

Professional Skills
Agile Fundamentals
Jira
Git
Databases Introduction
Java Beginner
Maven
Testing (Foundation)
Java Intermediate
HTML
CSS
Javascript
Spring Boot
Selenium
Sonarqube
Advanced Testing (Theory)
Cucumber
MongoDB
Express
NodeJS
React
Express-Testing
Networking
Security
<div><div></div>Security Basics</div>
<div><div></div>Authentication and Authorization</div>
<div><div></div>Security Attacks</div>
<div><div></div>Passwords</div>
<div><div></div>Hashing vs Encrypting vs Encoding</div>
<div><div></div>Introduction to DevSecOps</div>
<div><div></div>Pen Testing, DAST &amp; SAST</div>

## Introduction to DevSecOps

### Contents

- [Overview](#)
- [Methodology](#)
- [Shifting to the left](#)
- [Principles](#)
  - [Automated testing](#)
  - [Empower Teams](#)
  - [Be prepared for threats](#)
- [Tutorial](#)
- [Exercises](#)

### Overview

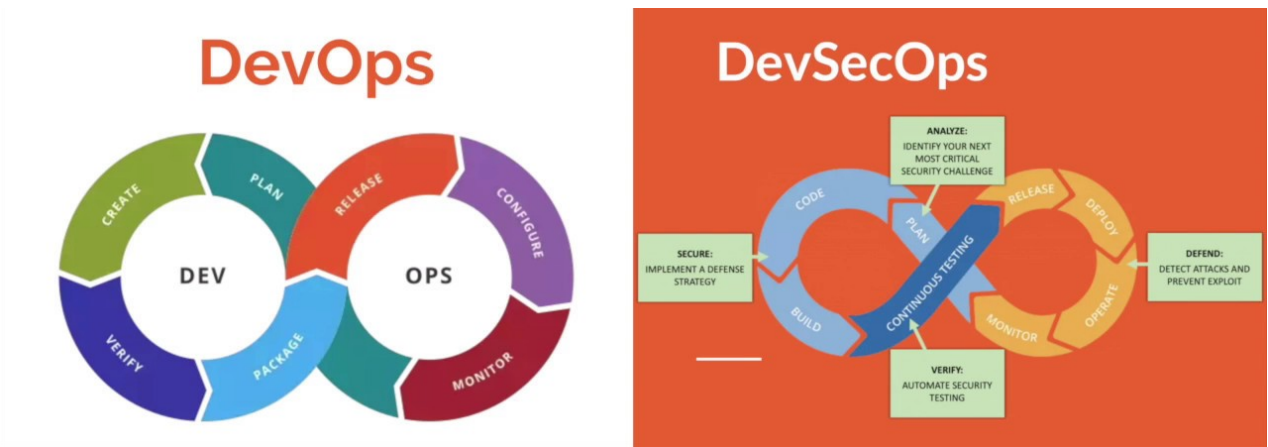
DevSecOps is a methodology which incorporates **Information Security** and **DevOps**, focusing on the security of an application throughout the development and deployment process.

### Methodology

Due to the speed at which DevOps teams operate and the focus on automation of processes, there has been a concern that they are unable to effectively handle security concerns, with DevSecOps the opposite is true.

The automation of quality control processes allows DevOps teams to deploy better quality applications, much quicker.  
The same principles are applied for security within DevSecOps teams.

Automating basic security processes not only allows for security checks earlier in the development cycle, but also shifts the security responsibility from a single individual to the entire team.




### Shifting to the left

Traditionally, security has been something which gets tested right at the end of the product creation, fitting well with a waterfall-like workflow.

The goal of DevSecOps is to have an entire team of engineers who all have a knowledge of basic security practices, and who all take responsibility for the security of a product throughout its life cycle.  
This implementation of security tests and practices earlier in the software development life cycle is beneficial, this is an example of **shift left testing**.

### Principles

#### Automated testing

 Introduction to OWASP
Cloud Fundamentals
AWS Foundations
AWS Intermediate
Linux
DevOps
Jenkins Introduction
Jenkins Pipeline
Markdown
IDE Cheatsheet

Wherever possible, DevSecOps should make use of automated security testing.

Automated tests are re-usable, highly scalable and consistent. Automated tests can be run many times throughout the development of an application, highlighting potential security risks early on. Furthermore, a test which is regularly run by a computer will be faster and more accurate than tests being run by an individual, allowing for comprehensive reports to be generated, showing how secure an application is.

### Empower Teams

Instead of having one or two security engineers who must test each application and fix security issues, DevSecOps encourages all members of a team to have a good knowledge of basic security practices. With the correct tools and basic training, members of the team are able to fix any concerns with the area of the project they are working on.

### Be prepared for threats

When teams left the testing and implementation of security measures to the end, it wasn't seen as something the DevOps team had to worry about, after all that's not their job.

DevSecOps encourages the whole team to think about potential security threats, and what would happen if the product was attacked. This constant mindfulness of threats leads to better designed applications which will be able to withstand attacks far better than ones which have security measures introduced right at the end, almost as an afterthought.

### Tutorial

There is no tutorial for this module.

### Exercises

There are no exercises for this module.