Snyk Security Plugin

V1 documentation

Getting Started

- This plugin adds Snyk Security Scanning to your project or pipeline, allowing you to test and monitor your projects for security and license issues.
- This plugin requires Docker installation on the Jenkins worker in order to scan your dependencies.
- Check https://hub.docker.com/r/snyk/snyk-cli/tags/ for a full list of runtimes and package managers supported by the Snyk CLI Docker image.
- Pull the relevant Snyk CLI docker image by running the following command: **docker pull snyk/snyk-cli:[tag]**.
- Add Jenkins user to the docker group: **sudo usermod -aG docker <jenkins-user>** and verify that the Jenkins user can run docker command without a sudo.

Job Setup and configuration

- The Snyk plugin expects to have an environment variable called **SNYK_TOKEN** that contains Snyk's API key. We recommend adding the **SNYK_TOKEN** using Jenkins Credentials plugin.
    - You can obtain your Snyk's API Key here - https://snyk.io/account
- The Snyk plugin expects to run after the installation of the project's dependencies. (i.e 'npm install' or 'mvn install')
- For Maven projects, the Snyk plugin expects to have an environment variable called **MAVEN_REPO_PATH** that contains the path to your maven repository (e.g **/home/user/.m2**)

How to use Snyk Plugin with Jenkinsfile:

Snyk pipeline integration expose **snykSecurity** function to scan your dependencies as part of your pipeline script.

Usage example: **snykSecurity**(tokenCredentialId: 'SNYK_TOKEN', failOnBuild: true, monitor: true)

This example call Snyk security with the credential Id we created using the Jenkins Credentials plugin, we choose to fail the build in case we find vulnerabilities and to take snapshot of the project current dependencies.

This function accepts the following parameters:

- ***tokenCredentialId*** *(Type: String) :* Snyk credential token id that contains Snyk's API token
- **failOnBuild** (*Type: Boolean*): Set to **true** to have the Jenkins build **FAIL** if Snyk detects issues in the project.
- **monitor** (*Type: Boolean*): Set to **true** to monitor the project on every build by taking a snapshot of its current dependencies on Snyk.io. Selecting this option will keep you notified about newly disclosed vulnerabilities and remediation options in the project.
- **organization** (*Type: String*): *OPTIONAL* - set to the Snyk organisation in which this project should be tested and monitored. Leave empty to use your default organisation.
- **packageName** (*Type: String*): *OPTIONAL* - set a custom name for the Snyk project created for this Jenkins project on every build. Leave empty for the project's name to be detected in the manifest file.
- **targetFile** (*Type: String*): *OPTIONAL* - set to the relative path of the manifest file in the project. Leave empty for Snyk to auto-detect the manifest file in the project's root folder.
- **envVars** (*Type: String*): *OPTIONAL* - set to the runtime agruments for the build tool invoked by Snyk. This is useful when you want to test a specific profile (in Maven) or configuration (in Gradle), or define system properties, such as **-Dpkg_version=1.4 -Pprod -s ./settings.xml** for Maven or **--configuration runtime -Pmyprop=myvalue** for Gradle.
- **dockerImage** (*Type: String)*: *OPTIONAL* - set to the Docker image to be used by the plugin. Leave empty to use 'snyk/snyk-cli'. Inspect the different tags at https://hub.docker.com/r/snyk/snyk-cli/tags to choose the right runtime for your project.
- **httpProxy** (*Type: String*): OPTIONAL - set to the HTTP Proxy URL to be used in the Snyk plugin Docker container. Leave empty for no proxy.
- **httpsProxy** (*Type: String*): OPTIONAL - set to the HTTPS Proxy URL to be used in the Snyk plugin Docker container. Leave empty for no proxy.

V2 Introduction

Snyk Security Scanner is a Jenkins plugin that enables Jenkins users to test their applications against the Snyk vulnerability database.

Configuration

Global Configuration

Configure your Jenkins settings to install the Snyk Security Scanner plugin:

1. Visit **Manage Jenkins > Manage Plugins > Available** and search for **Snyk Security**. Install the plugin.
2. Visit **Manage Jenkins > Global Tool Configuration** and add a **Snyk Installation** to have the Snyk CLI available during Jenkins builds. We recommend using the **latest** version to keep up to date with new releases of the Snyk CLI.

Note: in order to install a pre-released version of the plugin, change the **Update Site** to http://updates.jenkins-ci.org/experimental/update-center.json in the **Advanced** settings. See this post for more details.

Add a Snyk API Token to Jenkins to allow the Snyk Security Scanner to identify with Snyk.
Visit **Credentials > System**. Specify a meaningful credential ID value in the **ID** field (i.e. `my-org-snyk-api-token`).

Project Configuration

Freestyle Jobs

Enable the Snyk Security Scanner in the project configuration page. To add Snyk Security Scanner to the project's build, select **Build > Add build step > Invoke Snyk Security Task**.

Basic Configuration

- **When issues are found** - This specifies if builds should be failed or continued based on issues found by Snyk.
- **Monitor project on build** - Take a current application dependencies snapshot for continuous monitoring by Snyk.
- **Snyk token** - The ID for the API token from the Credentials plugin to be used to authenticate with Snyk (credential type must be "Snyk API token").
- **Target file** - The path to the application manifest file to be scanned by Snyk Security Scanner.
- **Organisation** - The Snyk organisation in which this project should be tested and monitored.
- **Project name** - A custom name for the Snyk project created for this Jenkins project on every build.

Advanced Configuration

To see the advanced configuration for the plugin, click the **Advanced** button. This section allows you to specify Snyk installation as well as additional runtime arguments for the Snyk Security Scanner.

- **Snyk installation** - The Snyk installation as configured in the **Global Tool Configuration**.
- **Additional arguments** - Refer to the Snyk CLI help page for information on additional arguments.

Pipeline Jobs

The Snyk Security Scanner pipeline integration exposes the **snykSecurity** function to scan your dependencies as part of your pipeline script. We recommend to use the "Snippet Generator" to generate the needed step statement to copy into your Jenkinsfile.

The **snykSecurity** function accepts the following parameters:

- **snykInstallation** - Snyk installation configured in the **Global Tool Configuration**.
- **snykTokenId** - The ID for the API token from the Credentials plugin to be used to authenticate to Snyk.
- **additionalArguments** (optional, default none) - Refer to the Snyk CLI help page for information on additional arguments.
- **failOnIssues** (optional, default **true**) - This specifies if builds should be failed or continued based on issues found by Snyk.
- **organisation** (optional, default none) - The Snyk organisation in which this project should be tested and monitored.
- **projectName** (optional, default none) - A custom name for the Snyk project created for this Jenkins project on every build.
- **severity** (optional, default **low**)- Only report vulnerabilities of provided level or higher (low/medium/high).
- **targetFile** (optional, default none) - The path to the manifest file to be used by Snyk.

Migration from v1

Note: the new v2 of the plugin contains incompatible changes to v1 and will require you to adapt your Jenkins jobs. You have to perform global configuration steps as described [here|#global-configuration].

- The plugin does not requires Docker installation on master or worker nodes. Add a Snyk installer in the **Global Tool Configuration** section.
- You don't need to pass Snyk API token as {{SNYK_TOKEN}} environment variable to the job. Add a credential of type Snyk API token.
- Parameters 'Runtime Arguments', 'Docker Image', 'HTTP Proxy', 'HTTPS Proxy' are obsolete.
- Pipeline syntax was changed, see 'Pipeline jobs' section for documentation.