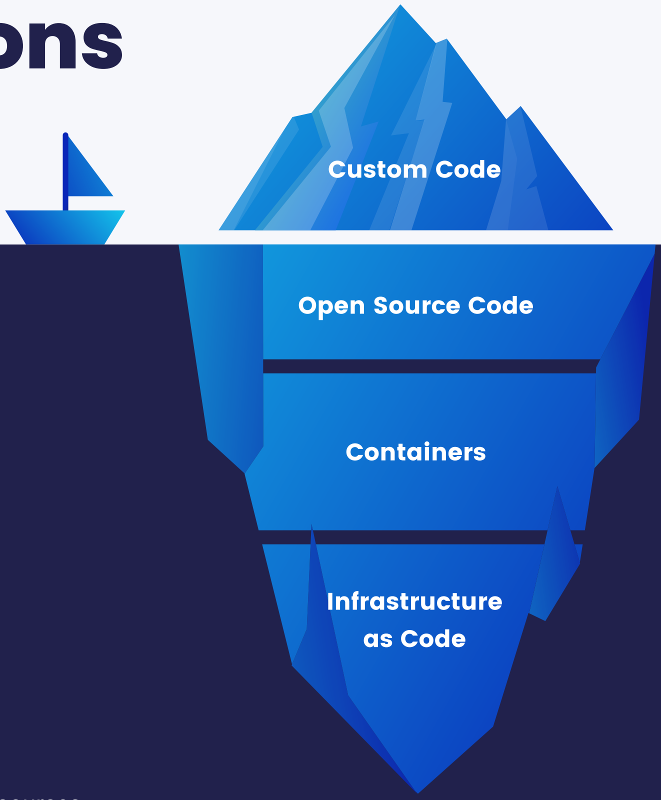
How to set sail on your voyage to secure your cloud applications



✓ Empower developers

Developer adoption requires a frictionless and intuitive solution to enable security without impacting pace.

✓ Automate fixes

The solution can't just report on what vulnerabilities exist. It must make it easy to fix the problems quickly.

✓ Be security deep

The solution must leverage complete, timely, and accurate vulnerability data and cannot rely solely on publicly available sources.

Destination DevSecWhat?

The tech industry is notorious for its overuse of technical jargon. Now, under normal circumstances, use of technical jargon is accepted despite the fact that it is often only understood by those that possess a technical background. However, what happens when even those with a technical background don't understand their own jargon or misunderstand it to the point of creating different interpretations for these terms?

Patrick Debois, who coined the term DevOps back in 2009 and is Snyk's Lab Researcher, combined *development* and *operations* in order to stress the importance of breaking down organizational silos and bringing various teams responsible for software development and deployment into one cohesive team that effectively collaborates towards a common goal. Since its inception that included business teams, developers, security & test engineers, operations, and others.

Today, we hear a lot about DevSecOps, a term that stresses the importance of *security* and its role in the software development lifecycle. Of course, this is really nothing new since it has been there all along just as much as quality assurance and testing. So, before we lose our minds and create another new and unnecessary term; *DevSecQAOps* or *DevQASecOps*, or something silly; let's take a step back and understand the problem we are trying to solve and how we can implement an effective strategy to solve it.

Charting the course

If you are not thinking about security from the start, then you have automatically exposed yourself and compromised the integrity of your application. This will result in teams reacting to incidents, further accumulation of technical debt and ultimately frustration. The outcome of this is avoidance behavior. That is, choosing to avoid dealing with what is perceived as difficult. The solution is to continuously integrate security into each step of the software development lifecycle SDLC and choose the right tools for each step.

So how do we course correct and where do we begin? First, this is a matter of *mindset*. The same tenets that drove organizational cultural transformations to *break down silos* and adopt DevOps as a way to support Agile methodologies apply to *DevSecOps* as well: Security is everyone's responsibility. Additionally, *visibility* is a critical part to the process as is making informed decisions which is not possible without having accurate data. This realization answers the question of where to begin: your source. **Empowering developers** and encouraging adoption requires a frictionless and intuitive solution that allows them to think about security early in the process without disruption.

Where do you keep your custom code, open source code, Dockerfile, IaC templates today? You likely keep these in your Source Code Management (SCM) system! For example, Bitbucket Cloud. Snyk offers native integration in Bitbucket Cloud, as well as Bitbucket Pipes and Jira, making your DevSecOps journey as smooth as possible.

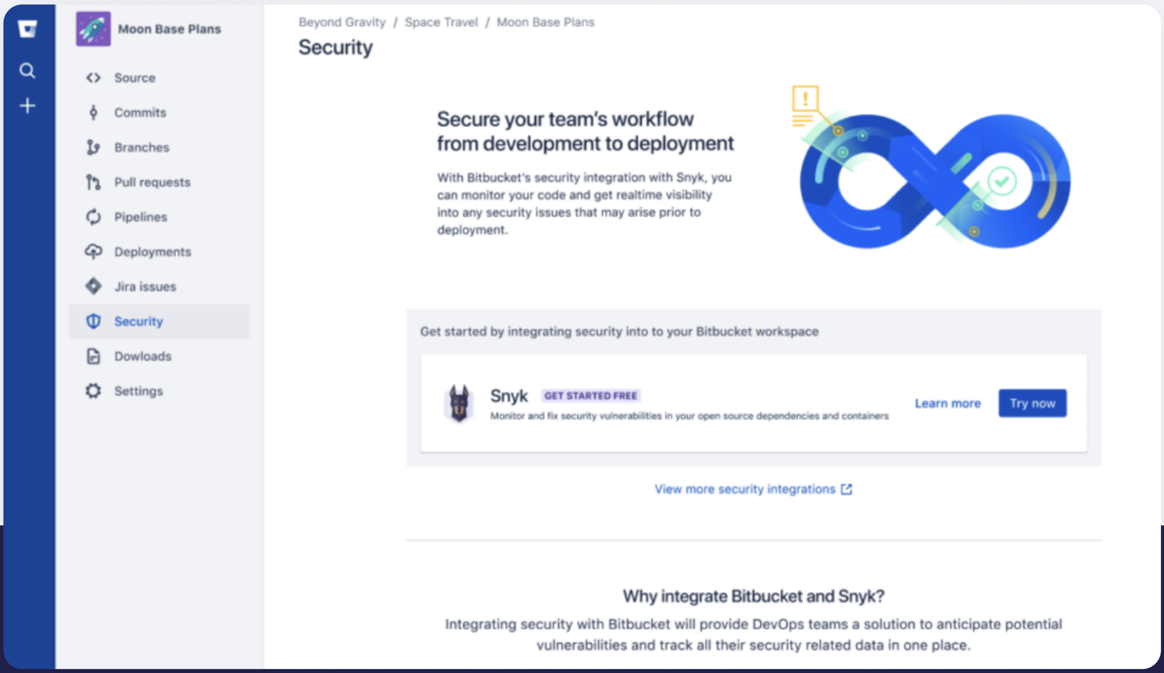
> Where do you keep your custom code, open source code, Dockerfile, and Infrastructure as Code templates today? You likely keep these in your Source Code Management (SCM) system! For example, Bitbucket Cloud. At Snyk, we are proud to offer seamless integrations into Bitbucket Cloud, as well as Bitbucket Pipes and Jira, making your DevSecOps journey as smooth as possible.

Bring 'em near

For Bitbucket Cloud users, when you set sail on your DevSecOps voyage, you can import, test, fix and monitor Bitbucket Cloud projects for open source vulnerabilities within your Bitbucket workflow.

Announcing the security tab in Bitbucket Cloud, powered by Snyk

From scans on Pull Requests, gating builds in CI/CD, continuous monitoring and alerting, and the coveted automated Fix Pull Requests, it's possible to efficiently embed security into your software Bitbucket development processes and workflow. The security tab inside Bitbucket is where you'll be able to start your journey to see risks that exist in your dependency files code base and container images, so you can resolve them before they are escalated by your security team.



Security

Secure your team's workflow from development to deployment

With Bitbucket's security integration with Snyk, you can monitor your code and get realtime visibility into any security issues that may arise prior to deployment.

Get started by integrating security into to your Bitbucket workspace

Snyk

Monitor and fix security vulnerabilities in your open source dependencies and containers

Why integrate Bitbucket and Snyk?

Integrating security with Bitbucket will provide DevOps teams a solution to anticipate potential vulnerabilities and track all their security related data in one place.

But let's take an even more conservative approach and focus just on visibility. What's in my code base and what does it mean? For this, prioritization as well as contextual information on open source vulnerabilities is incredibly useful. What options are available for Bitbucket users today to gain an understanding of risk in the context of real-world use cases? How can this help your organization prioritize vulnerabilities?

A common method used for prioritizing vulnerabilities today is based on the Common Vulnerability Scoring System (CVSS). One of the reasons this statement is true is that simply categorizing vulnerabilities as high severity is not indicative of the actual level of exposure. In contrast, a solution like Snyk also provides priority scoring powered by a unique algorithm that factors several key variables in addition to the CVSS score. For example, availability of fixes for known exploits, reachability, maturity, and more. This level of **security depth** allows you to intelligently implement an effective prioritization strategy based on accurate data.

In order to see and fix vulnerabilities in your code base, start by importing your Git repository as a project in Snyk. To help you do this as seamlessly as possible, Snyk integrates with Bitbucket Cloud and enables you to import your source code repositories as projects. Once a project is imported, Snyk monitors the source code for your repositories. Snyk tests the projects you've imported for any known security vulnerabilities found in the application's dependencies, testing at a frequency you control. However, this alone is not a complete solution. In order to be effective, the solution has to go beyond reporting on the existence of vulnerabilities. It must also make it easy to **fix** the issues quickly, which Snyk delivers.

Land ahoy!

Don't let the treacherous waters of misinformation steer you off course. The tools and processes are at your disposal to implement a holistic process for securing your *DevSecOps* pipelines when you turn to Snyk's seamless integration with Bitbucket Cloud, Bitbucket Pipes and Jira.

So, anchors aweigh: your captain and crew can now cart their course fast and stay secure. Or in developer terms, developer teams can now get real-time visibility into any security issues in their code and containers, identify vulnerability fixes early in development and monitor new risks post deployment.