Pipeline Security: CI with Snyk using GitHub Actions

GitHub Actions can be used as a CI tool for building, testing and deploying our code. With the aid of Synk, it can also automate the process of checking vulnerabilities.

Table of Contents

Introduction

Snyk is a developer security platform for securing code, dependencies, containers, and infrastructure as code.

It can be used in IDE such as IntelliJ, Visual Studio Code and so on as a tool to scan vulnerabilities in the code and librabries.

It can also be used to secure containers by finding and fixing issues in containers and continuously monitor container images.

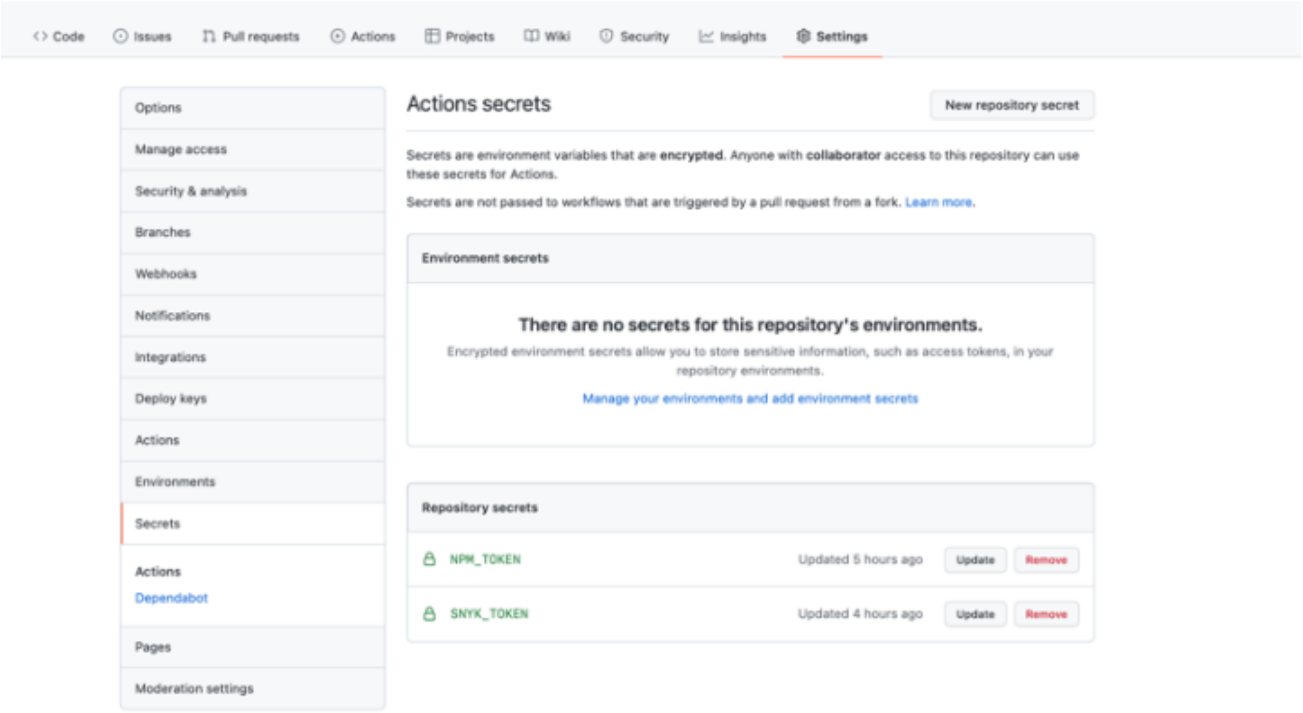
It can also be integrated with CI/CD tool to find and fix issues in application continuously.

In this article, we will integrate Synk with GitHub Action, a CI tool from GitHub, in order to secure our application continuously.

Create Action Secrets in GitHub

It is required to submit Access Tokens to Snyk and Npm Registry for authentication when performing security check and publishing respectively.

Action Secrets are environment variables that are encrypted. It is useful to store sensitive information such as access tokens. Action Secrets can be used in Action Workflow file.



Actions secrets

Secrets are environment variables that are **encrypted**. Anyone with **collaborator** access to this repository can use these secrets for Actions.

Secrets are not passed to workflows that are triggered by a pull request from a fork. Learn more.

Environment secrets

There are no secrets for this repository's environments.

Encrypted environment secrets allow you to store sensitive information, such as access tokens, in your repository environments.

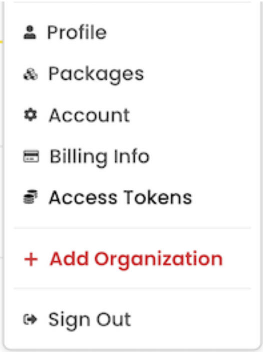
Manage your environments and add environment secrets

Repository secrets

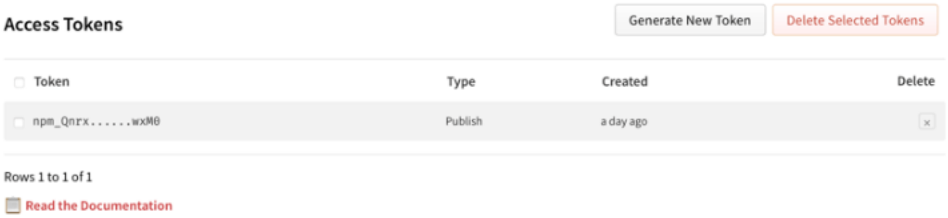
NPM_TOKEN	Updated 5 hours ago
SNYK_TOKEN	Updated 4 hours ago

Generate Access Token for publishing packages to npm Registry.

1. Login npm.
2. Click "Access Tokens" on the popup menu shown when the profile image is clicked.



3. Click the "Generate New Token" button on the "Access Tokens" page.



Access Tokens

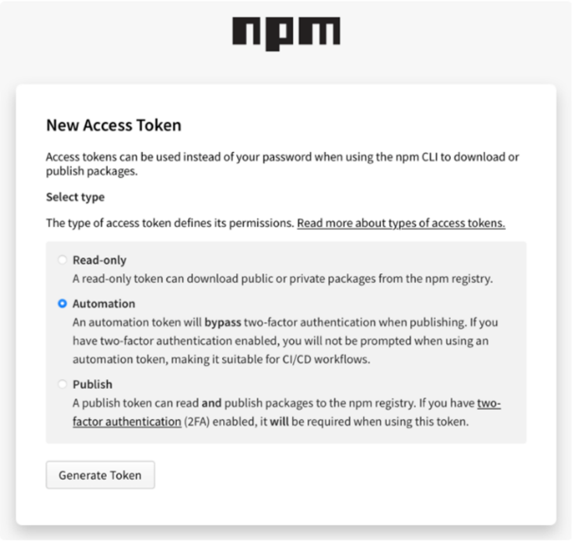
Generate New Token | Delete Selected Tokens

□ Token	Type	Created	Delete
□ npm_Qnrx......wxM0	Publish	a day ago	×

Rows 1 to 1 of 1

Read the Documentation

4. In the "New Access Token" page shown, select "Automation" from the "Select type" list, and click "Generate Token".



New Access Token

Access tokens can be used instead of your password when using the npm CLI to download or publish packages.

Select type

The type of access token defines its permissions. Read more about types of access tokens.

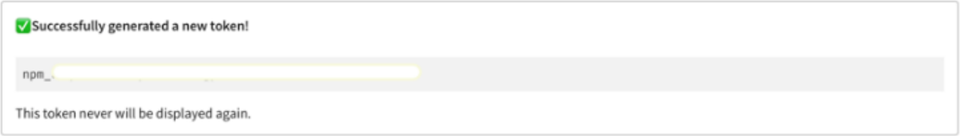
○ Read-only
A read-only token can download public or private packages from the npm registry.

● Automation
An automation token will **bypass** two-factor authentication when publishing. If you have two-factor authentication enabled, you will not be prompted when using an automation token, making it suitable for CI/CD workflows.

○ Publish
A publish token can read **and** publish packages to the npm registry. If you have two-factor authentication (2FA) enabled, it **will** be required when using this token.

Generate Token

5. A new Token should then be generated. Copy the token for later use.



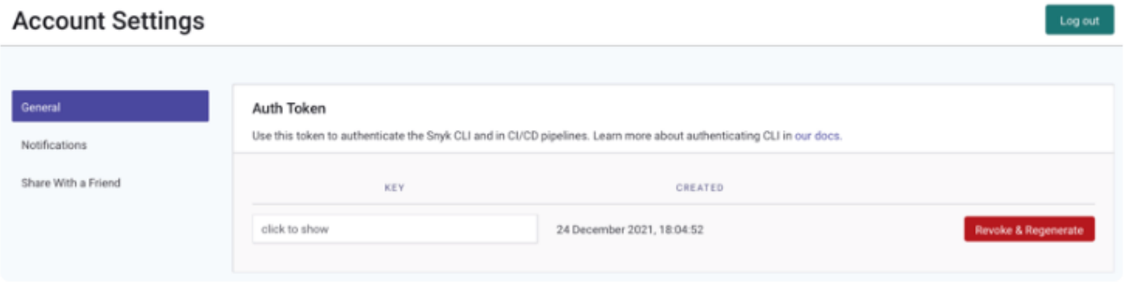
✅Successfully generated a new token!

npm_

This token never will be displayed again.

Generate Auth Token for authentication to Snyk.

1. Login Snyk.
2. Click Account Settings > API Token section.
3. In the KEY field, click "click to show", then select and copy your token.



Account Settings

Auth Token

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

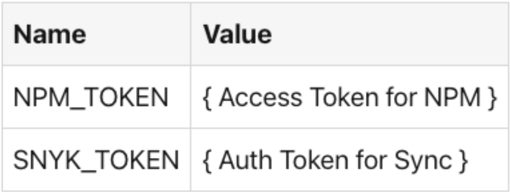
KEY	CREATED	
click to show	24 December 2021, 18:04:52	Revoke & Regenerate

Create Action Secrets

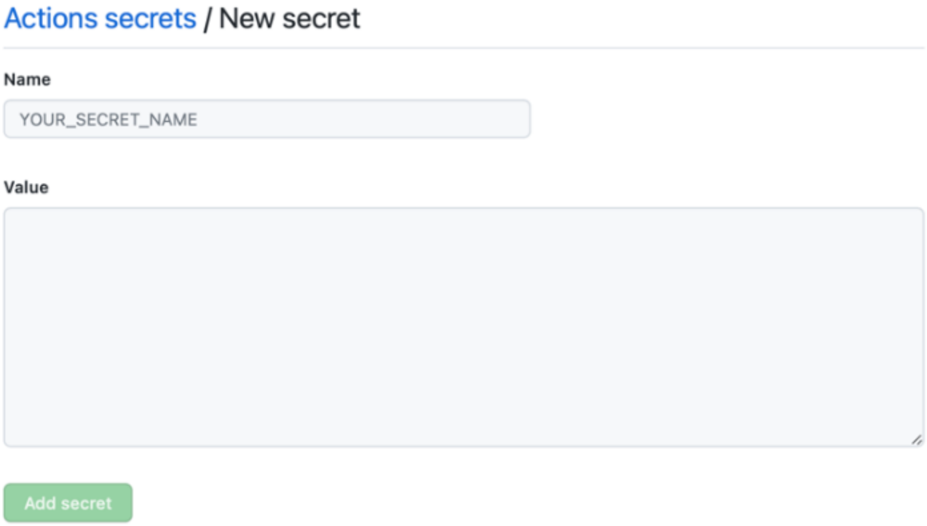
Create Actions Secrets NPM_TOKEN and SNYK_TOKEN for access to npm repository and Snyk respectively.

1. Login GitHub.
2. Click the target repository.
3. Select the "Settings" tab.

4. On the "Settings" page, select "Secrets" on the left navigation menu. "Action secrets" page should be shown.
5. Click the "New repository secret" button. A "New secret" page should be shown.
6. Input "Name" and "Value" of the tokens, and then click the "Add secret" button.

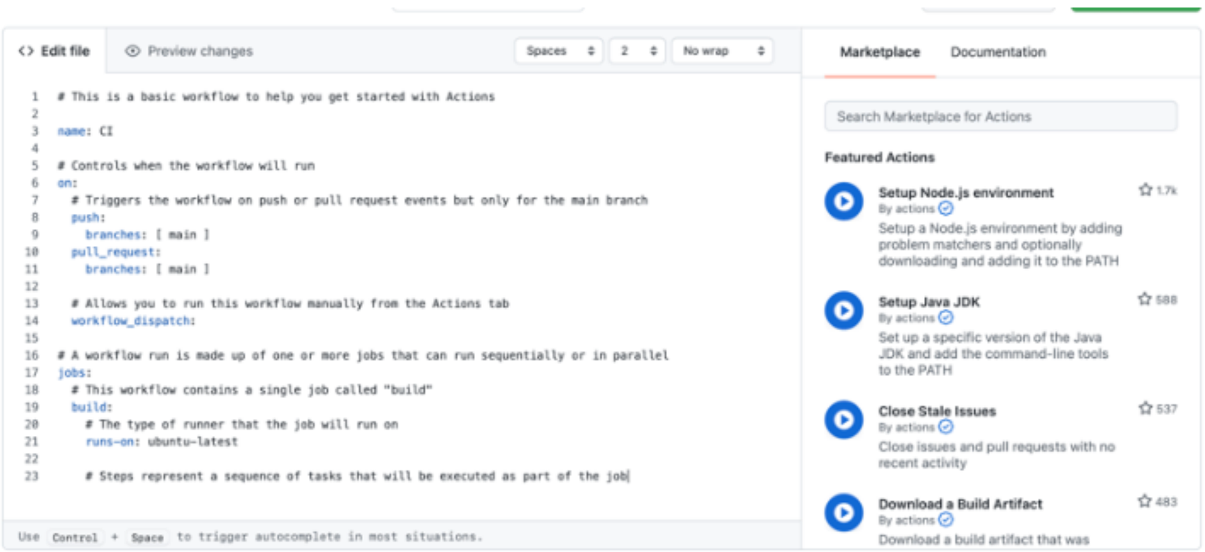


Name	Value
NPM_TOKEN	{ Access Token for NPM }
SNYK_TOKEN	{ Auth Token for Sync }



Create a GitHub Action

1. Open the target GitHub repository in browser.
2. Click on the *Actions* tab.
3. Click the link on "set up a workflow yourself".



```
# This is a basic workflow to help you get started with Actions

name: CI

# Controls when the workflow will run
on:
  # Triggers the workflow on push or pull request events but only for the main branch
  push:
    branches: [ main ]
  pull_request:
    branches: [ main ]

  # Allows you to run this workflow manually from the Actions tab
  workflow_dispatch:

# A workflow run is made up of one or more jobs that can run sequentially or in parallel
jobs:
  # This workflow contains a single job called "build"
  build:
    # The type of runner that the job will run on
    runs-on: ubuntu-latest

    # Steps represent a sequence of tasks that will be executed as part of the job
```

Use Control + Space to trigger autocomplete in most situations.

Featured Actions

Setup Node.js environment ☆ 1.7k
By actions
Setup a Node.js environment by adding problem matchers and optionally downloading and adding it to the PATH

Setup Java JDK ☆ 588
By actions
Set up a specific version of the Java JDK and add the command-line tools to the PATH

Close Stale Issues ☆ 537
By actions
Close issues and pull requests with no recent activity

Download a Build Artifact ☆ 483
By actions
Download a build artifact that was

Compose GitHub Action workflow file.

A basic GitHub Action workflow file consists of 3 secions:

- ***name***: Action Name
- ***on***: How the action will be triggered.
- ***jobs***: Jobs to be performed when the Action is triggered.

1. Update ***name*** section.

```
name: CI Publish, with security check using Snyk
```

2. Keep ***on*** section unchanged. By default, the action is triggered when a push or a pull request occurs.

```
# Controls when the workflow will run
on:
  # Triggers the workflow on push or pull request events but only for the
  push:
    branches: [ main ]
  pull_request:
    branches: [ main ]
```

3. Update ***jobs*** section.

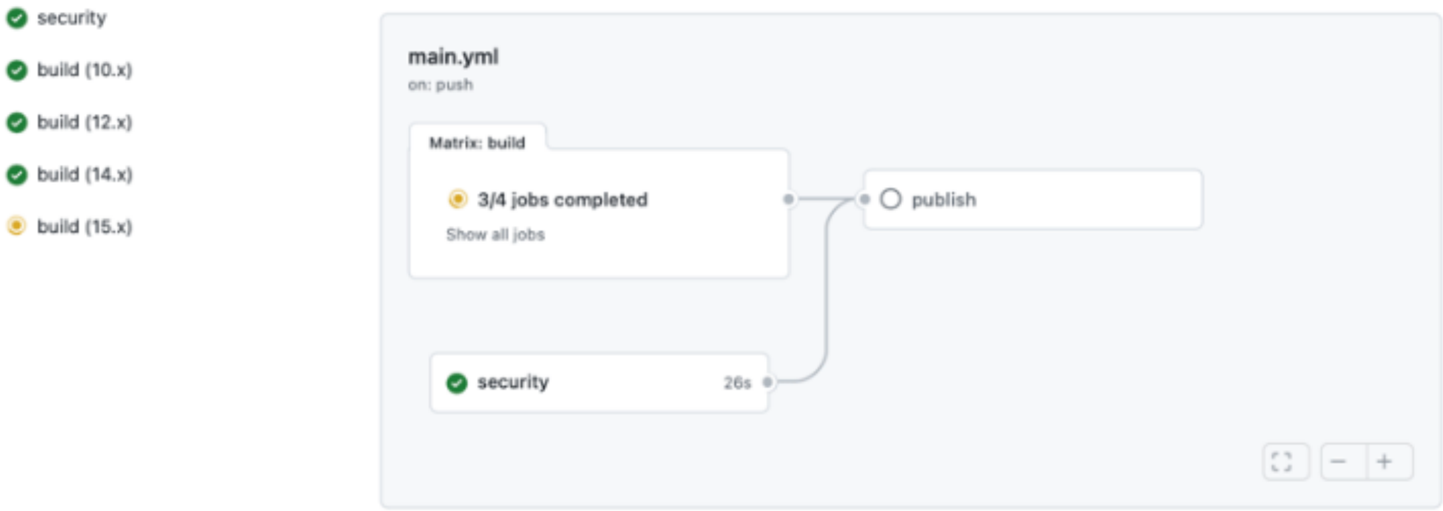
There are 3 jobs that are set up in this Action:

i. ***security***: Use Snyk to check for any vulnerability.

ii. ***build***: This job is used to build the code. In this example, we build a Node.js application with various Node versions defined in an array. This allows us to test the application running on different Node versions in a very easy approach.

iii. ***publish***: Publish the package to npm repository, allowing other developers to download and install the package, simple using the npm install command.

To set up a job that depends on other job(s) to be run successfully, `needs` can be used. For example, `needs: [security, build]` means that the job ***publish*** requires the jobs ***security*** and ***build*** to be executed successfully before it can be run. If either of the jobs fails, the ***publish*** job will not be executed.



Here below list the entire workflow file:

```
# A workflow run is made up of one or more jobs that can run sequentially or in parallel
jobs:
  security:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@master
      - name: Run Snyk to check for vulnerabilities
        uses: snyk/actions/node@master
        env:
          SNYK_TOKEN&#58; &#36;&#123;&#123; secrets.SNYK_TOKEN &#125;&#125;
        with:
          command: monitor

  build:
    runs-on: ubuntu-latest
    strategy:
      matrix:
      node-version: [10.x, 12.x, 14.x, 15.x]

    steps:
      # Checks-out your repository under $GITHUB_WORKSPACE, so your job can access it
      - uses: actions/checkout@v2
      - name: Use Node.js &#36;&#123;&#123; matrix.node-version &#125;&#125;
        uses: actions/setup-node@v2
        with:
          node-version&#58; &#36;&#123;&#123; matrix.node-version &#125;&#125;
      - name: Install dependencies
        run: npm ci
      - run: npm run build --if-present
      - run: npm test

  publish:
    needs: [security, build]
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v2
      - name: Use Node.js
        uses: actions/setup-node@v2
        with:
          node-version: '15.x'
    registry-url: 'https://registry.npmjs.org'
  - name: Install dependencies
   run: npm ci
  - name: Publish
   run: npm publish
   env:
    NODE_AUTH_TOKEN&#58; &#36;&#123;&#123; secrets.NPM_TOKEN &#125;&#125;
```

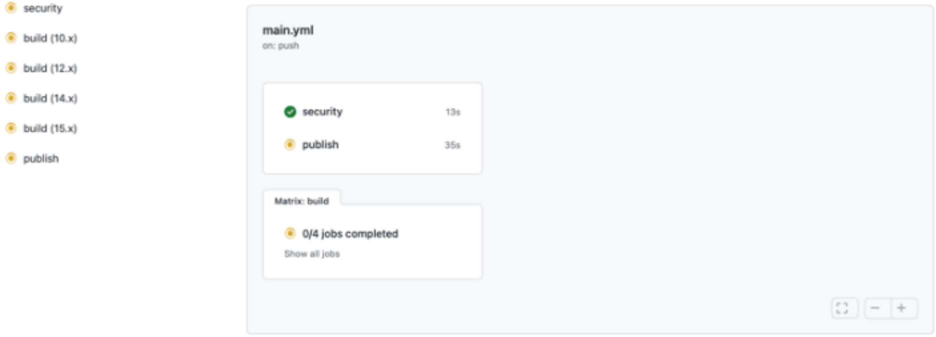
Commit the workflow file.

- Click "Start Commit" button on the left.
- Input description. It is better to input the Action Name, since it will be shown in the Action History.
- Click "Commit changes" button.
- Once the "Commit changes" button is clicked, the Action will be triggered to execute.

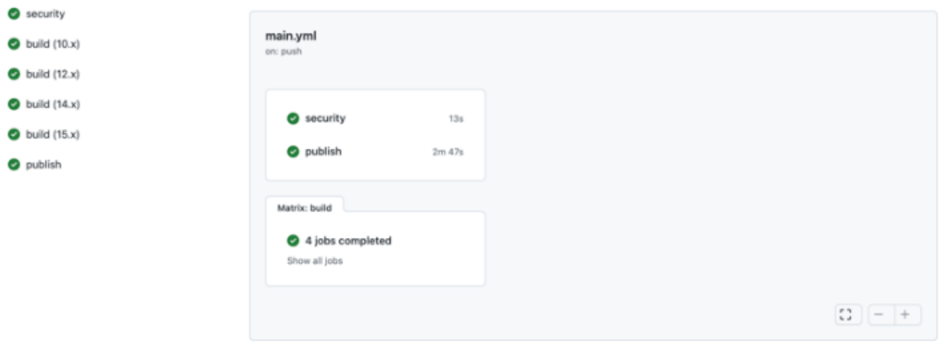
Run the GitHub Action

Our GitHub Action action can be triggered when the workflow file is updated, push or pull request occurs.

1. Once the Action is triggered, the defined jobs will be run.



2. Once the Action is completed successfully, a green tick will be shown.



```
security
build (10.x)
build (12.x)
build (14.x)
build (15.x)
publish
```

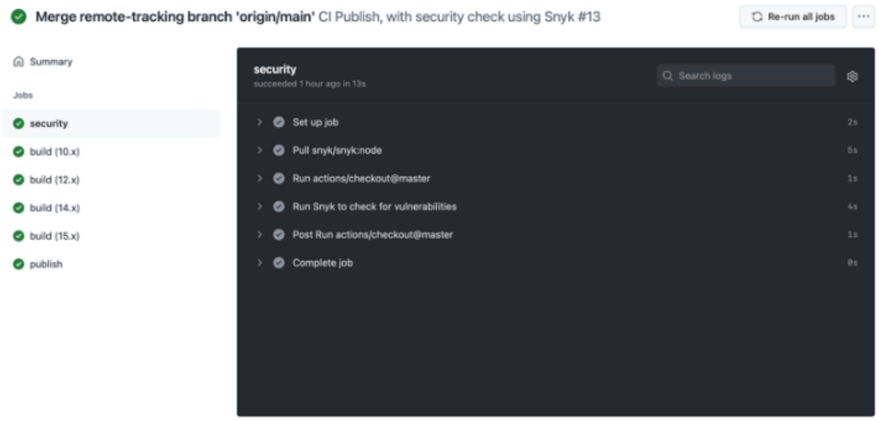
main.yml
on: push

```
security      13s
publish       2m 47s
```

Matrix: build

4 jobs completed

3. Upon completion, check the security job details by clicking the ***security*** link on the summary panel on the left.



Merge remote-tracking branch 'origin/main' CI Publish, with security check using Snyk #13

```
security
build (10.x)
build (12.x)
build (14.x)
build (15.x)
publish
```

security
succeeded 1 hour ago in 13s

```
Set up job                                   2s
Pull snyk/snyk:node                          5s
Run actions/checkout@master                  1s
Run Snyk to check for vulnerabilities        4s
Post Run actions/checkout@master             1s
Complete job                                 0s
```

```
security:
  runs-on: ubuntu-latest
  steps:
    - uses: actions/checkout@master
    - name: Run Snyk to check for vulnerabilities
      uses: snyk/actions/node@master
      env:
        SNYK_TOKEN&#58; &#36;&#123;&#123; secrets.SNYK_TOKEN &#125;&#125;
      with:
       command: monitor
```

With ***monitor command***, the scan is performed and report is generated, but the process will not be interrupted. In other words, even if vulnerabilities are found, the job is finished successfully without error and next job will not be interfered and will then start.

Performance will be evaluated based on the following:

To view the report, open the link stated as "Explore this snapshot at" in the result of "Run Snyk to check for vulnerabilities" section of the security job in a browser.

- Can you see this!!!! Please Show Me
- Check whether the package is published in NPM repository or not (Optional)