

|  |  |  |
| --- | --- | --- |
| Document name | DevSecOps: Snyk | System Recruitment Specification |
| Version no. | 1.0 | 1.2 |
| Release date | 12th oct 2023 | 1st Oct ‘09 |
| Document ID | *1* |  |
| Classification | Internal |  |

|  |
| --- |
| Snyk |

# Document History

| Ver. No | Release Date | Created By / Modified By and Date | Reviewed By and Date | Approved By and Date | Remarks and Changes Made |
| --- | --- | --- | --- | --- | --- |
| 1.0 | 12tth Oct 2023 | Ruturaj Kharde / 12th Oct 2023 |  |  |  |

# Distribution List

<Guidelines: Mention the name of all the members to whom the LLD will be distributed. >

| Sr. No. | Name | Role | Organization Name |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Contents

[Document History 2](#_Toc149472438)

[*Distribution List* 2](#_Toc149472439)

[1. Introduction: 4](#_Toc149472440)

[What is Snyk? 4](#_Toc149472441)

[The Snyk developer-first approach 4](#_Toc149472442)

[Use Snyk in your workflow 4](#_Toc149472443)

[Choose how to run Snyk 4](#_Toc149472444)

[2. Implementation steps with Repos 6](#_Toc149472445)

[**2.1** **Integration with Azure DevOps Repository** 6](#_Toc149472450)

[**2.2** **Add Projects to Snyk From Azure Repos** 7](#_Toc149472452)

[**2.3** **Note: for detailed repo integration visit below link** 8](#_Toc149472453)

[3. Implementation steps with Azure Build Pipeline 9](#_Toc149472454)

[**3.1** **Integration with Azure DevOps CI pipeline** 9](#_Toc149472459)

[**3.2** **Note: for detailed pipeline integration visit below link** 11](#_Toc149472460)

[4. Implementation steps with CLI(Local scanning) 12](#_Toc149472461)

[**4.1** **Integration with CLI** 12](#_Toc149472466)

[**4.2** **Note: for Other Operating System integration using CLI visit below link** 14](#_Toc149472467)

# Introduction:

## What is Snyk?

Snyk is a platform allowing you to scan, prioritize, and fix security vulnerabilities in your code, open source dependencies, container images, and infrastructure as code configurations

## The Snyk developer-first approach

Developers now assemble applications with a combination of proprietary and open-source code, run that code in containers, and then deploy with infrastructure as code configurations using technologies like Kubernetes and Terraform.

A good security process secures each of these components where they are built and maintained. Snyk integrates into DevOps processes to work with developers using the methods each prefers, while following and supporting industry best practices. Snyk integrates directly into your IDEs, workflows, and automation pipelines to add security expertise to your toolkit.

## Use Snyk in your workflow

* **Secure your code**: use [Snyk Open Source](/scan-applications/snyk-open-source) to fix vulnerabilities in your open source dependencies and [Snyk Code](/scan-applications/snyk-code) to fix vulnerabilities in your source code.
* **Secure your containers**: use [Snyk Container](/scan-applications/snyk-container) to fix vulnerabilities in container images and Kubernetes applications.
* **Secure your deployment**: use [Snyk Infrastructure as Code (IaC)](/scan-infrastructure/snyk-infrastructure-as-code) to fix misconfigurations in Terraform, CloudFormation, Kubernetes, and Azure templates. Use [IaC+](/scan-infrastructure/snyk-iac+) to fix misconfigurations in Amazon Web Services accounts, Microsoft Azure subscriptions, and Google Cloud projects.

## Choose how to run Snyk

You can run Snyk in the following ways:

* [**Web**](/getting-started/exploring-the-snyk-web-ui): the Snyk Web UI ([app.snyk.io](https://app.snyk.io)) provides a browser-based experience, along with functions such as configuration settings, filtering and fixing discovered issues, and reports.
* [**CLI**](/snyk-cli): the Snyk Command Line Interface enables you to run vulnerability scans on your local machine and integrate Snyk into your pipeline.
* [**IDEs**](/integrations/ide-tools): the Snyk IDE integrations enable you to embed Snyk in your development environment.
* [**API**](/snyk-api): the Snyk API enables you to integrate with Snyk programmatically, tuning Snyk security automation to your specific workflows.

# Implementation steps with Repos



## **Integration with Azure DevOps Repository**

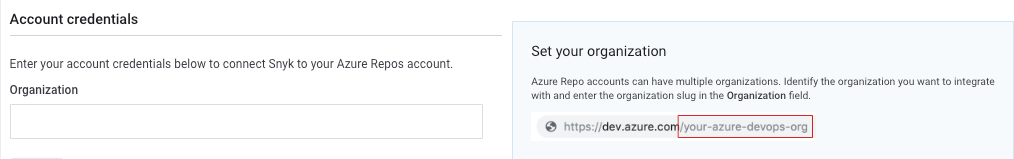
## Step 1 : Log/signup in to your Snyk account and navigate to Integrations **https://snyk.io**

(You can di the signup using Google account or GitHub account)

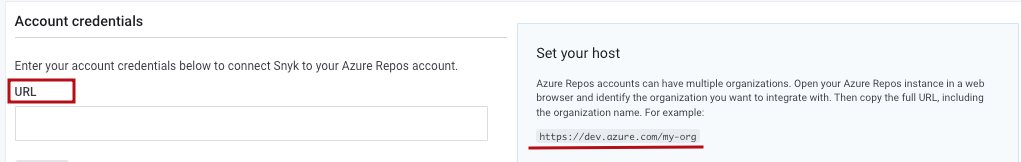
Step 2 : On the Azure Repos tile, click the settings icon to open Organization Settings > Integrations > Azure Repos > Account credentials.

Step 3: Pay special attention to the instructions given on the **Account Credentials** page. Depending on your plan, you may need to enter just the Azure DevOps Organization, or you may need to enter the entire URL.

* **Set your host**: enter the entire url. For example, enter https://dev.azure.com/your-azure-devops-org Alternatively, you may enter a custom url that is publicly reachable



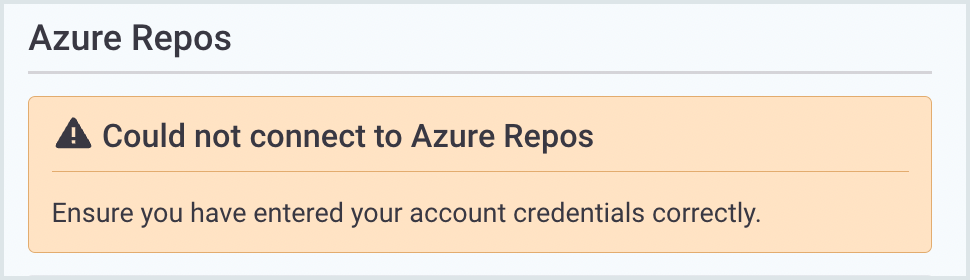
* **Set your host**: enter the entire url. For example, enter https://dev.azure.com/your-azure-devops-org Alternatively, you may enter a custom url that is publicly reachable



Step 4 : Click Save, and then enter the PAT that you generated.

Step 5: Click **Save**. Snyk tests the connection values and the page reloads, displaying the Azure Repos integration information. A message to confirm that the details were updated appears at the top of the screen.

If the connection to Azure fails, a notification appears under the **Azure Repos** card title.



## **Add Projects to Snyk From Azure Repos**

To add a default Project:

Step 1: In Snyk, navigate to Projects > Add projects.

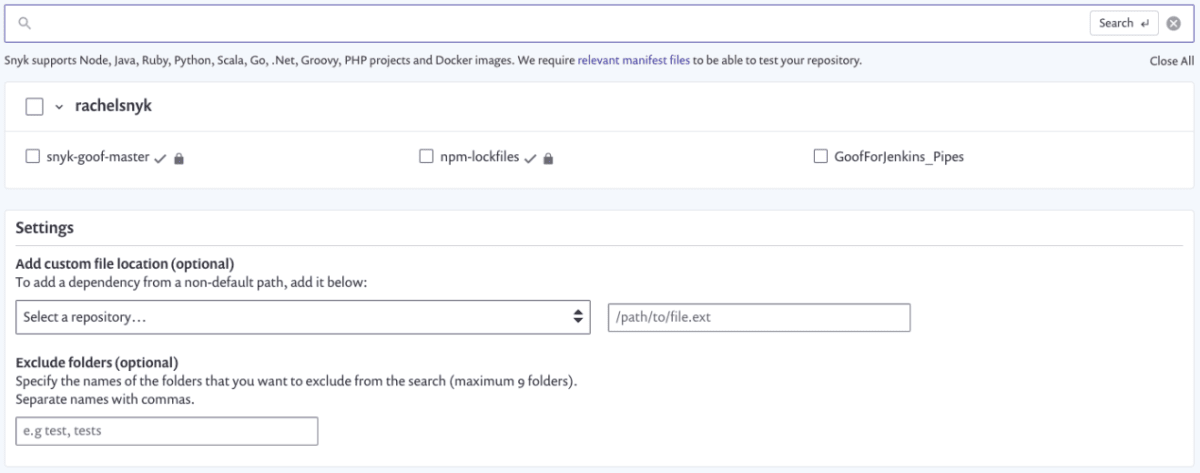
Step 2: Choose the relevant repository or tool from which to import your projects.

Step 3: The available repositories for the integration you chose are displayed in a new window.

Step 4: Select the repositories that you want Snyk to monitor for security and license issues.

Step 5: To import all the repos for a specific Organization, check the Organization.

Step 6: Click Add selected repositories.Snyk scans the entire file tree for dependency files and imports them to Snyk as Projects.



## **Note: for detailed repo integration visit below link**

[Snyk Azure Repositories (TFS) integration - Snyk User Docs](https://docs.snyk.io/integrations/git-repository-scm-integrations/snyk-azure-repositories-tfs-integration)

# Implementation steps with Azure Build Pipeline



## **Integration with Azure DevOps CI pipeline**

Step 1: Access your Snyk account.

Step 2: Token:

2.1 For free plans, go to your General Account Settings and find, copy, and save your personal API authentication token on the side.

2.2 For paid plans, navigate to the Organization where you want to integrate; then go to Settings to create a new service account token. Copy and save it on the side.

Step 3: Access your Azure DevOps account and navigate to the Extensions -> Browse marketplace.

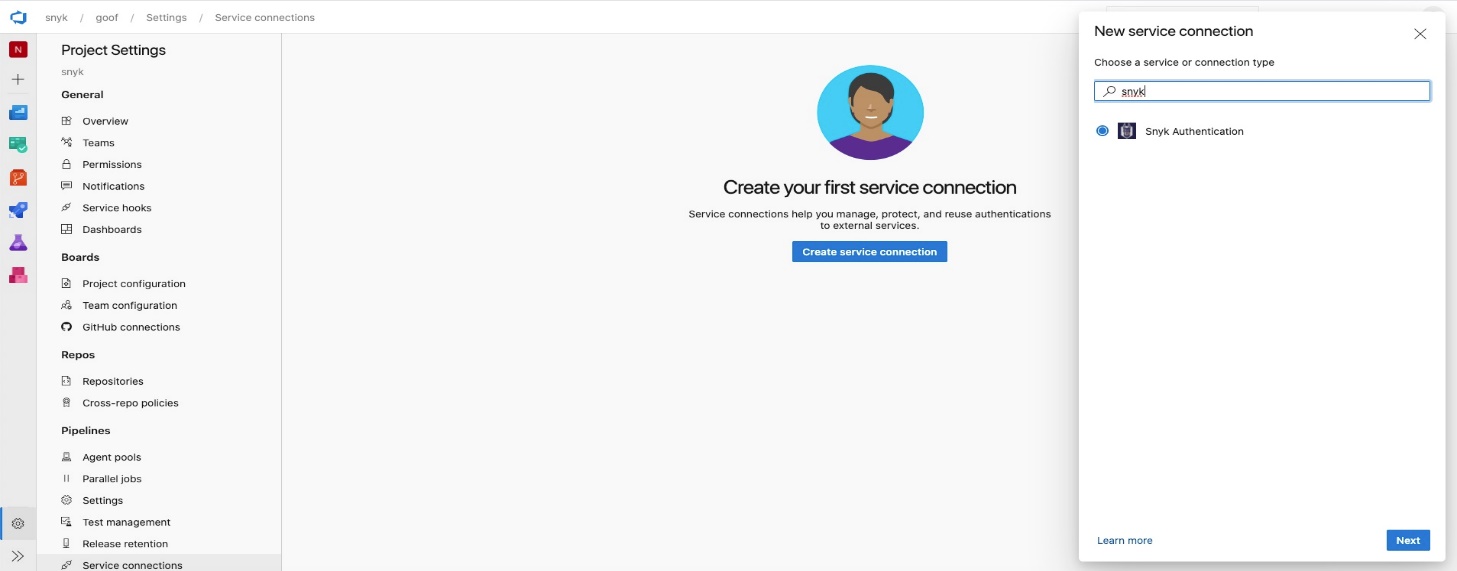
Step 4: Search for the Snyk Security Scan extension and click Get it free.

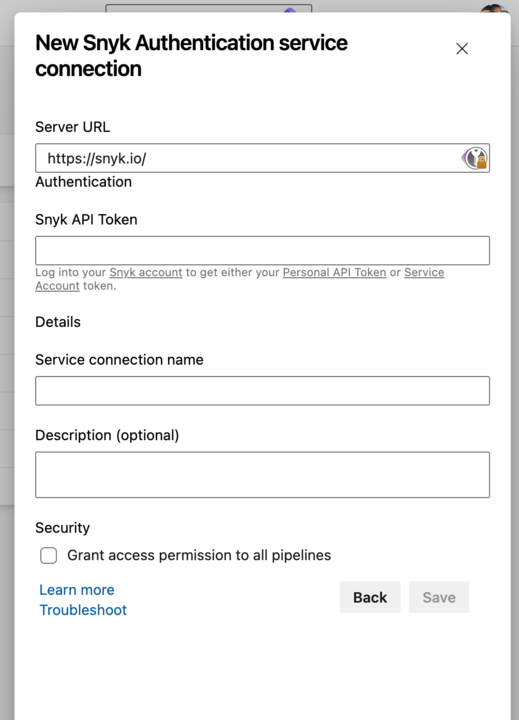
Step 5: Create a new Service Connection in your Project via Project Settings —> Pipelines —> Service Connections.

Step 6: Select the Snyk Authentication service connection:

6.1 In the Snyk Authentication service connection form, enter the Snyk API Token.

6.2 Click Save, ensuring the new service connection appears in your list of service connections.





## **Note: for detailed pipeline integration visit below link**

[Install the Snyk extension for your Azure pipelines - Snyk User Docs](https://docs.snyk.io/integrations/snyk-ci-cd-integrations/azure-pipelines-integration/install-the-snyk-extension-for-your-azure-pipelines)

# Implementation steps with CLI(Local scanning)



## **Integration with CLI**

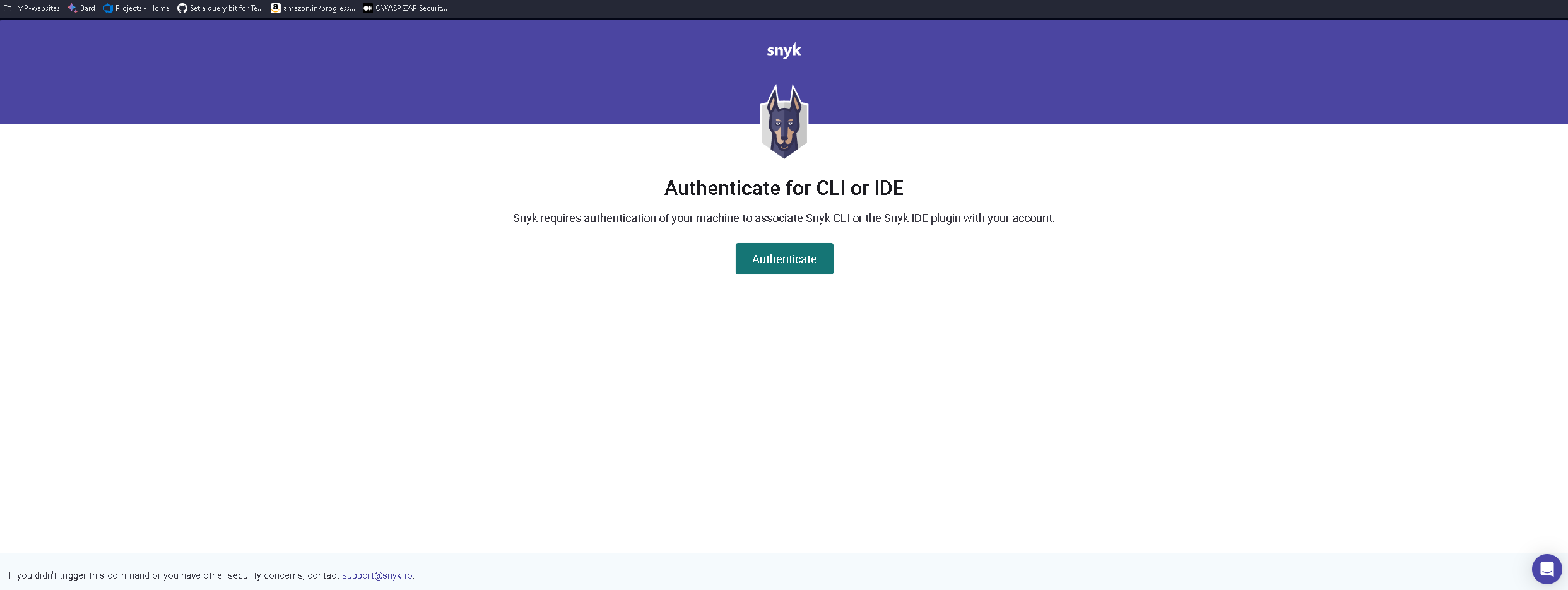
Step 1: Open PowerShell in admin mode.

Step 2: For window execute the below command

*curl* [*https://static.snyk.io/cli/latest/snyk-win.exe -o snyk.exe*](https://static.snyk.io/cli/latest/snyk-win.exe%20-o%20snyk.exe)

Step 3: An Snyk executable file would have been downloaded in your system, add its path in environment variable.

Step 4: Execute the command snyk *auth ,* It will open the interface shown in below image on browser . Click on **Authenticate**



Step 5: Now to do the analysis run the below command depending on your requirements from the root path of the project

* Scanning of project with source code

*snyk code test --org=2c3eda80-fad0-4717-8a34-026e93847424*

|  |  |
| --- | --- |
| snyk code test | Scans your source code for vulnerabilities introduced by your first party code. |
| org=<ORG\_ID> | Specifies the Organization ID to run Snyk commands for a specific organization. It influences where new projects are created after running the monitor command. Find the Org ID on your Organization’s Settings page (Settings > General). |
|  |  |

* Scanning of Containers

To scan container images for vulnerabilities copy the command below and specify the container image by **replacing <repository> and <tag>**.

1. To scan your projects locally

*snyk container test <repository>:<tag> --org=2c3eda80-fad0-4717-8a34-026e93847424*

1. To continuously monitor your projects and view the latest snapshots in Snyk Dashboard

*snyk container monitor <repository>:<tag> --org=2c3eda80-fad0-4717-8a34-026e93847424*

|  |  |
| --- | --- |
| snyk container test | Scans your container images for any known vulnerabilities. |
| snyk container monitor | Captures the container image layers and dependencies and monitor for vulnerabilities. View the latest snapshots and scan results in the Web UI, on the Projects page. |
| --all-projects | Auto-detects all projects in the working directory. |
| --org=<ORG\_ID> | Specifies the Organization ID to run Snyk commands for a specific organization. It influences where new projects are created after running the monitor command. Find the Org ID on your Organization’s Settings page (Settings > General). |

## **Note: for Other Operating System integration using CLI visit below link**

[CLI | Snyk](https://app.snyk.io/org/abcruturaj/add/cli)