

CLI cheat sheet

1. Installation

```
$ npm install -g snyk  
$ brew tap snyk/tap && brew install snyk  
$ scoop bucket add snyk  
https://github.com/snyk/scoop-snyk  
$ scoop install snyk
```

2. Authenticate

You need to authenticate with your Snyk account in one of the following ways

```
$ snyk auth (authenticate via browser)  
$ snyk auth <token> (for local testing)
```

Set environment variable `SNYK_TOKEN`. Can be used for local testing and CI pipeline

3. Test your application

Scan your project for vulnerabilities locally by running

```
$ snyk test
```

from the root of your project.

Vulnerabilities and remediation will be shown in your console.

4. Monitor

Monitor your application for vulnerabilities by sending a snapshot of the dependencies to your snyk dashboard using:

```
$ snyk monitor
```

The results are shown on your Snyk dashboard.

5. Ignore vulnerabilities

Ignore a specific vulnerability for 30 days by using `snyk ignore` as seen below:

```
$ snyk ignore --id=npm:tough-cookie:20160722  
Add a custom expiration date and reason by using the flags:  
--expiry=2020-11-11  
--reason='Not currently exploitable'
```

6. Common CLI options

```
snyk [cmd] --file=package.json  
Specify a manifest file you want to test:
```

```
$ snyk test --file=req.txt --package-manager=pip  
Specify GitHub manifest file and ecosystem
```

```
$ snyk test --all-projects  
Test all manifest in folder and subfolder
```

```
$ snyk test --dev  
Include devDependencies in the scan
```

```
$ snyk [cmd] --org=my-team  
Run command for a specific organization
```

```
$ snyk test https://github.com/snyk/goof  
Test a github repository (npm only)
```

```
$ snyk test lodash  
Test the latest version of a package
```

```
$ snyk test ionic@1.6.5  
Test a specific version of a package
```

```
$ snyk test --json  
Output full test results as json
```

```
$ snyk test --json-file-output=vuln.json  
Store test result as json file
```

```
$ snyk [cmd] --severity-threshold=<low|medium|high>  
Set severity level that will be visible
```

7. Container Scanning

Scan and monitor Docker images:

```
$ snyk container test <image>  
$ snyk container monitor <image>
```

Add a Dockerfile for base-image remediation advice using:
`--file=path/to/Dockerfile`

Scan Docker and OCI container images as seen below:

```
$ snyk container test docker-archive:container.tar  
$ snyk container test oci-archive:container.tar
```

8. Infrastructure as Code (IAC)

Scan Kubernetes and Terraform files for security issues using:

```
$ snyk iac test /path/to/Kubernetes.yaml  
$ snyk iac test /path/to/terraform_file.tf
```

9. Troubleshooting

Help section and options explained

```
$ snyk help
```

Debug output for your command

```
$ snyk [cmd] -d
```

Unexpected results?

Build your project and download all dependencies first, for example:

```
$ npm install  
$ mvn install  
$ dotnet restore  
$ dep ensure
```

Running out of test on an OS project:

- run `snyk monitor`
- open the Snyk UI and go to the settings of the project
- enter the URL of your OS repo in “Git remote URI”



Author

@BrianVerm
Developer Advocate
at Snyk

