

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
Cloud Fundamentals
AWS Foundations
AWS Intermediate
Linux
DevOps

Tools

Contents

- [Overview](#)
- [Source code repository](#)
- [Build server](#)
- [Configuration management](#)
- [Virtual infrastructure](#)
- [Test automation](#)
- [Pipeline orchestration](#)
- [Tutorial](#)
- [Exercises](#)

Overview

There is a number of tools required for DevOps to be effective, this module will try to briefly introduce them.

Source code repository

Source code repository is a place where developers check in their code and make changes to it.
It also manages different versions of code that are being checked in, so that developers don't override each others work accidentally.

Source control has been around for roughly 40 years, it is also a crucial component for continuous integration.
Popular source control tools are:

- Git
- Subversion
- Cloudforce
- Bitbucket
- TFS

Build server

Build server also called Continuous integration server is a tool that compiles code from the source code repository into executable code base.
More popular tools are: Jenkins, Travis, Bamboo.

Configuration management

Configuration management is about configuring a server or an environment.
An example of this would be something like installing Git, Java etc.
More popular tools are: Puppet and Chef.

Virtual infrastructure

Google Cloud Platform, Amazon Web Services and Microsoft azure are examples of virtual infrastructure.
Vendors sell virtual infrastructure or platform-as-a-service (PaaS) services.
These services allow tools like Chef or Puppet to create and manage new machines through an API.

Automation tools together with virtual infrastructure allow companies to do software development in the DevOps way removing the need to do it manually.

<div><div></div> DevOps as a Culture</div> <div><div></div> Continuous Integration</div> <div><div></div> Tools</div> <div><div></div> CD in the Enterprise</div>
Jenkins Introduction
Jenkins Pipeline
Markdown
IDE Cheatsheet

New code can be tested out by simply automatically sending it to the cloud infrastructure, it then gets deployed onto an environment that was automatically configured and lastly automated tests can be then executed. All of the mentioned would be achieved without any intervention from a human.

Test automation

Test automation isn't a new thing as it has been around for a while now. A rise in popularity has been seen with the increase of DevOps adaptation for it. The main thing it's used for is to make sure within your pipeline that the new code is ready for deployment. To make sure that you're doing continuous integration right, you would need to ensure that automated tests are executed without any human intervention, and that they will tell you whether your build is safe to deploy.

More popular tools are: Selenium

Pipeline orchestration

If you imagined a factory where an item is build through multiple assembly lines going through them one at a time, and the item is gradually built, pipeline is a similar concept. The code goes from developer hands through multiple stages until it get's deployed to a production environment. These multiple stages would check things like: whether the code compiles, tests pass, the environment it will get deployed to won't be broken because of it etc.

Tutorial

There is no tutorial for this module.

Exercises

There is no exercise for this module.