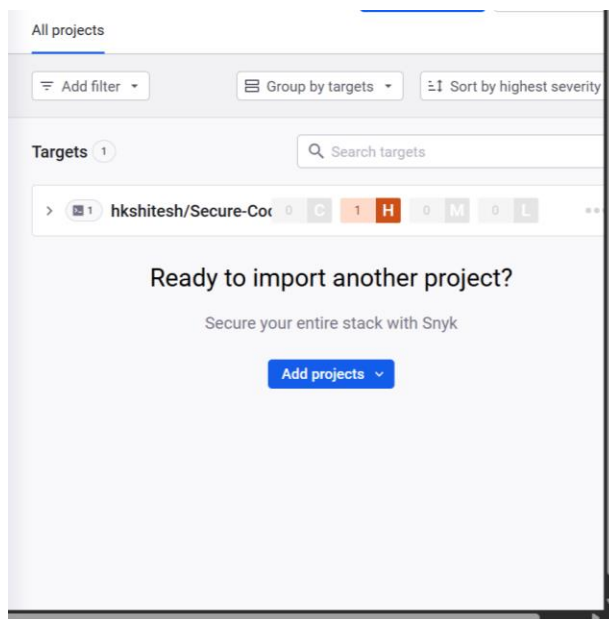
Tip: Detected multiple supported manifests (1), use --all-projects to scan all of them at once.

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
```

REST API Jenkins 2.516.2

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.

