

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

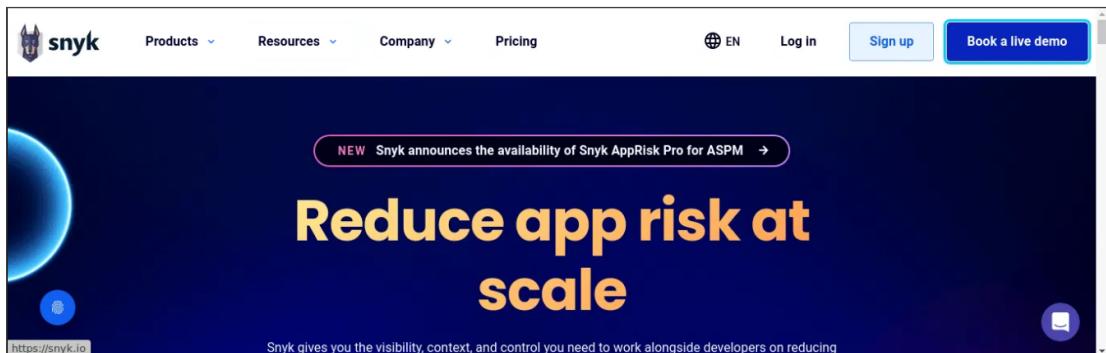
Prerequisites: None

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



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

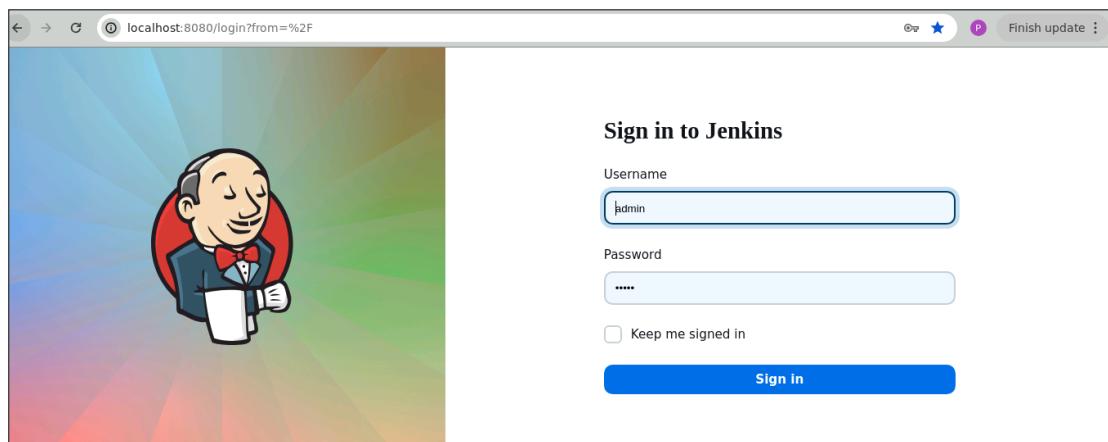
The screenshot shows the Snyk integration interface. On the left, there's a sidebar with 'thepraktik2468' organization details, including a profile picture, 'Dashboard', 'Projects', 'Integrations' (which is highlighted in purple), 'Members', and 'Settings'. Below this are 'Product updates', 'Help', and 'Pulkit'. The main content area has a header 'Integrations' and a sub-header 'Source Control'. It features a search bar 'Search integrations...'. Under 'Source Control', there are cards for GitHub, GitHub Enterprise (with a 'Upgrade plan to enable' button), GitHub Cloud App (marked as 'NEW'), and GitHub Server App (also with an 'Upgrade plan to enable' button). Below these are cards for GitLab, Bitbucket Server (with an 'Upgrade plan to enable' button), Bitbucket Cloud App, and Bitbucket Cloud. A single card for 'Azure Repos' is shown below the GitHub row. Under 'Container registries', there are cards for Docker, Google Container Registry, AWS Lambda, and Azure Container Registry.

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

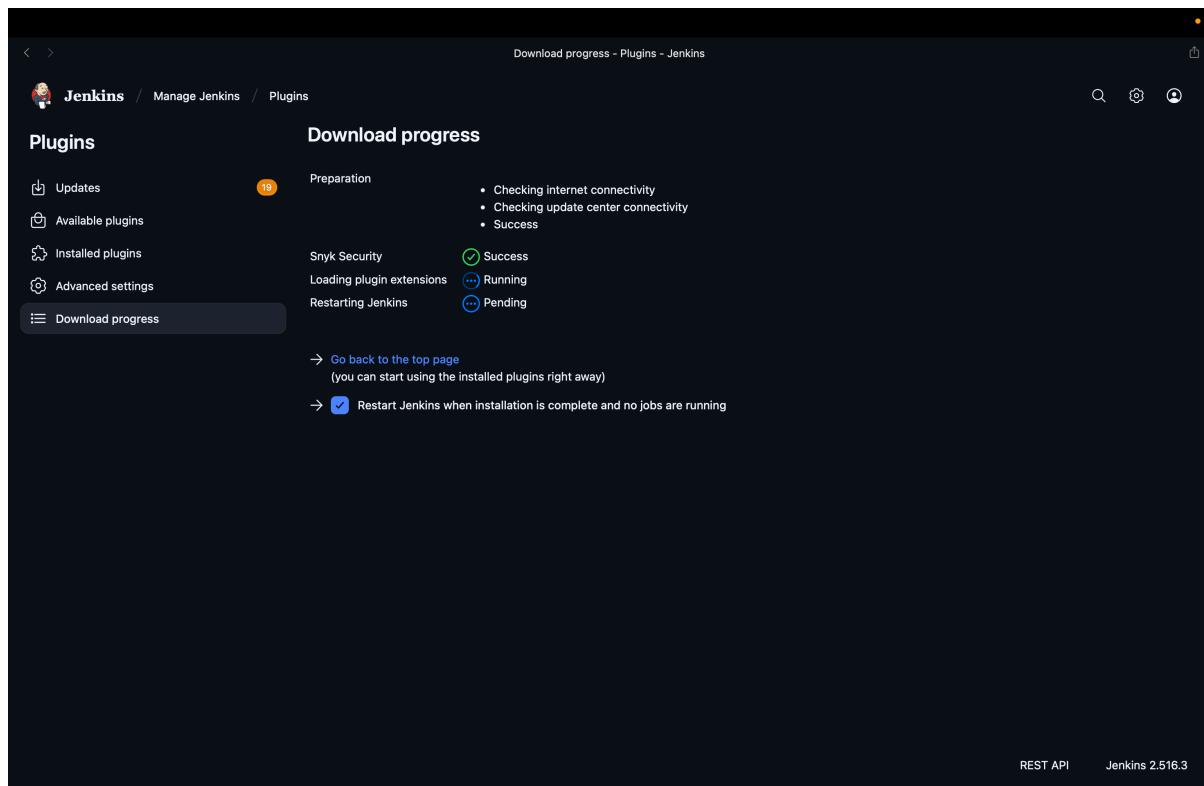
The screenshot shows the 'Jenkins plugin integration with Snyk' documentation page. The top navigation bar includes 'Support', 'Snyk Learn', 'API reference', 'Product updates', 'Sign up for free', and a search bar. The left sidebar has sections for 'DEVELOPER TOOLS' (CircleCI, Jenkins, Maven, TeamCity, Terraform) and 'SCAN WITH SNYK' (Overview, Start scanning, Import Project repository, Pull Requests, Snyk Open Source, Snyk Code, Snyk Container, Snyk IaC, Snyk Essentials). The main content area has a breadcrumb 'DEVELOPER TOOLS > SNYK CI/CDS' and the title 'Jenkins plugin integration with Snyk'. It explains that the plugin supports Snyk Open Source, Snyk Code, Snyk Container, and Snyk IaC. It provides a link to the 'Snyk Jenkins Plugin repository'. Below this, it lists steps to use the plugin: 1. Install the Snyk Security Jenkins Plugin, 2. Configure a Snyk installation, 3. Configure a Snyk API token credential, 4. Add Snyk Security to your Project, and 5. View your Snyk Security Report. To the right, there's a sidebar with links for installing the plugin, configuring Snyk, adding security to Jenkins projects, and viewing reports. At the bottom, there's a 'Was this helpful?' poll and a cookie consent banner.

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

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



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



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

The screenshot shows the Jenkins Manage Jenkins interface. In the top left, there's a Jenkins logo and the text "Jenkins / Manage Jenkins". On the right, there's a search bar labeled "Search settings" and some global configuration icons. The main area is titled "Manage Jenkins" and contains several sections:

- System Configuration**: Includes links for "System" (Configure global settings and paths), "Nodes" (Add, remove, control and monitor nodes), "Clouds" (Add, remove, and configure cloud instances), and "Tools" (Configure tools, their locations and automatic installers). The "Tools" section is highlighted.
- Security**: Includes links for "Security" (Secure Jenkins; define who is allowed to access/use the system) and "Users" (Create/delete/modify users that can log in to this Jenkins).
- Status Information**: Includes links for "System Information" (Displays various environmental information to assist trouble-shooting), "System Log" (System log captures output from java.util.logging output related to Jenkins), and "Load Statistics" (Check your resource utilization and see if you need more computers for your builds).
- About Jenkins**: A link to the About Jenkins page.

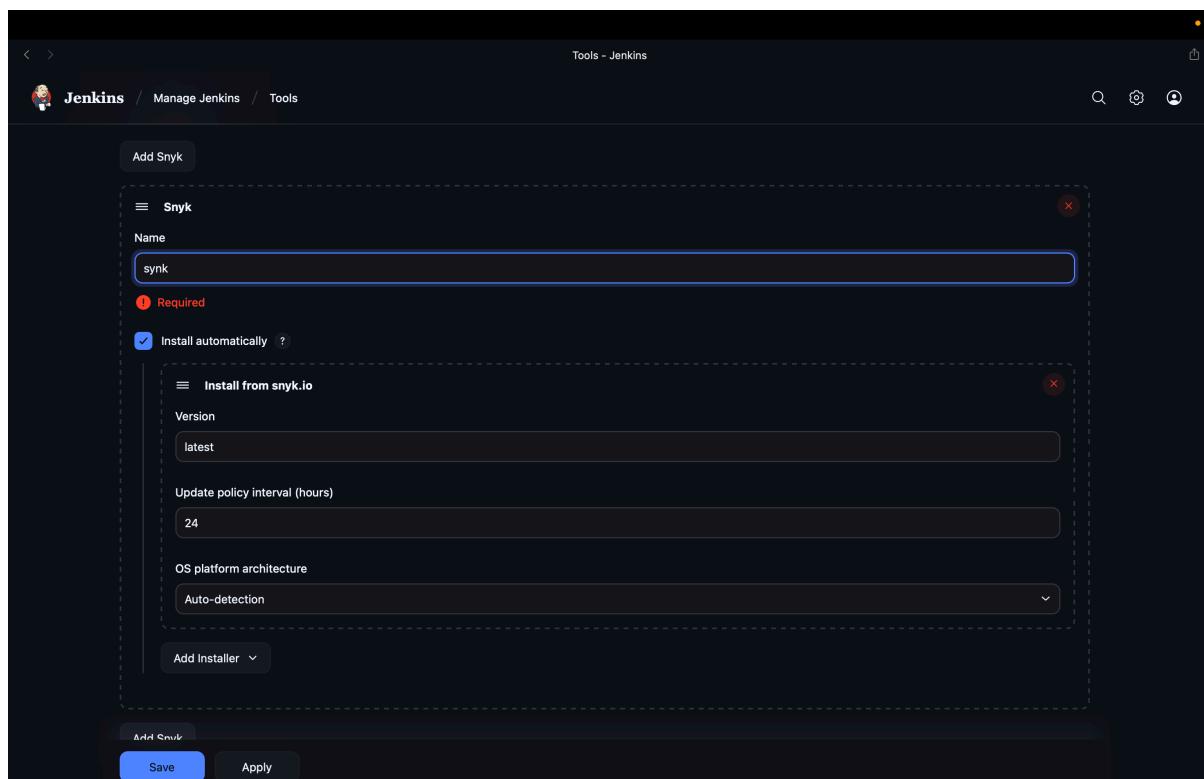
4. To add Maven, click on **Add Maven** under **Maven installations** and enter **Maven** as the Name

The screenshot shows the Jenkins Tools configuration page. At the top, it says "Tools - Jenkins" and has a back/forward navigation bar. Below that, it shows the current path: "Jenkins / Manage Jenkins / Tools". The main content area is titled "Maven installations" and includes a "Add Maven" button. A modal window is open, titled "Maven", with the following fields:

- Name**: A text input field containing "Maven".
- Required**: A red exclamation mark icon indicating it's a required field.
- Install automatically**: A checked checkbox with a question mark icon.
- Install from Apache**: A sub-section with a "Version" dropdown set to "3.9.11".
- Add Installer**: A button to add more installers.

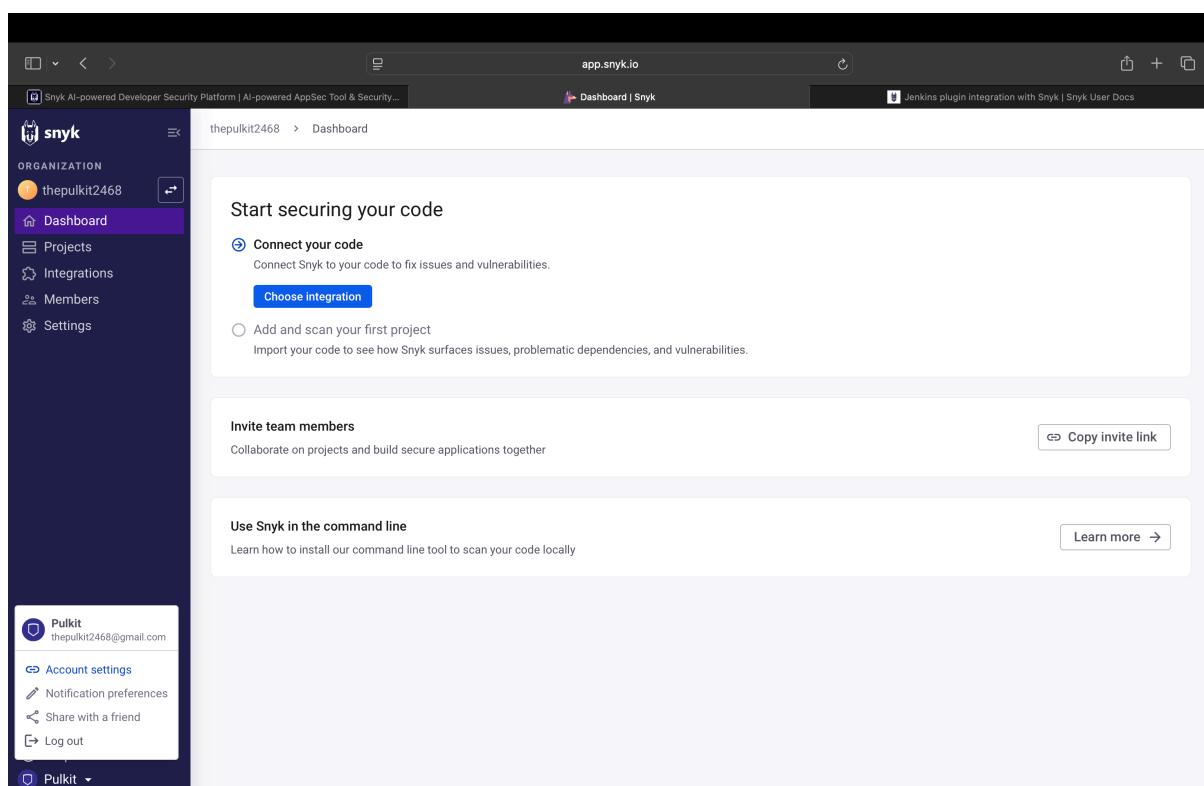
At the bottom of the modal are "Save" and "Apply" buttons.

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



Step 3: Manage Snyk API and Jenkins credentials

- 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



The screenshot shows the Snyk account settings interface. On the left is a dark sidebar with user information (the pulkit2468) and navigation links (Dashboard, Projects, Integrations, Members, Settings). The main area has a header "Account > General". Under "Auth Token", there is a section titled "Auth Token" with a sub-section "General". It contains a token key (4548d8bb-9e36-498b-93d3-46690747d2d2), its creation date (29 September 2025, 12:14:04), and a "Revoke & Regenerate" button. Below this is a section titled "Authorized Applications" which says "No applications". Under "Preferred Organization", it shows "the pulkit2468" and a "Update Preferred Org" button. At the bottom is a "Delete Account" button.

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

The screenshot shows the Jenkins Credentials management screen. The URL is "Jenkins » Credentials - Jenkins". The top navigation bar includes "Jenkins", "Manage Jenkins", and "Credentials". The main title is "Credentials". A table header includes columns "T", "P", "Store", "Domain", "ID", and "Name". Below this is a section titled "Stores scoped to Jenkins" with a table showing "System" under "Store" and "(global)" under "Domain". There is a "Add credentials" button at the bottom right of this section. The footer includes links for "REST API" and "Jenkins 2.516.3".

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

New credentials - Jenkins

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

New credentials

Kind: Snyk API token

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

Token:

ID:

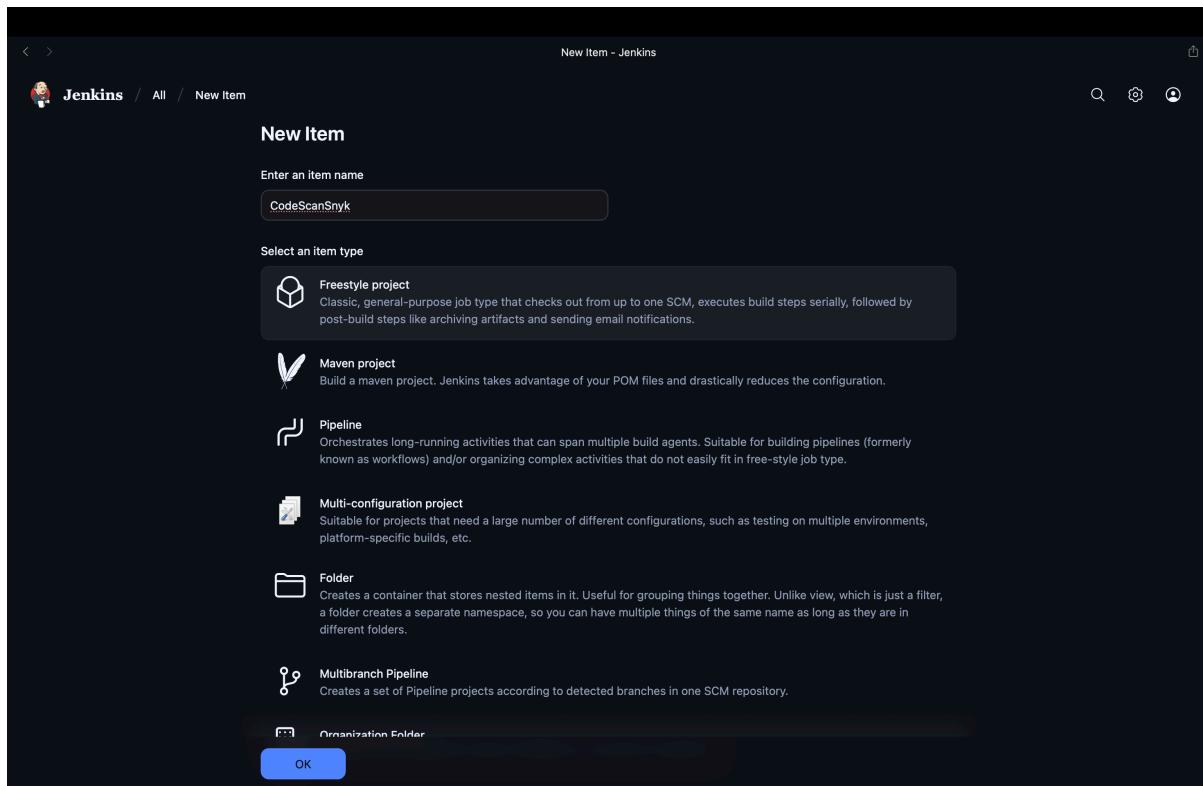
Description: snyk token

Create

REST API Jenkins 2.516.3

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**



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>

The screenshot shows the Jenkins configuration interface for a job named "CodeScanSnyk". The left sidebar has "Source Code Management" selected under "Configure". The main area is titled "Source Code Management" and contains a "Git" configuration. It includes fields for "Repository URL" (set to <https://github.com/hkshitesh/Secure-Coding.git>), "Branch Specifier" (set to */master), and "Repository browser" (set to "Auto"). There are "Save" and "Apply" buttons at the bottom.

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

The screenshot shows the Jenkins configuration interface for the same job, now in the "Build Steps" section. The left sidebar has "Build Steps" selected under "Configure". The main area is titled "Build Steps" and contains a "Invoke Snyk Security task" step. It includes options for "When issues are found": "Fail the build, if severity at or above" (selected) and "Let the build continue" (unchecked). Other checked options include "Fail the build if errors occur" and "Monitor project on build". It also includes fields for "Snyk API token" (set to "synk token"), "Target file" (empty), "Organisation" (empty), and "Project name" (empty). There are "Save" and "Apply" buttons at the bottom.

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

The screenshot shows the Jenkins job details for 'CodeScanSnyk #9'. The build was started by user 'pullkit' at 2:42:55 pm on September 29, 2025. It took 1.5 seconds and was completed 19 seconds ago. The build status is green with a checkmark. The console output shows a single line: '</> No changes.' The sidebar on the left includes links for Status, Changes, Console Output (which is selected), Edit Build Information, Delete build '#9', Timings, Git Build Data, and Previous Build.

The screenshot shows the Jenkins 'Console Output' page for 'CodeScanSnyk #9'. The log starts with 'Started by user pullkit' and 'Running as SYSTEM'. It details the git configuration and fetching of upstream changes from a GitHub repository. The log ends with 'Finished: SUCCESS'. The sidebar on the left is identical to the previous screenshot, showing the 'Console Output' link is selected.

```
Started by user pullkit
Running as SYSTEM
Building in workspace /Users/pullkit/.jenkins/workspace/CodeScanSnyk
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /Users/pullkit/.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 # 'git version 2.50.1 (Apple Git-155)'
> 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
Finished: SUCCESS
```

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

The screenshot shows the Snyk web application interface. On the left, there's a dark sidebar with a purple navigation bar containing the 'Projects' option. The main content area has a header 'palak.kharbanda > Projects'. Below the header, it says 'All projects' and 'Targets 1'. A single target is listed: 'anujdevopslearn/MavenBuild'. At the bottom of the page, there's a message 'Ready to import another project?' followed by a link 'Secure your entire stack with Snyk'.

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.