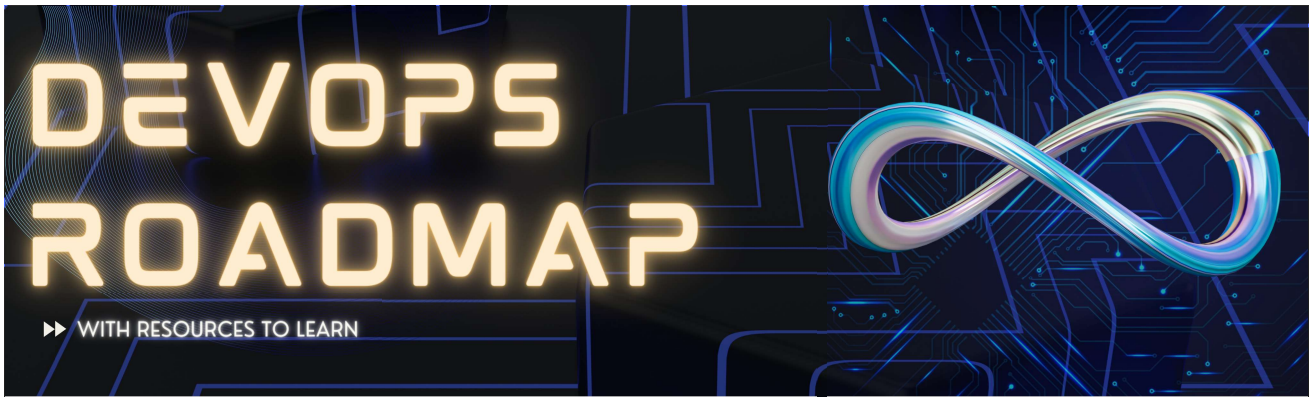
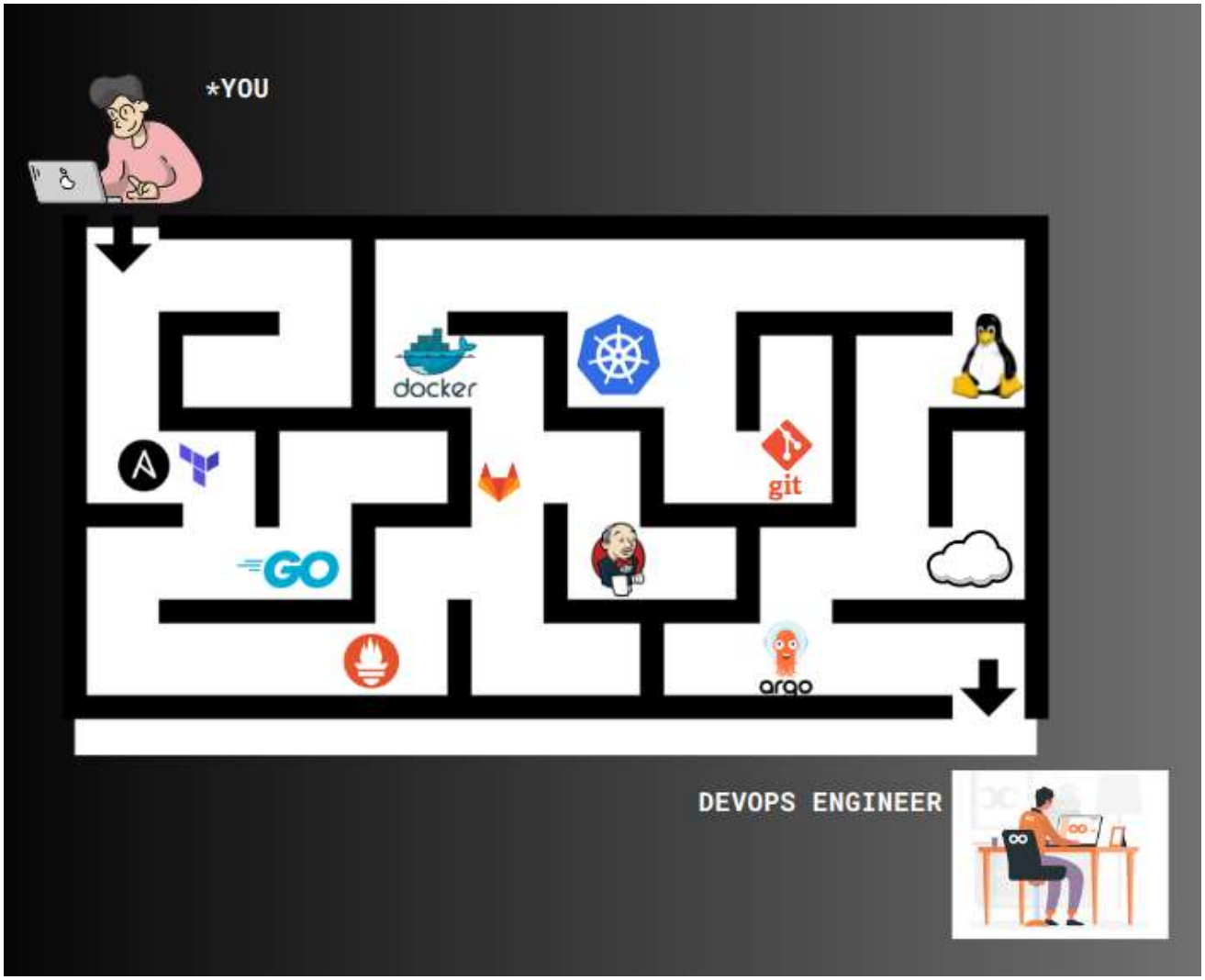


This roadmap is created by [Cloud Champ](), to support please subscribe the Youtube channel 🙏

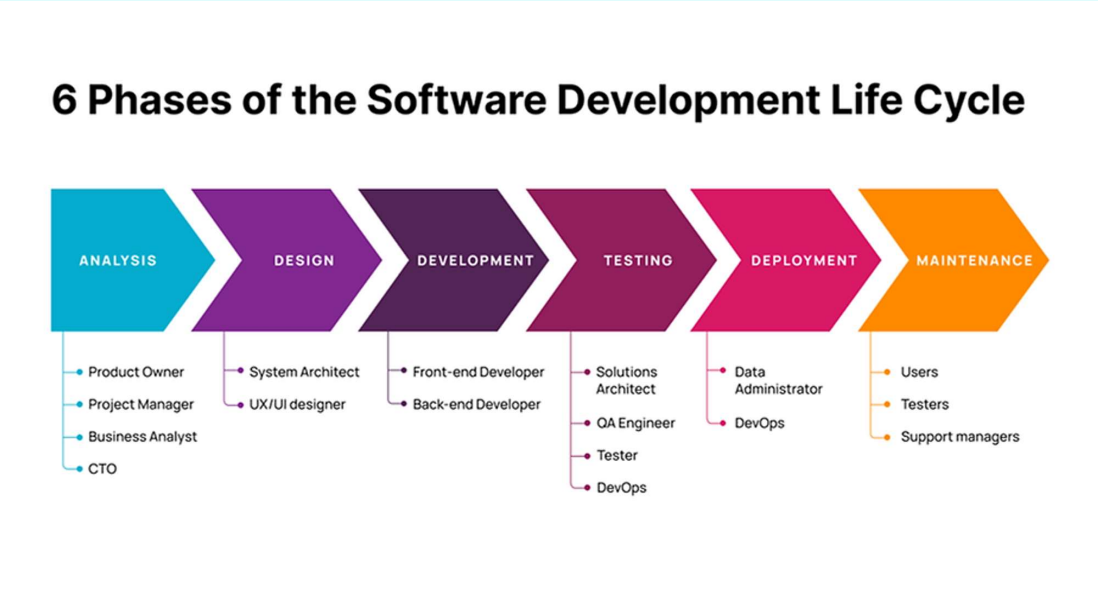
Everything you need to know to become a Great DevOps Engineer!!



1. Basics of SDLC

Understand **how is a software created** and where do “DevOps” fall in Software Development Life Cycle.

SDLC, or Software Development Life Cycle, is the step-by-step process that software goes through, from planning and coding to testing and deployment.



Resources to learn:

SDLC (Blog)

What is SDLC? (Youtube Video)

https://youtu.be/Gkp8wLZAtpY

❗ You don't have to get too detailed; just a grasp of software deployment processes and DevOps basics is sufficient. However, **hands-on experience** with the below technologies 👇 is more valuable and beneficial.

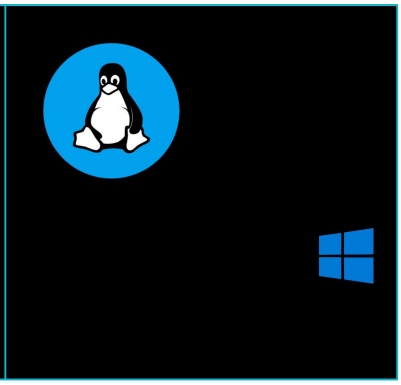
2. Operating System (Linux🐧)

As a DevOps engineer, you manage the servers where applications run.

Linux is the go-to choice for servers, and Most DevOps tools are Linux-based; so knowing linux commands helps a lot in server management and handling Devops Tasks.

▢ **What to learn:**

- Shell Commands
- SSH for remote access
- Virtualization
- Text Editors for File Editing (vim, nano)
- File System Permissions
- Package Management (apt, yum)
- Process and Service Management (ps, kill)



✏️ Things you can do to practice Linux:

- **Web Server setup**
- **Database setup**
- **User and Group Management**
- **Install Different Devops tools (docker, jenkins, terraform etc)**

> Popular Linux Distributions: **Ubuntu**, **CentOS**, **Debian**, Arch linux and more.
>
> Best way to learn is to have Linux as OS for your laptop, but you can also spin up servers on Cloud, WSL or virtualbox.

Resources to learn:

Linux Cheatsheet (Free)

Linux course on Udemy (Paid)

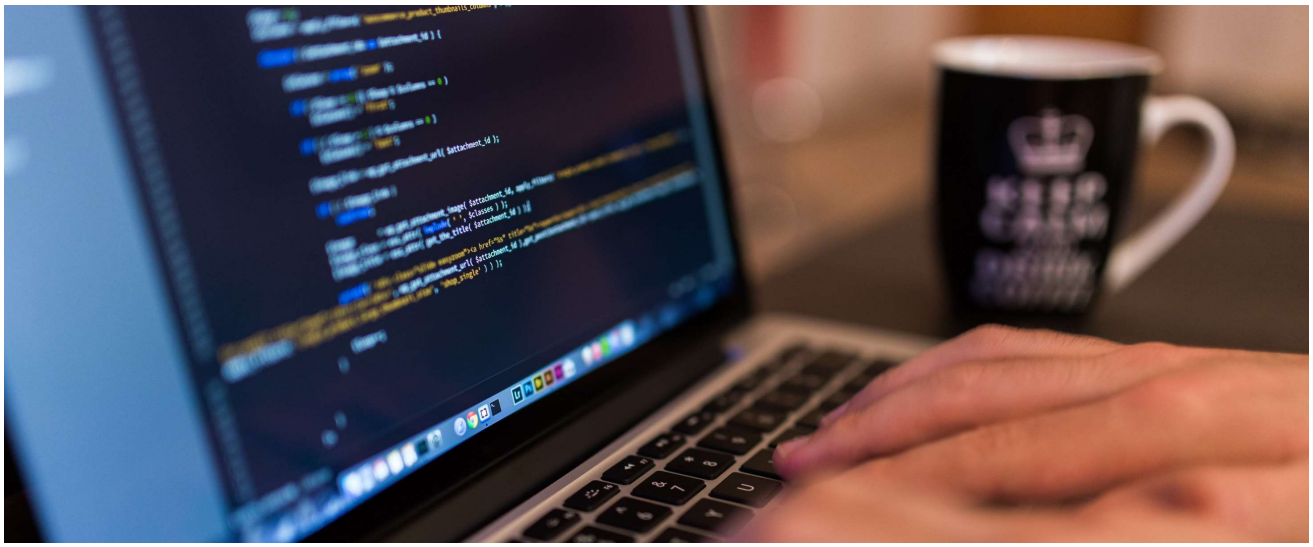
Linux course on Youtube (Free)

Linux Project (Free)

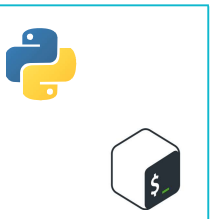
Linux Commands for DevOps Engineer! (Free)

3. Scripting/Programming

After you learn the basics of Linux, use your knowledge to create scripts.



DevOps Engineers automate tasks to streamline software delivery and enhance efficiency. Scripting is typically carried out using OS-based languages such as BASH and Powershell, or standalone languages like Python or Golang.



✏️ How is scripting used in DevOps

- Backup Automation
- Working with APIs
- Integration with other devops tools
- System Cleanup Automation
- Email Notification Script
- Software Update Automation
- Log Analysis Automation
- Network Diagnostics Script
- Data Transformation Automation

- User Account Management Script

If you have a question - Do devops engineer require coding?
The answer is yes, DevOps engineer needs coding knowledge, not necessarily for application development, but to utilize tools, SDKs, and scripting for seamless integration and automation in the DevOps workflow.

Resources to learn:

Bash scripting Course on Youtube (Free)

Automate the boring stuff with Python (Udemy Course)

How much coding is required:
https://lnkd.in/efPrD4qT

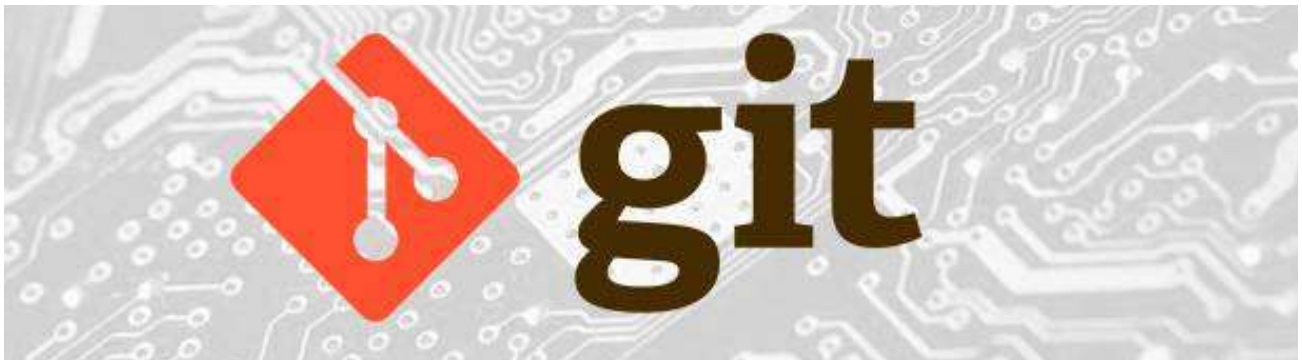
Is DSA knowledge for DevOps Engineering roles?
https://lnkd.in/dges2qU3

Best Programming Language for Devops Engineers ?👨‍💻
https://lnkd.in/dKGpqpnD

Golang: https://lnkd.in/dT-qrmNM

Python for DevOps engineer

4. Git for Version Control



Version control, using Git, is important for tracking changes in code, collaborating with teams, ensuring a reliable history of software development, and facilitating essential DevOps practices such as **CI/CD**, **GitOps**, and storing Infrastructure as Code (IaC) files.

What should you know:

1. Branching:

- Create, switch, and delete branches.
- Understand the role of feature branches.

2. Git Commands:

- Master basic commands (**init**, **clone**, **add**, **commit**, **push**, **pull**, **merge**).
- Explore advanced commands (**rebase**, cherry-pick) for complex workflows.

3. Pull/Merge Requests:

- Create, review, and merge pull requests (GitHub) or merge requests (GitLab).
- Emphasize collaborative code review practices.

4. GitHub/GitLab :

- Learn how to use and store code on remote git repositories.

Git is Important to work as DevOps engineer but also helps if want to make open source contributions 👍

Resources to learn:

Git & GitHub Hands On Tutorial! (Free)

https://ohmygit.org (Game to learn Git)

Pro Git Book FREE

Learn Git by Atlassian FREE

5. Networking & Security 🔐

With rise in DevSecOps, incorporating security practices into the entire development lifecycle is vital. This ensures proactive identification and early addressing of security concerns. Having solid networking knowledge is crucial for securing systems, enabling the configuration of firewalls and implementation of essential network security measures.

Concepts to know:

- IP Addressing
- Ports and Protocols (TCP/IP, UDP, HTTP, DNS)
- Network Services (DHCP, DNS)
- Routing and Switching Basics
- Authentication and Authorization
- Security Best Practices
- Shift-Left Security
- Firewalls and Network Security

Resources to learn:

Networking Course (Free)

Networking Book (Free)

Networking for DevOps (Free)

6. Learn any one Cloud

Companies now prefer deploying applications on cloud providers like **AWS**, **Azure**, or **GCP** instead of on-premise. This shift enables organizations to take advantage of the scalability, flexibility, and cost efficiency provided by cloud services.

AWS is most popular cloud provider. Other popular ones are Microsoft azure, Google cloud Platform, Oracle cloud etc.

What should you know as a DevOps Engineer:

for AWS, you should know the basics of:

1. VPC (Virtual Private Cloud): Create isolated networks.
2. EC2 (Elastic Compute Cloud): Manage virtual servers.
3. S3 (Simple Storage Service): Scalable object storage.
4. RDS (Relational Database Service): Managed databases.
5. IAM (Identity and Access Management): Secure access control.
6. Lambda: Serverless code execution.
7. Route 53: Scalable DNS service.

8. CloudWatch: Monitoring and observability.

9. EKS (Elastic Kubernetes Service): Managed Kubernetes service for container orchestration

Resources to learn:

AWS Course on Udemy (Paid)

AWS services to know as DevOps engineer (Free)

AWS vs Azure vs GCP (Free)

AWS SkillBuilder Platform

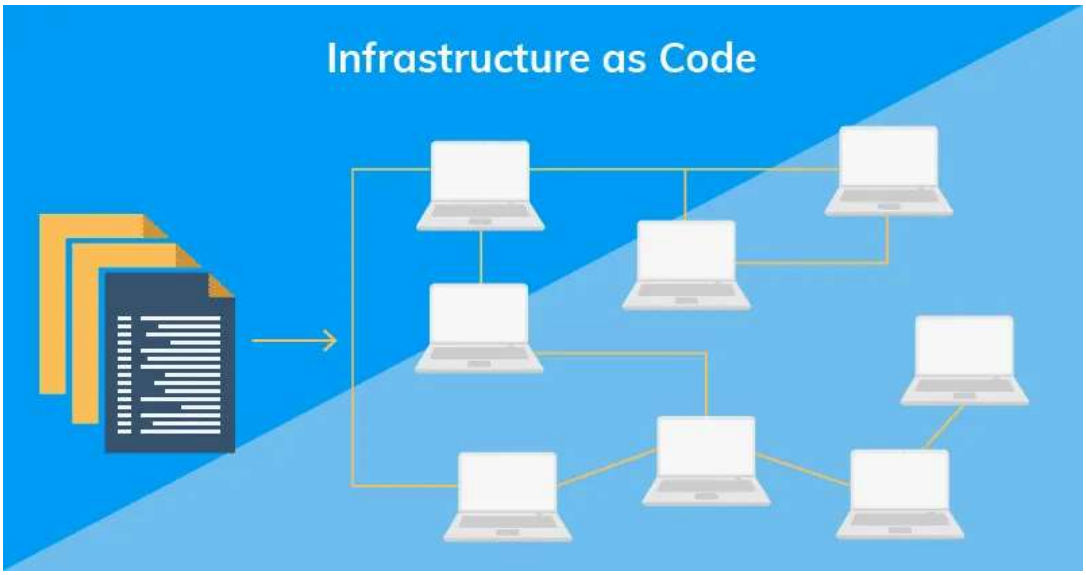
✏️ **Tip!**

Once you master and get certified in any one cloud, it's easy to learn others.

Popular cloud certifications for DevOps engineers include AWS Certified DevOps Engineer – Professional, Microsoft Certified: Azure DevOps Engineer Expert, and Google Cloud Professional DevOps Engineer.

7. Infrastructure as Code

Tools like Terraform and Ansible uses the IaC concept to automate the manual creation and configuration which are time consuming and error prone.



Using IAC we code to create and configure infrastructure & there are 2 types of IAC tools:

1. Infrastructure Provisioning
2. Configuration Management

Infrastructure Provisioning

Creating virtualized resources programmatically, such as servers in the cloud, using tools like **Terraform** to define and deploy infrastructure.

Configuration Management

Ensuring consistent configurations of virtualized resources over time, managed through code, for example using **Ansible** playbooks to define and enforce server

Benefits of having everything in code -

- Ensures consistency accross different environments (stage, dev, prod)
- Can be stored and version controlled and collaborated.

- Less chances of issues or configuration errors.
- Saves time avoiding to repeat creation and configuration.

Resources to learn:

Difference between Terraform and Ansible

Terraform Course in 1 hour (Free)

Ansible course on Udemy

How Ansible works?

Terraform Project

8. Microservices and Containerization 📦

Microservices is the new way to deploy applications independently for better scaling and management, and they are packaged in a container. 📦

Container: A packaged application with its dependencies, isolated for easy deployment, and capable of running consistently on any machine.

Docker is the most popular container technology.

What should you know

- Microservices and Containerization concepts
- How to write docker files? (Containerize the application)
- Docker Commands (build, run, exec, ps)
- How to push and pull docker images in DockerHub

Resources to learn:

What is microservices (Free)

Microservices DevOps Project - Python application on K8s (Free)

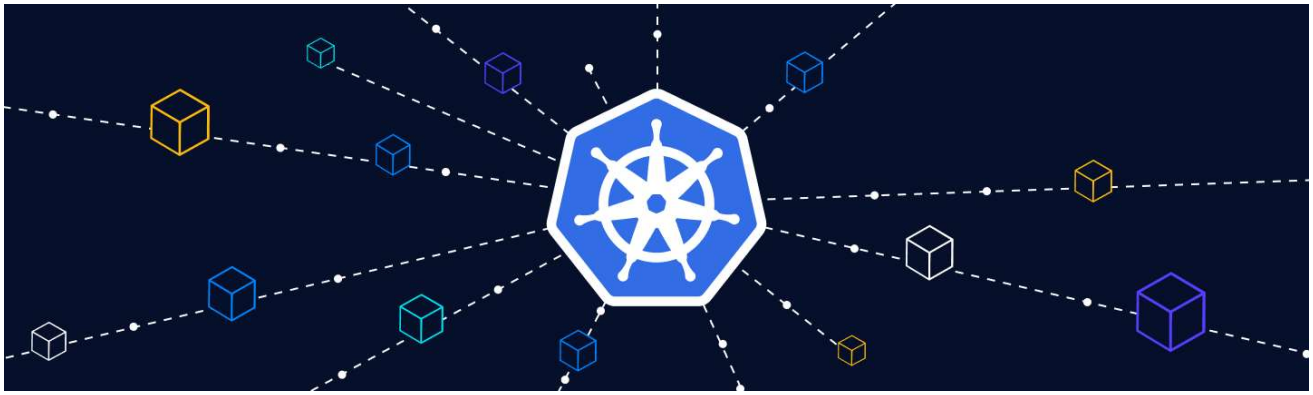
Docker Project (Free)

Docker Course (Free)

Docker Course (Paid)

9. Container Orchestration (Kubernetes)

When using microservices an application can have hundreds or thousands of containers on multiple servers. Managing, scaling and other operations for these containers require container orchestration tools like Kubernetes.



Kubernetes is vast so what should a DevOps engineer know

- What is Kubernetes and How it works
- Cluster creation and administration
- Deploying application K8s cluster
- Kubernetes Commands

Popular Kubernetes Certifications:

CKA, **CKAD**, **CKS**

✏️ **Learnining Kubernetes now, is the best thing you can do to your career!**

Resources to learn:

Prerequisites before learning Kubernetes (Free)

Kubernetes course by KodeKloud

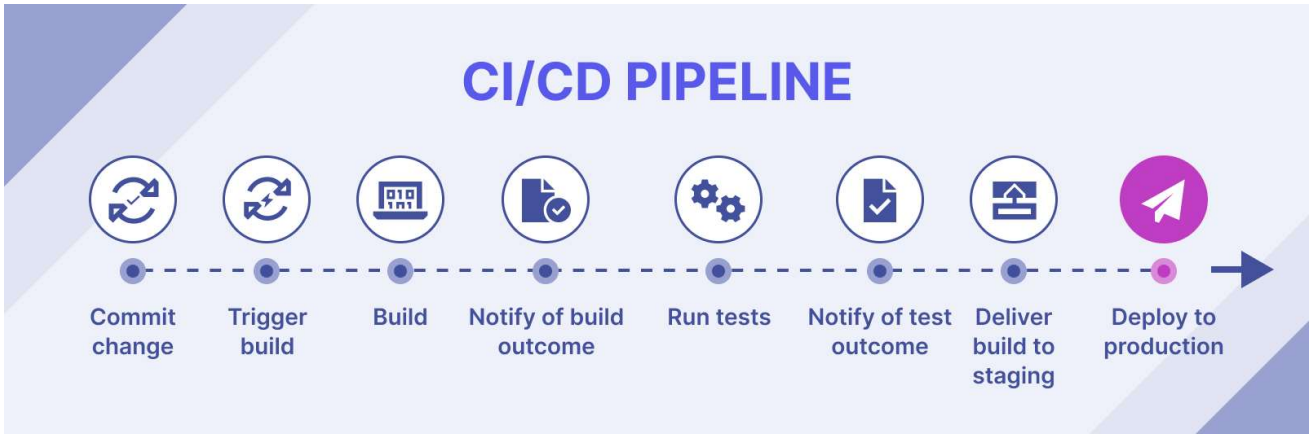
Kubernetes course on Youtube (Free)

DevSecOps project on Kubernetes (Free)

Live Kubernetes Project (Free)

10. CICD Pipelines

CICD is really important to know!!! Because every company deploys software in automated fashion using CICD.



CI/CD Pipeline ensures that each code change, whether introducing new features or fixing bugs, seamlessly integrates into the existing application and deploys continuously and automatically for end-users. This automation streamlines development processes, promotes consistency, and speeds up the delivery of software innovations.

Things to know:

- Writing Pipeline scripts
- Integrating Tests
- Trigger build

Popular CICD tools : Gitlab CI, GitHub Actions, Jenkins, Circle CI.

Resources to learn:

What is CICD Pipeline? (Free)

Gitlab CICD Complete Hands On course (Free)

Jenkins Course (Free)

Github Actions CICD Tutorial (Free)

11. Monitoring and Logging

Once the software is released you need to continuosly monitor it to track performance and discover problems

Monitoring in DevOps serves the purpose of identifying and addressing issues early on, providing benefits such as enhanced performance, ensured reliability, and support for proactive maintenance.

Key aspects include monitoring **software**, **infrastructure**, and **user experience**, ultimately resulting in a resilient and continuously improving software delivery process.

Logging captures system events for troubleshooting and performance monitoring, providing insights for issue diagnosis, ensuring reliability, and supporting data-driven decisions in software delivery.

Popular tools used for Observability: **Prometheus**, **Grafana**, Cloudwatch, Nagios, **ELK** etc

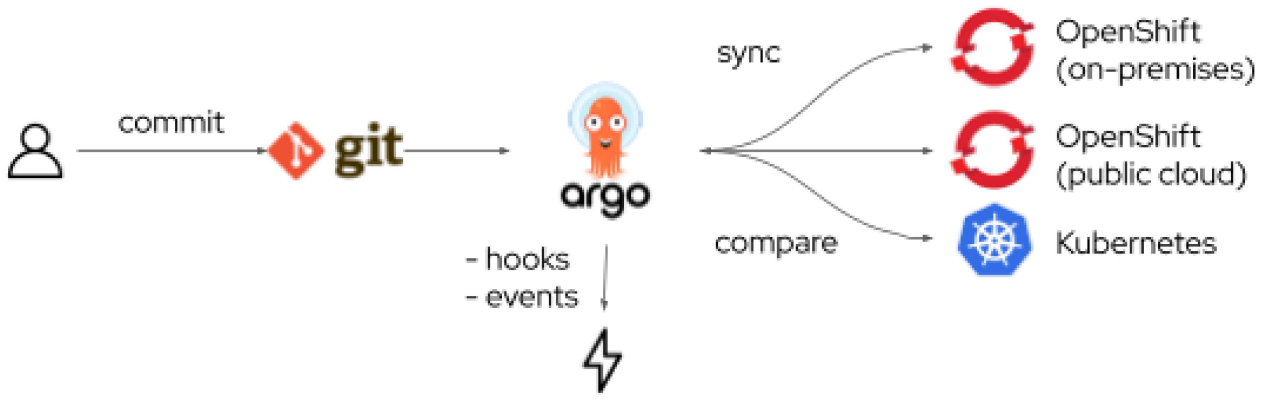
Resources to learn:

[Popular monitoring tools in devops]

[How Prometheus Monitoring works]

12. Gitops

GitOps is a development methodology that leverages Git repositories as the single source of truth for both infrastructure and application code. In simple terms, it involves using Git to manage and automate the entire software delivery process, from version control to deployment.



What to know for GitOps:

- Git
- CICD
- IaC

Popular GitOps tools : **ArgoCD** and FluxCD

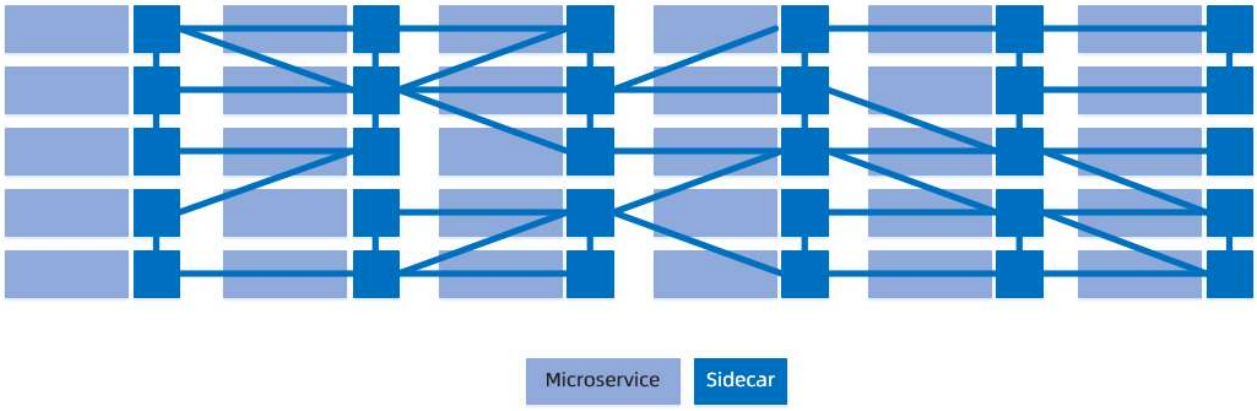
Resources to learn:

[GitOps explained with a Project on ArgoCD]

[GitOps By Gitlab]

13. Service Mesh

Companies use service mesh to enhance the reliability, observability, and control of their microservices architecture. It simplifies the management of communication between microservices, providing features like load balancing, service discovery, encryption, and monitoring.



How Service mesh helps in Microservices architecture:

- Service Discovery
- Traffic Management
- Monitoring and Logging
- Manage Permission
-

Popular Service meshes: **Istio**, Consul connect, Linkerd

Resources to learn:

Project to learn Istio

Service mesh Overview