Go Cloud-Native with DevOps Best Practices

Foundation Stack / Minimum skill set / Be Project oriented.
6 months plan

Get Clarity of what DEVOPS is all about

So What is DevOps?

DevOps is an organizational and cultural movement that aims to increase software delivery velocity, improve service reliability, and build shared ownership among software stakeholders.

DevOps Research & Assessment (DORA)

DevOps is a methodology in which teams own the entire process from application development to production operations, hence DevOps. It goes beyond implementing a set of technologies and requires a complete shift in culture and processes. DevOps calls for groups of engineers that work on small components (versus an entire feature), decreasing handoffs – a common source of errors.

CLOUD NATIVE GLOSSARY

Software delivery consists of all of the work you need to do to make the code available to a customer, such as running that code on production servers, making the code resilient to outages and traffic spikes, and protecting the code from attackers.

Credit: Yevgeniy Brikman - Terraform Up and Running.

DevOps Engineer

You basically write code to make sure that all the code the backend and frontend engineers write is always working properly & never crashes..

You are responsible for managing infrastructure, deploying code, configuring servers, scaling clusters, backing up data, monitoring apps, etc.

CNCF Cloud Native Definition v1.0

Cloud native technologies empower organizations to build and run scalable applications in modern, dynamic environments such as public, private, and hybrid clouds. Containers, service meshes, microservices, immutable infrastructure, and declarative APIs exemplify this approach.

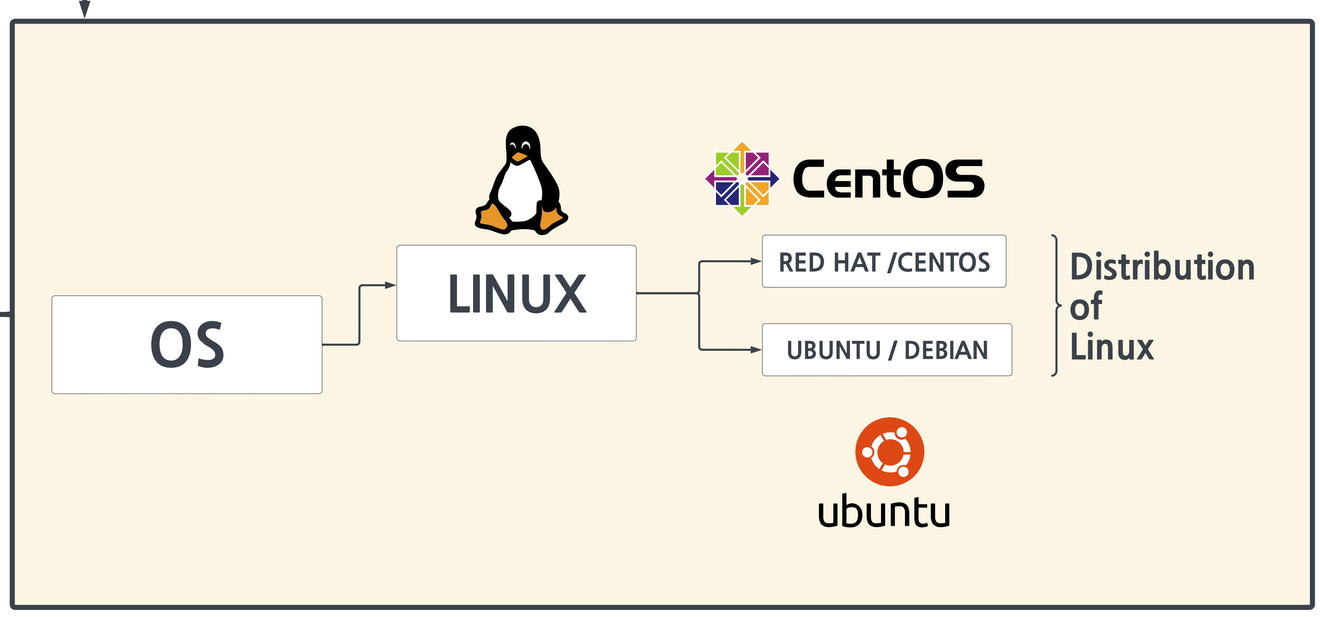
These techniques enable loosely coupled systems that are resilient, manageable, and observable. Combined with robust automation, they allow engineers to make high-impact changes frequently and predictably with minimal toil.
From: `https://github.com/cncf/toc/blob/master/DEFINITION.md`

CLOUD NATIVE COMPUTING FOUNDATION

A microservices architecture is an architectural approach that breaks applications into individual independent (micro)services, with each service focused on a specific functionality. These services work together closely, appearing to the end user as a single entity. Take Netflix as an example. Its interface allows you to access, search, and preview videos. These capabilities are likely powered by smaller services that each handle one functionality, e.g., authentication, search, and running previews in your browser.

This architectural approach allows developers to push out new features or update functionality much faster than if they were all tightly coupled, such as in a **monolithic application** (more to that below).

CLOUD NATIVE GLOSSARY



How to start & stop services

Basic troubleshooting

How logging works

Networking in Linux

User / file management

Text Editor (VIM)

Systemd

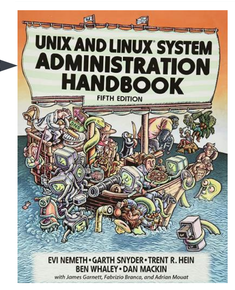
namespace

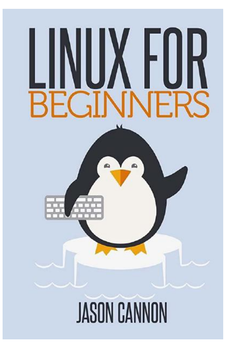
cgroups

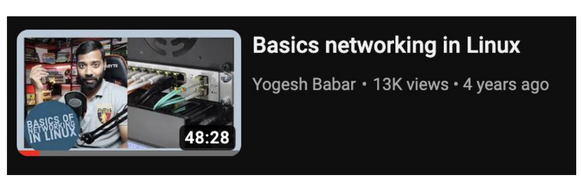
Command Line

mkdir
ls
cd
touch
vi
cat
cp
mv
chmod
systemctl
firewall-cmd
journalctl
ps
tail
netstat
export
curl

Resources:



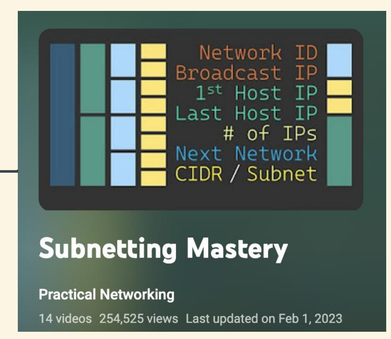


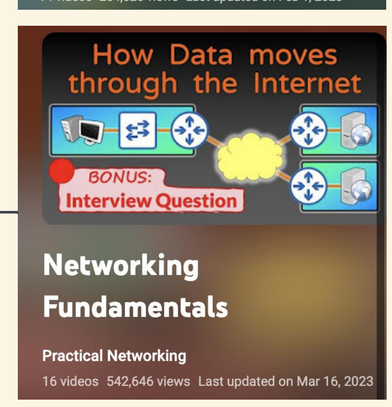


NETWORKING

- IP ADDRESS & TYPES
- IPV4 / IPV6
- DHCP
- DNS
- PING / TRACEROUTE
- SWITCHING & ROUTING
- OSI MODEL
- SUBNETTING

Resources

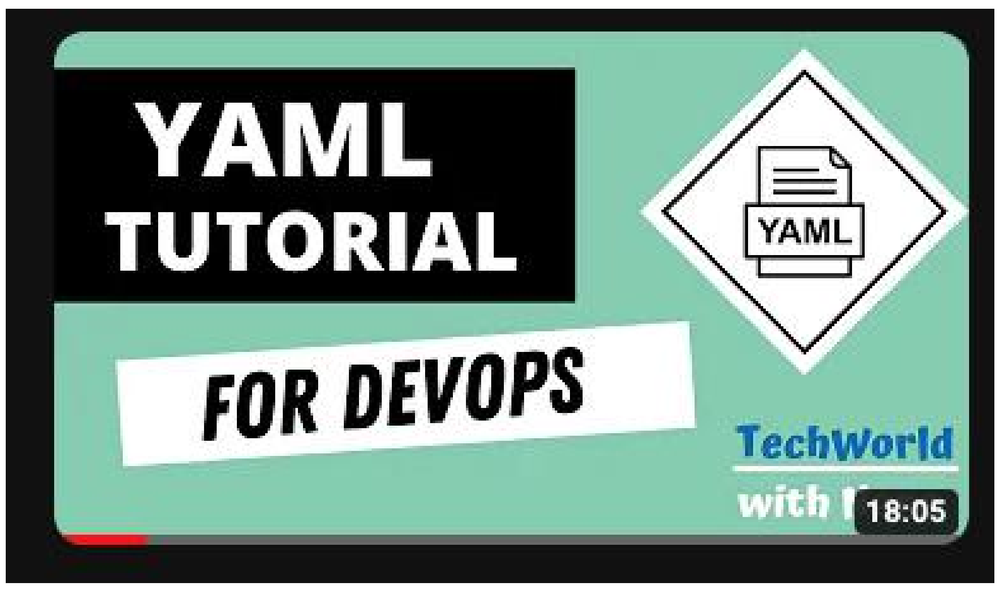


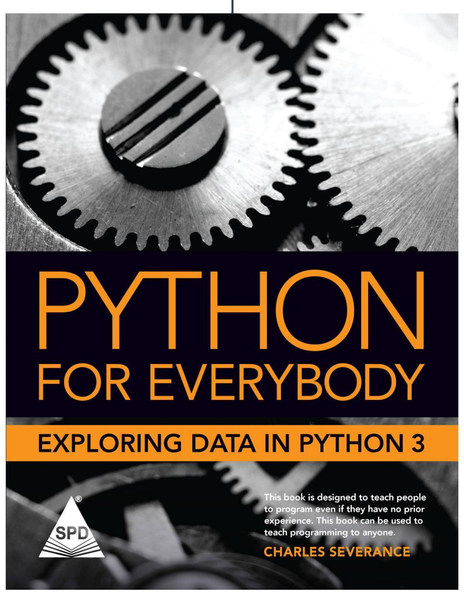


FOUNDATIONAL SERVICES

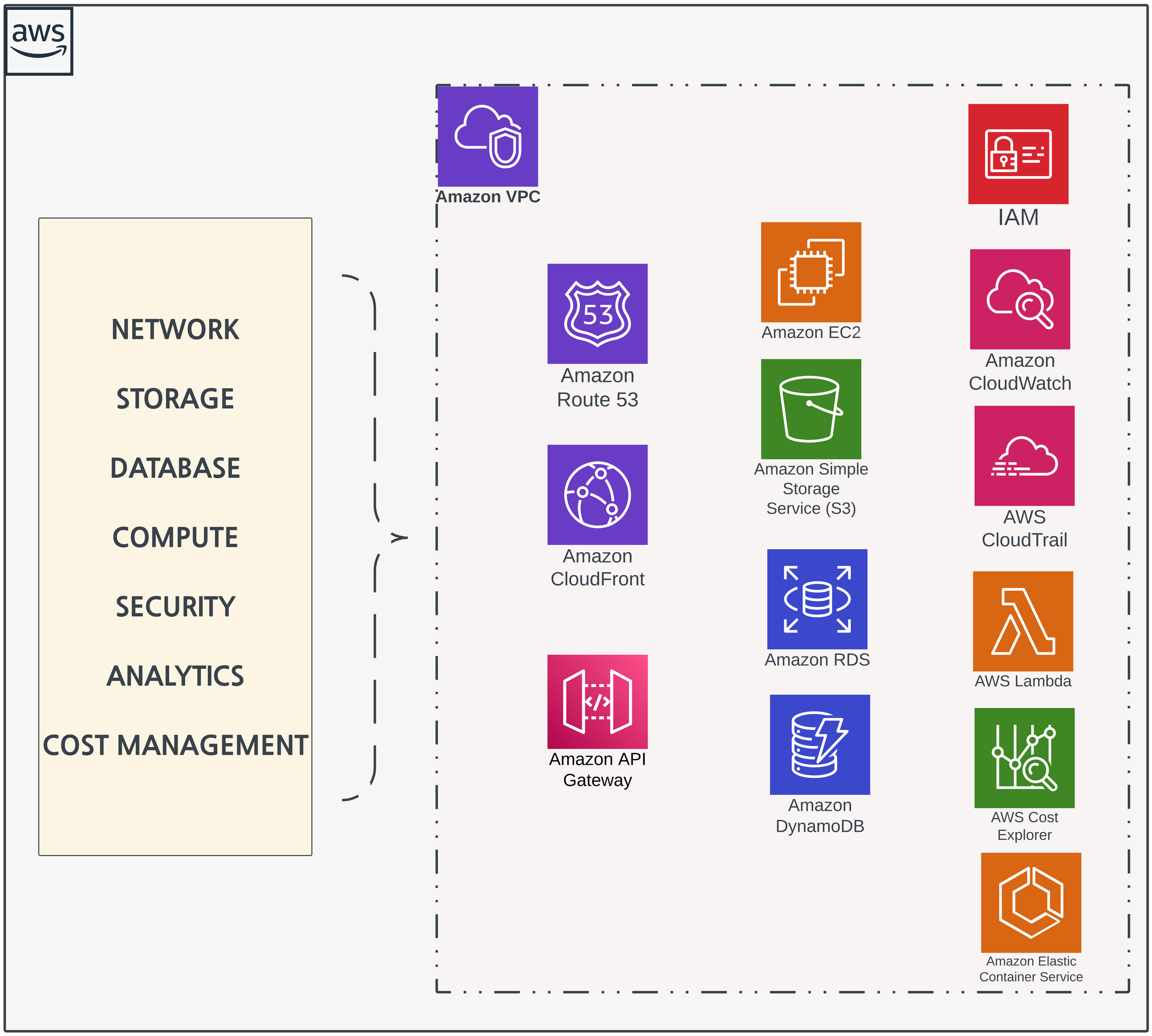
- SHELL SCRIPTING / PYTHON / YAML
- GIT / GITHUB
- Create Repos
- Commit and Push Code
- Branching
- Public / Private repo
- Pull, Rebase, Fork, Clone
- Git branching strategy:
- Trunk-based vs. Git Flow

Resources





CLOUD



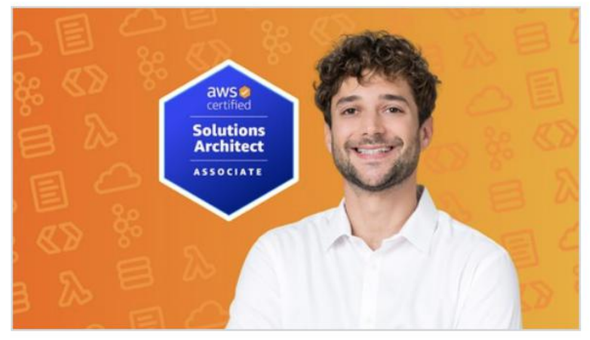
Resources

livelessons

Networking in Amazon Web Services (AWS)

Richard A. Jones

video



Ultimate AWS Certified Solutions Architect Associate SAA-C03

Stephane Maarek | AWS Certified Cloud…

4.7 ★★★★½ (174,953)

27 total hours · 388 lectures · All Levels

DevOps Core Values:

Culture

Automation

Measurement

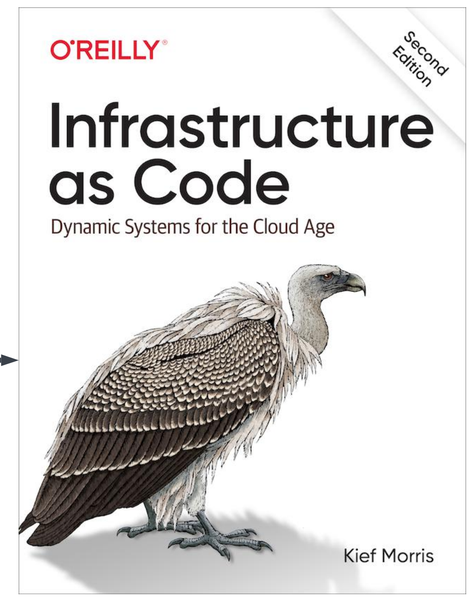
Sharing

Refer to the cloud-native landscape for more..

l.cncf.io

v20200501

In order to achieve faster and efficient delivery of software, it's important to automate as much of the software delivery process as possible, this translates to building your infrastructure as code (IaC)



4 broad Categories of IAC tools:

Server templating tools

Orchestration tools

Provisioning tools

CI/CD tools implemented as pipeline-as-code (PaC)

SERVER TEMPLATING TOOLS

VIRTUALIZATION

docker

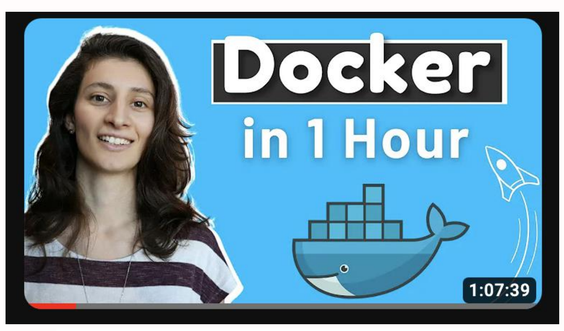
CONTAINERS

containerd

vmware

VIRTUAL MACHINE (VM) (VMWARE, EC2)

HashiCorp Packer



Docker Mastery: with Kubernetes +Swarm from a Docker Captain

Bret Fisher, Docker Captain Program

4.7 ★★★★½ (58,462)

21.5 total hours · 209 lectures · All Levels

ORCHESTRATION TOOLS

Kubernetes

sc — Storage Class

vol — Volume

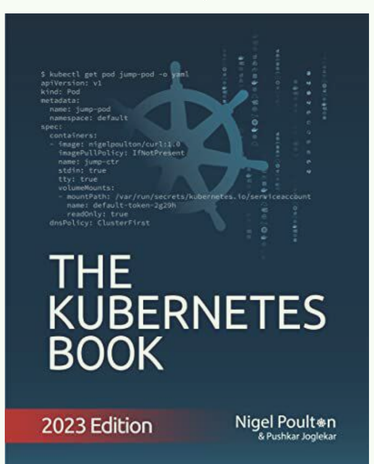
ep — Endpoint

pv — Persistent Volume

pvc — Persistent VolumeClaim

ing — Ingress

HELM



CKA Certification Course – Certified Kubernetes Administrator

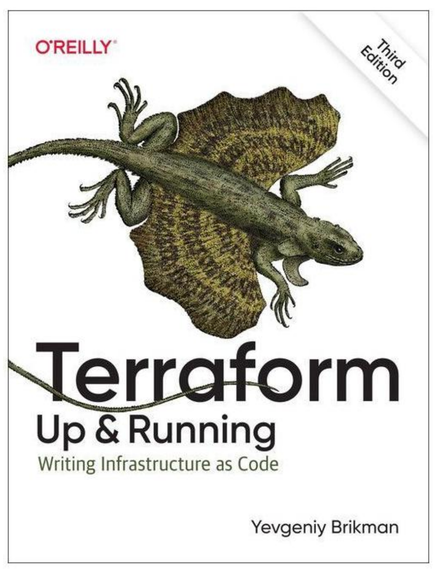
Prepare for the Kubernetes Administrators Certification (CKA) with live practice tests right in your browser

4.8 ★★★★½ HOW STUDENTS RATE KODEKLOUD 500000+ STUDENTS

Taught by: Mumshad Mannambeth - Founder of KodeKloud , an IT Consultant and a Certified Kubernetes Administrator

PROVISIONING TOOLS

Terraform

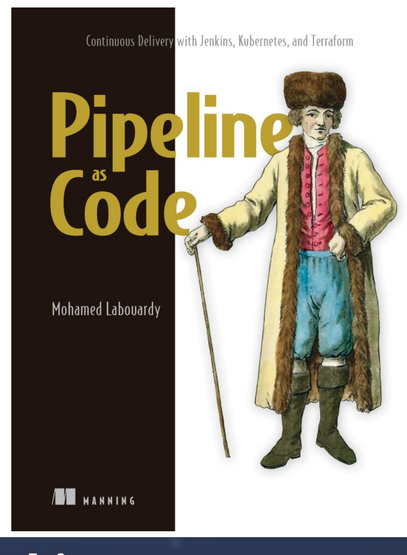


CI/CD TOOLS

Continuous integration, often abbreviated as CI, is the practice of integrating code changes as regularly as possible. CI is a prerequisite for continuous delivery (CD). Traditionally, the CI process begins when code changes are committed to a source control system (Git, Mercurial, or Subversion) and ends with a tested artifact ready to be consumed by a CD system.

Continuous delivery, often abbreviated as CD, is a set of practices in which code changes are automatically deployed into an acceptance environment (or, in the case of continuous deployment, into production). CD crucially includes procedures to ensure that software is adequately tested before deployment and provides a way to rollback changes if deemed necessary. Continuous integration (CI) is the first step towards continuous delivery (i.e., changes have to merge cleanly before being tested and deployed).

Jenkins argo



Jenkins

Get started with the most popular CI/CD tool

4.8 ★★★★½ HOW STUDENTS RATE KODEKLOUD 1000000+ STUDENTS

Taught by: Michael Levan, DevOps Researcher, Advisor, & Consultant

Adopt continous learning principles. Read Documentations...You won't die!

@_AS_CODE

GITHUB.COM/OSEMIDUH

GitHub