

Book Cover

# Table of Contents

<b>Book Layout</b>	<b>3</b>
<b>Section 1: Theory</b>	<b>6</b>
I am a SysAdmin	7
I am a Developer	12
I am almost a DevOps	14
DevOps 101	15
DevOps 102	16
DevOps 201	17
DevOps 202	18
DevOps 203	19
DevOps 204	20
DevOps 205	21
DevOps 205	22
DevOps 205	23

# Book Layout

## About The Book

The DevOps cookbook is a free open-source book that aims to provide general knowledge about DevOps, and a collection of recipes for common DevOps practices. The book is written in Markdown and is published using [Ibis](#).

## Book Sections

Chapters are organized in a way that you can read them in order, or you can jump to the chapter you need. The book is divided into 4 sections.

## Section 1: Theory

### Section 1.1: Must-Have Theory

- **I am a SysAdmin!:** This section is about the general knowledge a DevOps engineer should have from system administration perspective. It covers topics like System Administration basics, the Linux command line, and the basics of networking.
- **I am a Developer!:** This section is about the general knowledge a DevOps engineer should have from development perspective. It covers topics like the basics of programming, the basics of databases, and the basics of web development.
- **I am *almost* a DevOps!:** This section is a checkpoint. Take your time and grab a cup of tea, you are now ready to become a DevOps engineer.

### Section 1.2: DevOps Basics

- **DevOps 101:** This section covers topics like what is DevOps, the DevOps culture, and the DevOps tools.
- **DevOps 102:** This section covers topics like the DevOps lifecycle, the DevOps mindset, and the DevOps principles.

### Section 1.3: DevOps Advanced

- **DevOps 201:** This section covers topics like continuous integration, continuous delivery, and continuous deployment.
- **DevOps 202:** This section covers topics like infrastructure as code, and configuration

management.

- **DevOps 203:** This section covers topics like monitoring, and logging.
- **DevOps 204:** This section covers topics like security, and disaster recovery.
- **DevOps 205:** This section covers topics like containers, orchestration, and microservices.
- **DevOps 206:** This section covers topics like serverless, and cloud computing.
- **DevOps 207:** This section covers topics like DevOps in the enterprise.

## Section 2: Practice

### Section 2.1: Linux

- **Linux 2: Install LEMP Stack**
- **Linux 2: Install Laravel**
- **Linux 3: Automate Laravel Deployment With Bash**
- **Linux 4: Load Balance Laravel With HAProxy**
- **Linux 5: Migrate a Laravel App from one server to another**
- **Linux 6: Write a Bash Script to Monitor System Resources (MEM, CPU, STORAGE) and send a Discord Notification**

### Section 2.2: Databases

- **Database 1: Replicate a MySQL Database to a Slave Server**

### Section 2.3: Containers

- **Container 1: Deploy a Laravel App to a Docker Container**
- **Container 1: Deploy a Laravel App with Docker Compose**
- **Container 2: Deploy a Laravel App to a Kubernetes Cluster**

### Section 2.4: Cloud

- **Cloud 1: Deploy a Laravel App to a DigitalOcean Droplet**
- **Cloud 2: Configure a DigitalOcean Managed Database for your Laravel App**
- **Cloud 3: Configure a DigitalOcean space for your Laravel static assets**
- **Cloud 4: Deploy a Laravel App to a DigitalOcean Kubernetes Cluster**

### Section 2.5: Infrastructure as Code

- **IaC 1: Deploy a DigitalOcean Droplet using Terraform**
- **IaC 2: Deploy a DigitalOcean Droplet, Kubernetes Cluster, and a managed Database with Terraform**
- **IaC 3: Deploy a Laravel Application on a DigitalOcean Droplet with a managed Database with Terraform and Ansible**

## How to Contribute

The book is open-source and you can contribute to it by sending a pull request to the [GitHub repository](#)

# Section 1: Theory

# **I am a SysAdmin**

Since you want to become a DevOps engineer, you should have a good understanding of system administration. This section is about the general knowledge a DevOps engineer should have from system administration perspective. It covers topics like System Administration basics, the Linux command line, and the basics of networking.

## **What is System Administration?**

System administration is the process of managing computer systems and networks. It includes maintaining the hardware, software, and configurations of computer systems. Most companies distribute the work of system administration among a multiple teams of administrators. These teams usually include an SRE(site reliability engineer), a network administrator, a database administrator, a storage administrator, and a security administrator.

System administrators are responsible for the security, availability, and performance of computer systems. They also make sure that the systems are up-to-date and running smoothly. System administrators are also responsible for:

- Installing and configuring new hardware and software.
- Monitoring the performance of the systems.
- Troubleshooting problems.
- Performing backups and disaster recovery.
- Managing user accounts and permissions.
- Managing the network.
- MANAGING EVERYTHING!

## **What knowledge do you need from the System Administration world?**

You should have a good understanding of the following topics:

- Linux command line.
- Networking.
- Cloud.
- Databases.
- Security.
- Monitoring.
- Backups and disaster recovery.

Let's take a look at each of these topics.

# Linux Command Line

## What is the Linux command line?

The Linux command line is a text interface for your computer. It's a program that takes in commands, which it passes on to the computer's operating system to run. The command line is also known as the terminal, console, or shell.

The Linux command line is a powerful tool that allows you to do things that are difficult or impossible to do with a graphical user interface. It's also a great way to learn about the inner workings of your computer. It is a must-have skill for any DevOps and System engineer.

## How to learn the Linux command line?

There are many resources available online to learn the Linux command line. Here are some of the best resources:

- [Linux Command Line Basics](#)

# Networking

## What is Networking?

Networking is the practice of connecting two or more computing devices together for the purpose of sharing data. In a data network, computing devices exchange data with each other using connections (data links) between nodes. These data links are established over cable media such as wires or optic cables, or wireless media such as WiFi.

## How to learn Networking?

You should have a good understanding of the following topics:

- OSI Model.
- IP addresses.
- Subnets.
- Ports.
- Protocols.
- DNS.
- Load balancing.
- Firewalls.
- VPNs.

There are many resources available online to learn Networking. Here are some of the best resources:



- [OSI Model](#)
- [IP addresses](#)
- [Subnets](#)
- [Ports](#)
- [Protocols](#)
- [DNS](#)
- [Load balancing](#)
- [Firewalls](#)

## Cloud

### What is Cloud?

Cloud computing is the on-demand delivery of IT resources and applications via the Internet with pay-as-you-go pricing. Instead of buying, owning, and maintaining physical data centers and servers, you can access technology services, such as computing power, storage, and databases, on an as-needed basis from a cloud provider like DigitalOcean, Amazon Web Services (AWS), Microsoft Azure, or Google Cloud Platform.

### How to learn Cloud?

You should have a good understanding of the following topics (actually there are a lot more but let's start with these absolutely mandatory ones before we jump into DevOps):

- Cloud computing.
- Virtualization.

There are many resources available online to learn Cloud. Here are some of the best resources:

- [Cloud computing](#)
- [Virtualization](#)

## Databases

### What is a Database?

A database is an organized collection of data. It is the place where you store all your data. Databases are used to store and retrieve related information. They are used to manage large amounts of data efficiently.

### How to learn Databases?

You should have a good understanding of the following topics:

- Relational databases.
- Non-relational databases.
- SQL.
- NoSQL.

There are many resources available online to learn Databases. Here are some of the best resources:

- [Relational databases](#)
- [Non-relational databases](#)
- [SQL](#)
- [NoSQL](#)

## Security

### What is Security?

Security is the process of protecting information by preventing unauthorized access, use, disclosure, disruption, modification, inspection, recording, or destruction. It is a must-have skill for any DevOps and System engineer.

### How to learn Security?

You should have a good understanding of the following topics:

- Encryption.
- Authentication.
- Authorization.
- Access control.
- Vulnerabilities.
- Security best practices.

There are many resources available online to learn Security. Here are some of the best resources:

- [Encryption](#)
- [Authentication](#)
- [Authorization](#)
- [Access control](#)
- [Vulnerabilities](#)
- [Security best practices](#)

# Monitoring

## What is Monitoring?

Monitoring is the process of observing the performance of a system over time. It is a must-have skill for any DevOps and System engineer.

## How to learn Monitoring?

You should have a good understanding of the following topics:

- Monitoring tools.
- Monitoring best practices.

There are many resources available online to learn Monitoring. Here are some of the best resources:

- [Monitoring Tools: Grafana](#)
- [Monitoring Tools: Prometheus](#)
- [Monitoring Tools: Nagios](#)
- [Monitoring Tools: Zabbix](#)
- [Monitoring Tools: PagerDuty](#)

# **I am a Developer**

Since you want to become a DevOps engineer, you should have a good understanding of software development. This section is about the general knowledge a DevOps engineer should have from software development perspective.

## **What is software development?**

Need Information

## **What knowledge do you need from the Development world?**

You should have a good understanding of the following topics:

- Topic 1
- Topic 2
- Topic 3

### **Topic 1**

Need Information

#### **Sub-Topic 1**

Need Information

### **Topic 2**

Need Information

#### **Sub-Topic 2**

Need Information

### **Topic 3**

Need Information

### **Sub-Topic 3**

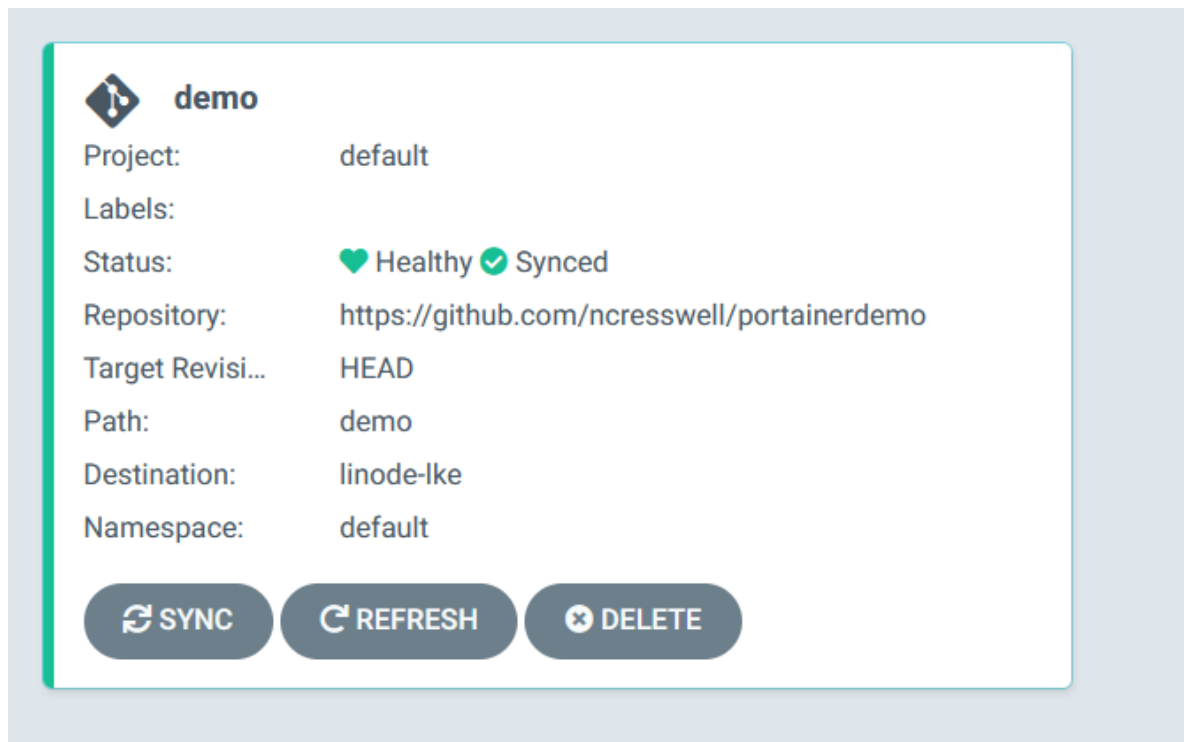
Need Information

etc.

## I am almost a DevOps

This is a checkpoint. Take your time and grab a cup of tea, you are now ready to become a DevOps engineer!

Here's an image of a healthy ArgoCD cluster for your future self to enjoy:



## **DevOps 101**

Text

## **What is DevOps**

Text

## **DevOps Culture**

Text

## **DevOps Tools**

Text

## **DevOps 102**

Text

## **DevOps Lifecycle?**

Text

## **DevOps Mindset**

Text

## **DevOps Principles**

Text



# **DevOps 201**

continuous integration, continuous delivery, and continuous deployment.

## **Continuous Integration**

Text

## **Continuous Delivery**

Text

## **Continuous Deployment**

Text

## **DevOps 202**

infrastructure as code, and configuration management

### **Infrastructure as Code**

Text

### **Configuration Management**

Text

## **DevOps 203**

monitoring, and logging

### **Monitoring**

Text

### **Logging**

Text

## **DevOps 204**

security, and disaster recovery

### **Security**

Text

### **Disaster Recovery**

Text

# **DevOps 205**

containers, orchestration, and microservices

## **Containers**

Text

## **Orchestration**

Text

## **Microservices**

Text

## **DevOps 205**

serverless, and cloud computing

### **Serverless**

Text

### **Cloud Computing**

Text

## **DevOps 205**

DevOps in the enterprise

### **DevOps in the Enterprise**

Text