

<> Code

Issues 5

Pull requests 5

Actions

Security

Insights



DevOps Roadmap for 2024. with learning resources

[newsletter.techworld-with-milan.com/](#)

Apache-2.0 license

10.5k stars 1.7k forks 178 watching 1 Branch 0 Tags Activity

Public repository

1 Branch

0 Tags

Go to file

t

Go to file

+

Add file

Code

milanm

Merge pull request #26 from mohamadalzhori/networks\_video\_resource

b0e884e · 3 weeks ago

.gitignore	PDF improvements	2 years ago
DevOps Roadmap No Backg...	Update for 2024	6 months ago
DevOps Roadmap.pdf	Update for 2024	6 months ago
DevOps Roadmap.png	Update for 2024	6 months ago
DevOpsBooks.jpg	Add DevOps books	2 years ago
DevOpsBurger.jpg	Add DevOps as a Burger (DaaB)	5 months ago
LICENSE	Add LICENCE	2 years ago
README.md	Merge pull request #26 from moha...	3 weeks ago
devops tools.png	Add new DevOps tools	last year
pdfversion.png	Update for 2024	6 months ago

README

Apache-2.0 license

# DevOps Roadmap 2024.

This is a step-by-step guide on how to become a DevOps engineer, with links to relevant learning resources.

If you want to learn more about DevOps, be sure to subscribe to [my newsletter](#).

## Disclaimer

---

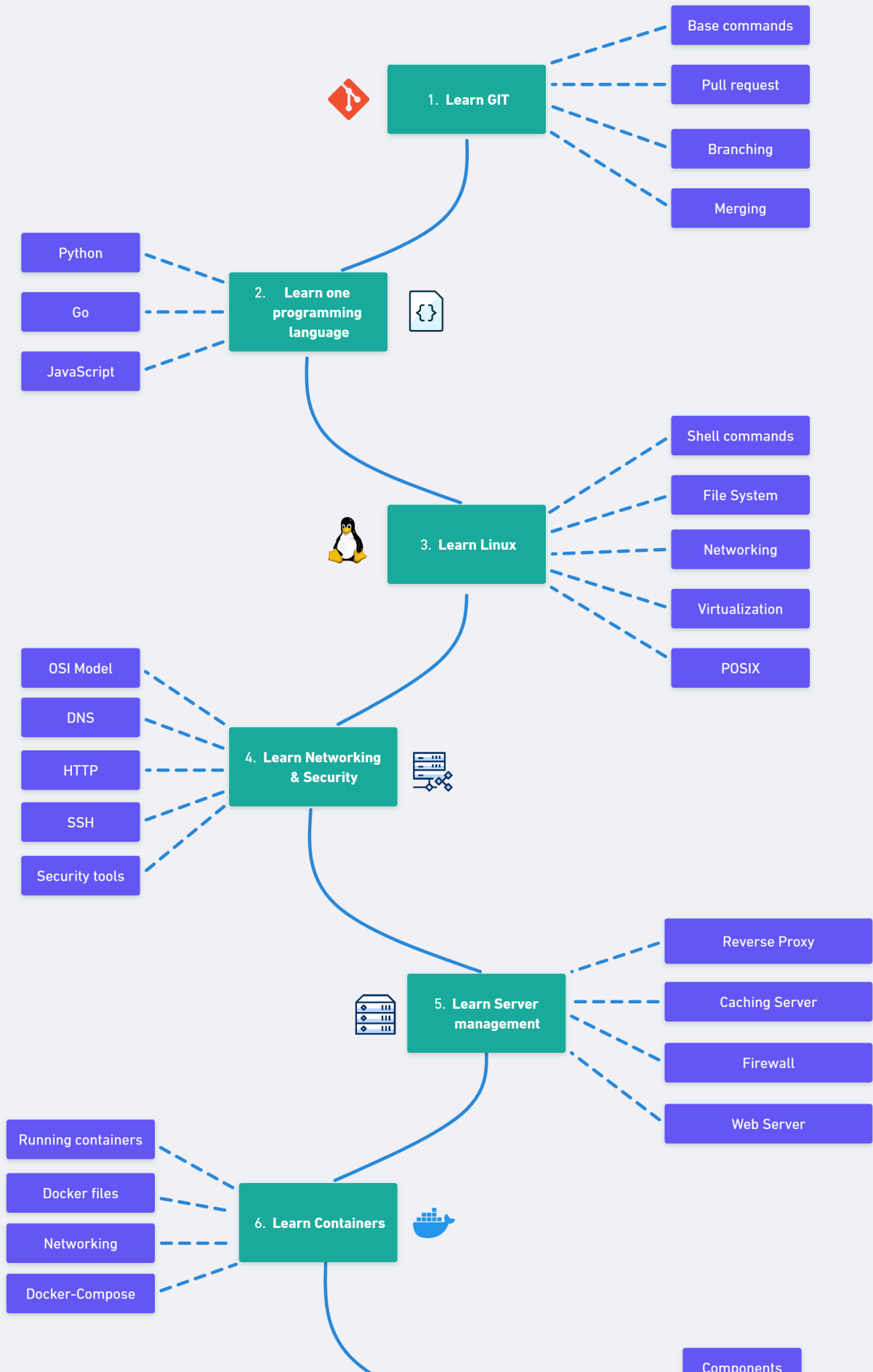
The purpose of this roadmap is to give you an idea about the landscape. The road map will guide you if you are confused about what to learn next, rather than encouraging you to pick what is hype and trendy. You should grow some understanding of why one tool would be better suited for some cases than the other and remember that hype and trendy do not always mean best suited for the job.

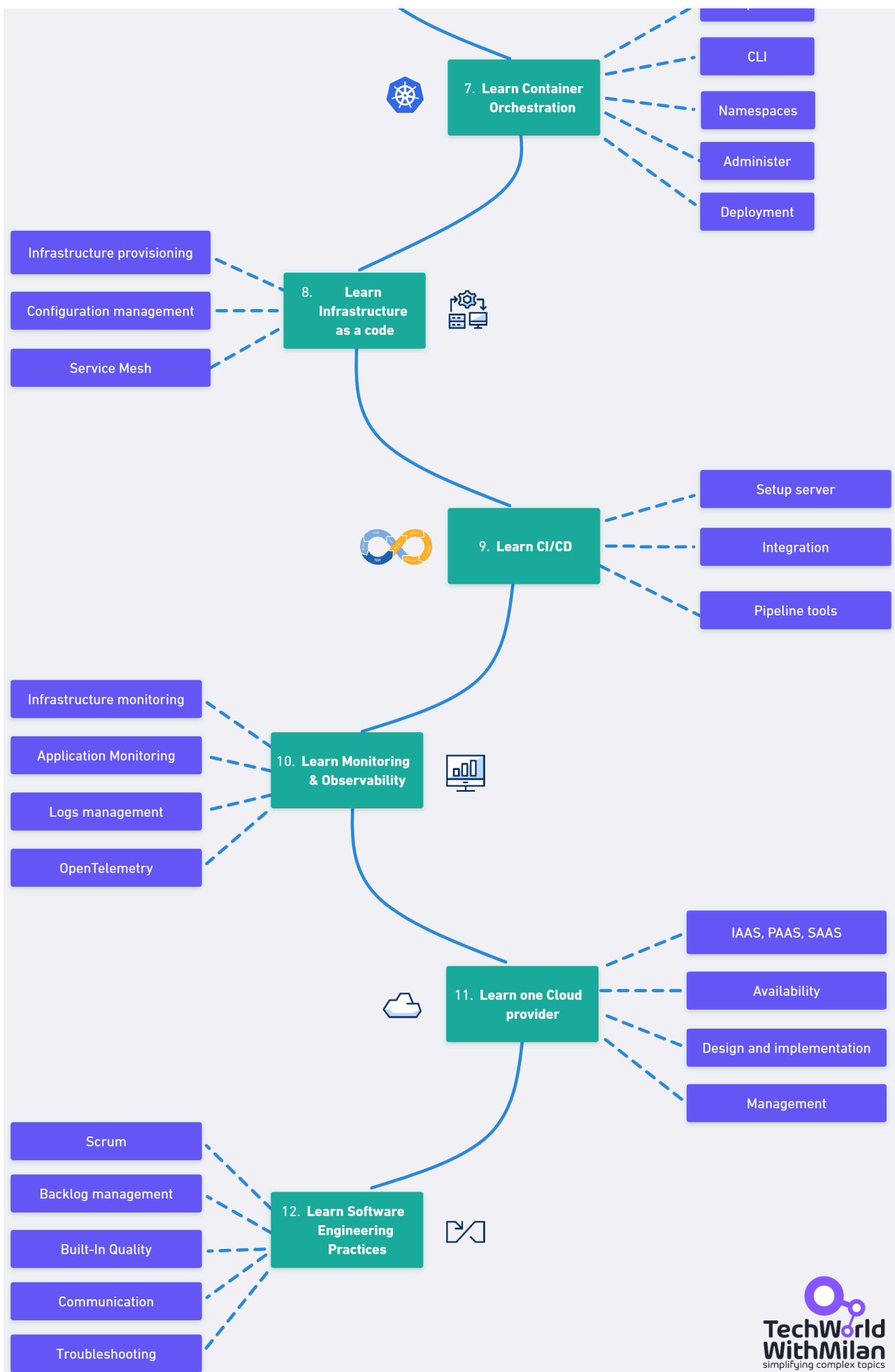
## Give a Star!

---

If you like or are using this project to learn or start your solution, please give it a star. Thanks!

# DevOps Roadmap 2024.







Download [PDF version](#).

## Table of Contents

---

- [Learning resources for DevOps Engineers \(mostly free\)](#)
  - [1. GIT](#)
  - [2. Learn one programming language](#)
  - [3. Learn Linux & Scripting](#)
  - [4. Learn Networking & Security](#)
  - [5. Learn Server Management](#)
  - [6. Learn Containers](#)
  - [7. Learn Container Orchestration](#)
  - [8. Learn Infrastructure as a code](#)
  - [9. Learn CI/CD](#)
  - [10. Learn Monitoring & Observability](#)
  - [11. Learn one Cloud provider](#)
  - [12. Learn Software Engineering Practices](#)
- [Additional resources](#)
  - [Tools](#)
  - [Books](#)

## Learning resources for DevOps Engineers (mostly free)

---

### 1. GIT

All your resources (files) will be held in a GIT repository. Those files are **application code** but also **infrastructure as a code**.

**Git** is a free tool used for source code management. Git is used to tracking changes in the source code, enabling multiple developers to work together on non-linear development.

The two most popular Git platforms are **GitLab** and **GitHub**.

Here you need to learn Git commands, like git clone, branch, merge, and how to collaborate on a project with pull requests.

Resources:

- [Pro Git Book](#) <sup>FREE</sup>
- [Learn Git by Atlassian](#) <sup>FREE</sup>
- [Learn Git Branching](#) <sup>FREE</sup>
- [Learn Git & GitHub on CodeAcademy](#) <sup>FREE</sup>
- [Git Command Explorer](#) <sup>FREE</sup>
- [Git Immersion](#) <sup>FREE</sup>
- [A Visual Git Reference](#) <sup>FREE</sup>

## 2. Learn one programming language

As an engineer, it is recommended to know at least one programming language that you can use to write **automation scripts**.

Some popular programming languages for DevOps-es are **Python, Go, and JavaScript**.

Python is a multi-paradigm language. Being an interpreted language, code is executed as soon as it is written, and the syntax allows for writing code in different ways. **Python** is frequently recommended as the first language new coders should learn, because of its focus on readability, consistency, and ease of use.

Here you need to learn basic concepts of programming languages, such as syntax, if/else, loops, data structures, etc.

Resources:

- [Automate the Boring Stuff with Python book](#) <sup>FREE</sup>
- [Python Crash Course](#) <sup>FREE</sup>
- [The Modern JavaScript Tutorial](#) <sup>FREE</sup>
- [JavaScript Crash Course For Beginners](#) <sup>FREE</sup>
- [Eloquent JavaScript, 3rd edition](#), Marjin Haverbeke <sup>FREE book</sup>
- [Go by Example](#) <sup>FREE</sup>

## 3. Learn Linux & Scripting

An Operating system serves as a bridge between a computer's user and its hardware. Its function is to offer a setting in which a user can conveniently and effectively run programs.

As most servers use **Linux OS**, you need to make yourself comfortable with Linux and its CLI.

One easy-to-start distribution is **Ubuntu**.

In addition, you need to know **scripting** to automate tasks for development and operations.

Here you can learn OS-specific languages, such as **Bash or Powershell**, or independent, like Python or Go.

Resources:

- [Operating System - Overview](#) <sup>FREE</sup>

- [Shell Scripting Tutorial](#) <sup>FREE</sup>
- [Powershell Tutorial for Beginners: Learn Powershell Scripting](#) <sup>FREE</sup>
- [Bash Reference Manual](#) <sup>FREE</sup>
- [Ultimate Guide: Getting Started With Ubuntu](#) <sup>FREE</sup>
- [FreeBSD Handbook](#) <sup>FREE</sup>
- [Linux command handbook](#) <sup>FREE</sup>

## 4. Learn Networking & Security

A **network protocol** is an established set of rules that determine how data is transmitted between different devices in the same network. Essentially, it allows connected devices to communicate with each other, regardless of any differences in their internal processes, structure, or design.

Here you will need to know how a network works, how to configure **firewalls**, understand how **DNS** works, **OSI model**, IP addresses, ports, etc.

Resources:

- [OSI Model Explained](#) <sup>FREE</sup>
- [Computer Networking: A Top-Down Approach](#) <sup>book</sup> [Video Content](#) <sup>video</sup>
- [TCP/IP and Networking Fundamentals for IT Pros](#) <sup>Pluralsight course</sup>
- [DevSecOps : Master Securing CI/CD | DevOps Pipeline](#) <sup>Udemy course</sup>
- [Hands-On Security in DevOps: Ensure continuous security, deployment, and delivery with DevSecOps](#) <sup>Book</sup>
- [Securing DevOps: Security in the Cloud](#) <sup>Book</sup>

## 5. Learn Server Management

Server management includes all the infrastructure monitoring and maintenance required for servers to operate reliably and at optimal performance levels. The primary goals of an **effective server management strategy** are to:

- Minimize server slowdowns and downtime while maximizing reliability.
- Build secure server environments.
- Scale servers and related operations to meet the needs of the organization over time.

Here you will need to know what is **forward and reverse proxies**, **caching servers**, and how to operate **Web Servers**, such as Nginx, Apache, or IIS.

Resources:

- [What is a reverse proxy?](#) <sup>FREE</sup>
- [Cache Server](#) <sup>FREE</sup>
- [Reverse Proxy vs. Forward Proxy: The Differences](#) <sup>FREE</sup>
- [What is load balancing?](#) <sup>FREE</sup>
- [What is a Firewall?](#) <sup>FREE</sup>
- [The NGINX Handbook](#) <sup>FREE</sup>
- [Learn Apache Server](#) <sup>FREE</sup>
- [Learn IIS](#) <sup>FREE</sup>

## 6. Learn Containers

A **container** is a standard unit of software that packages up code and all its dependencies, so the application runs quickly and reliably from one computing environment to another.

**Docker** is by far the most popular container technology today. A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries, and settings. Here you need to know how to run containers, Docker Networking, Volumes, Dockerfiles, and run multiple containers with Docker-Compose.

Resources:

- [What are Containers?](#) <sup>FREE</sup>
- [Learning Containers From The Bottom Up](#) <sup>FREE</sup>
- [Docker Tutorial for Beginners by TechWorld with Nana](#) <sup>FREE</sup>
- [Docker Mastery: with Kubernetes +Swarm from a Docker Captain](#) <sup>Udemy course</sup>
- [What is Service Mesh?](#) <sup>FREE</sup>

## 7. Learn Container Orchestration

Container orchestration **automates** the deployment, management, scaling, and networking of containers.

Container orchestration can be used in any environment where you use containers. It can help you to deploy the same application across different environments without needing to redesign it. And microservices in containers make it easier to orchestrate services, including storage, networking, and security.

Here you need to learn how **Kubernetes** works, and how to administer the Kubernetes cluster and deploy applications on it.

Resources:

- [Kubernetes Crash Course for Absolute Beginners by TechWorld with Nana](#) <sup>FREE</sup>
- [Primer: How Kubernetes Came to Be, What It Is, and Why You Should Care](#) <sup>Article</sup>
- [Certified Kubernetes Administrator \(CKA\) with Practice Tests](#) <sup>Udemy course</sup>
- [Learn Kubernetes - Beginners to Advanced by KodeKloud](#) <sup>Course</sup>
- [Understand when to use Cluster Services, Ingresses or API Gateways](#) <sup>FREE</sup>
- [Understand which Problems Service Mesh solve \(Use an Abstraction smi-spec.io\)](#) <sup>FREE</sup>
- [Learn how to automate TLS and DNS](#) <sup>FREE</sup>
- [Kubernetes Up and Running](#) <sup>Book</sup>
- [Kubernetes Learning Path - 50 days from zero to hero from Microsoft](#) <sup>FREE</sup>

## 8. Learn Infrastructure as a code

Sometimes referred to as **IaC**, it refers to the techniques and tools used to define infrastructure, typically in a markup language like YAML or JSON. Infrastructure as code allows Engineers to automate environment setup and teardown. Accelerates and de-risks deployment by provisioning gold copy environments on demand.

**Terraform** is the most popular infrastructure provisioning tool, but there are others such as Ansible, Chef, Puppet, and Vagrant.

Here you need to know how to do **infrastructure provisioning** and **configuration management**.

Resources:

- [GUIs, CLI, APIs: Learn Basic Terms of Infrastructure-as-Code](#) <sup>FREE</sup>
- [Official Terraform Tutorials](#) <sup>FREE</sup>



- [A Comprehensive Guide to Terraform](#) <sup>FREE</sup>
- [Automate Terraform documentation like a pro!](#) <sup>FREE</sup>
- [Writing reusable Terraform modules](#) <sup>FREE</sup>
- [What is Istio?](#) <sup>FREE</sup>
- [Puppet overview](#) <sup>FREE</sup>
- [Puppet Courses](#) <sup>FREE and PAID</sup>
- [Getting Started With Ansible](#) <sup>FREE</sup>
- [Learning Ansible Basics](#)
- [Get started with Red Hat Ansible](#) <sup>FREE and PAID</sup>
- [Mastering Ansible](#) <sup>Udemy Course</sup>
- [Learn Chef](#) <sup>FREE</sup>

## 9. Learn CI/CD

Continuous Integration / Continuous Deployment (CI/CD) is a method to frequently deliver apps to customers by introducing **automation** into the stages of app development. CI/CD is a solution to the problems integrating new code can cause for development and operations teams.

CI/CD introduces continuous automation and **continuous** monitoring throughout the lifecycle of apps, from integration and testing phases to delivery and deployment. These connected practices are often referred to as a "**CI/CD pipeline**" and are supported by development and operations teams.

There are **different stages** of a CI/CD pipeline, such as: **build, test and deploy**, but there could be much more activities included:

- Checking code from version control and building it
- Having staged gates for different kinds of approvals
- Managing environment variables
- Restarting services
- Executing tests
- And more...

Here you need to learn how to set up CI/CD server, integrate code and trigger pipelines automatically, and build and package management tools.

Some **popular CI/CD tools** are Jenkins, TeamCity, CircleCI, Bamboo, GitLab, and Azure DevOps.

Resources:

- [Continuous Integration](#) <sup>FREE</sup>
- [CI/CD Pipeline: A Gentle Introduction](#) <sup>FREE</sup>
- [Learn GitLab with tutorials](#) <sup>FREE</sup>
- [Get started with GitLab CI/CD](#) <sup>FREE</sup>
- [Jenkins, From Zero To Hero: Become a DevOps Jenkins Master](#) <sup>Udemy course</sup>
- [Learn GitHub actions](#) <sup>FREE</sup>
- [Workflow syntax for GitHub Actions](#) <sup>FREE</sup>
- [Learn Azure DevOps](#) <sup>FREE</sup>
- [GitLab Cheatsheets](#) <sup>FREE</sup>

## 10. Learn Monitoring & Observability

Monitoring entails overseeing the entire development process from planning, development, integration and testing, deployment, and operations. It involves a complete and **real-time view** of the status of applications, services, and infrastructure in the production environment.

This is especially important when our software is in **production**, and we need to track all kinds of issues in our infrastructure and application.

The two most popular tools are **Prometheus** and **Grafana**.

Here you need to know how to set up monitoring and visualize data.

Resources:

- [What Is Observability? Comprehensive Beginners Guide](#) <sup>FREE</sup>
- [The Hows, Whys and Whats of Monitoring Microservices](#) <sup>FREE</sup>
- [DevOps Monitoring](#) <sup>FREE</sup>
- [Applying Basic vs. Advanced Monitoring Techniques](#) <sup>FREE</sup>
- [Learn Prometheus](#) <sup>FREE</sup>
- [Learn Grafana](#) <sup>FREE</sup>
- [Elastic Stack](#) <sup>FREE</sup>

## 11. Learn one Cloud provider

Cloud providers provide a layer of APIs to abstract infrastructure and provision it based on security and billing boundaries. The cloud runs on servers in data centers, but the abstractions cleverly give the appearance of interacting with a single "platform" or large application. The ability to quickly provision, configure and secure resources with cloud providers has been key to both the tremendous success, and complexity, of modern DevOps.

The most popular cloud providers in the market are **AWS** and **Azure**, as well as **Google Cloud**.

Here you need to know how to manage users and administration, networks, virtual servers, etc.

Resources:

- [Exam AZ-900: Microsoft Azure Fundamentals](#) <sup>FREE</sup>
- [Microsoft Azure Fundamentals Certification Course \(AZ-900\)](#) <sup>FREE</sup>
- [AZ-900 | Microsoft Azure Fundamentals Full Course, Free Practice Tests, Website and Study Guides](#) <sup>FREE</sup>
- [Ultimate AWS Certified Cloud Practitioner - 2022](#) <sup>Udemy</sup>
- [AWS Developer by A Cloud Guru](#) <sup>Learning path</sup>

## 12. Learn Software Engineering Practices

As a DevOps engineer, you will probably work in a team with other developers in an Agile world, such as **Scrum**. So, it is very important to know different parts of **SDLC**, as well as the tools which are used there.

In addition, it would be good to know how **automation testing** is working, as you will need to set up it in CI/CD way.

Here you need to know what is **Scrum**, all phases of **SDLC**, how **automation testing** works, etc.

Resources:

- [What is Scrum?](#) <sup>FREE</sup>
- [Ways To Learn About Scrum](#) <sup>FREE</sup>

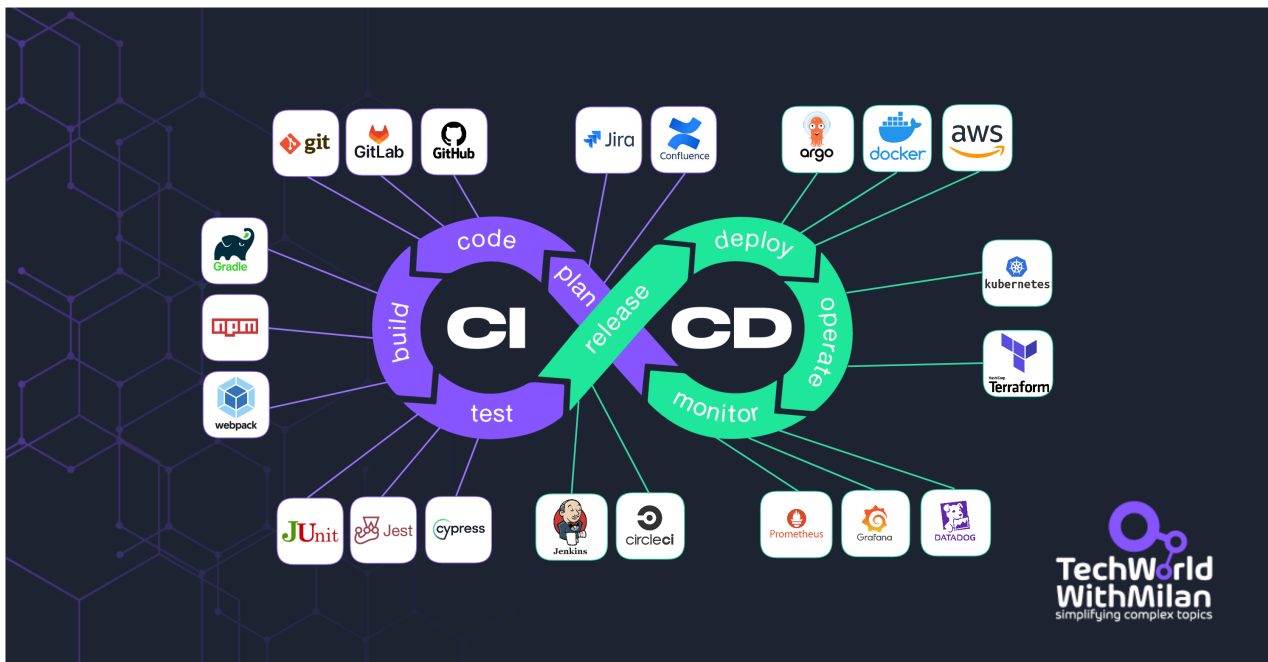
- [Software Development Life Cycle \(SDLC\) Phases & Models](#) <sup>FREE</sup>
- [The Beginner's Guide to Agile in Jira: Course description](#) <sup>FREE</sup>
- [Learn SAFe](#) <sup>FREE</sup>
- [Learn Automation Testing](#) <sup>FREE</sup>
- [GitLab - Beginner's Guide to DevOps](#) <sup>FREE</sup>
- [Common SDLC Models](#) <sup>FREE</sup>

## Additional resources

---

### Tools

- **Work Tracking:** [Asana](#), [Monday](#), [Jira](#), [Trello](#), [Azure Boards](#).
- **Source code control:** [Git](#), [Github](#), [GitLab](#), [BitBucket](#), [Azure DevOps](#).
- **CI/CD:** [Jenkins](#), [Team City](#), [Github Actions](#), [Travis CI](#), [Bamboo](#), [Circle CI](#), [Azure Pipelines](#), [Octopus Deploy](#), [Harness](#), [CloudBees CodeShip](#).
- **Source Code Analysis:** [SonarQube](#), [Veracode](#).
- **Artifact management:** [Artifactory](#), [Docker Container Register](#), [npm](#), [Yarn](#), [NuGet](#).
- **Configuration Management:** [Terraform](#), [Ansible](#), [Puppet](#), [Chef](#).
- **Container orchestration:** [Docker](#), [Kubernetes](#), [Red Hat OpenShift](#).
- **Monitoring:** [Prometheus](#), [Grafana](#), [Splunk](#), [Dynatrace](#), [Kibana](#).



### Books

- [The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations](#), Gene Kim, Patrick Debois, John Willis, Jez Humble

The book introduces product development, quality assurance, IT operations, and information security. It is a great read for those who are new to DevOps or who want to learn more about how the various components of DevOps work together.

- [Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations](#), Nicole Forsgren, Jez Humble, Gene Kim

This book presents both the findings and the science behind measuring software delivery performance. For promoting DevOps to senior management, it's a fantastic tool.

- **Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation**, Jez Humble, David Farley

It introduces automated architecture management and data migration. Many of the deployment pipeline concepts that have since become standard were established in this book. Config as Code, build and deployment automation, and efficient testing techniques are covered in some technical detail. Its mostly technical book.

- **Team Topologies: Organizing Business and Technology Teams for Fast Flow**, Matthew Skelton, Manuel Pais

The book talks about how to organize teams in a way that enables fast flow of value to customers. It provides a set of four fundamental team topologies: Stream-Aligned Teams, Enabling Teams, Complicated-Subsystem Teams, and Platform Teams, which can be combined and adapted to suit different organizational contexts.

- **Effective DevOps: Building a Culture of Collaboration, Affinity, and Tooling at Scale**, Jennifer Davis, Ryn Daniels

The book provides effective ways to improve team coordination. It shows how to break down information silos, monitor relationships, and repair misunderstandings that arise between and within teams in your organization.

- **The Phoenix Project: A Novel about IT, DevOps, and Helping Your Business Win**, Gene Kim, Kevin Behr, George Spafford

It is a classic novel about effectiveness and communications. IT work is like manufacturing plant work, and a system must be established to streamline the workflow. One of the best books out there.

- **Site Reliability Engineering**, Betsy Beyer, Chris Jones, Jennifer Petoff, Niall Richard Murphy

This book explains the whole life cycle of Google's development, deployment, and monitoring, and how to manage the world's biggest software systems (also known as SRE). Anyone who considers themselves to be more on the "Ops" end of the DevOps side or who wants to know how to strengthen the bonds between various Dev and Ops teams should read SRE.



# DevOps as a Burger (DaaB)

We can even present this roadmap as a burger :).



Contributors 8

