



Devops on *AWS* for beginner

Linh Nguyen

Engineering Consultant, AWS Cloud Solution Architect

Giới thiệu về DevOps & CICD

“Không có việc gì khó, chỉ sợ không biết làm!”

Copyright@Linh Nguyen on Udemy
All right reserved

Giới thiệu về DevOps & CI/CD

Nội dung

- *Định nghĩa DevOps là gì? CI/CD là gì?*
- *Các nguyên tắc của DevOps và CI/CD.*
- *DevOps thay đổi cách chúng ta làm việc như thế nào?*
- *Bạn có cần biết thật nhiều tool/công nghệ để trở thành một DevOps Engineer?*
- *Cần trang bị gì để trở thành một DevOps Engineer?*

Copyright@Linh Nguyen on Udemy
All right reserved

Định nghĩa DevOps

DevOps là một phương pháp phát triển phần mềm và quản lý hệ thống, nhằm thúc đẩy sự hợp tác giữa các nhóm phát triển (Development) và quản lý / vận hành hệ thống (Operations). Tên gọi "DevOps" là sự kết hợp của hai từ: "Dev" trong Development và "Ops" trong Operations.

Mục tiêu của DevOps là cải thiện quy trình phát triển và triển khai phần mềm, để tạo ra các sản phẩm và dịch vụ có **chất lượng cao**, đáng **tin cậy** và có khả năng điều chỉnh **nhANH chóng** để đáp ứng nhu cầu **thay đổi** của khách hàng.

DevOps giúp giảm thời gian và rủi ro trong việc triển khai phần mềm, tăng cường sự linh hoạt và phản hồi nhanh chóng, và tạo ra môi trường làm việc tích hợp giữa các nhóm khác nhau để đạt được mục tiêu kinh doanh một cách hiệu quả hơn.

Copyright@Linh Nguyen on Udemy
All right reserved

Định nghĩa DevOps

Hiểu đúng về DevOps

DevOps là một văn hoá, triết lý trong phát triển và triển khai phần mềm.

DevOps là một cách tiếp cận và tổ chức quy trình làm việc.

Hiểu sai về DevOps

DevOps là tên gọi của một công nghệ.

DevOps là tập hợp các tools (công cụ).

DevOps là tên một chức danh trong công ty / đội dự án.

Copyright@Linh Nguyen on Udemy
All right reserved

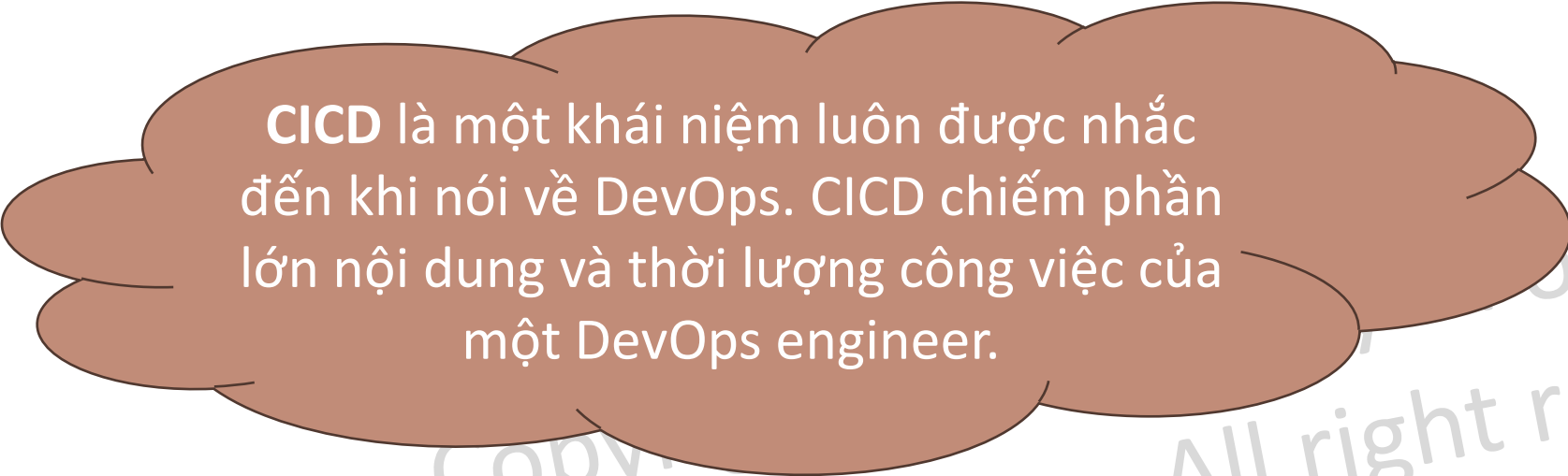
Các nguyên tắc của DevOps

- 1. Tự động hóa** quy trình phát triển, kiểm thử, triển khai, và quản lý hệ thống để giảm thiểu sự can thiệp thủ công và lỗi nhân tạo (human misstake).
- 2. Tích hợp liên tục** (Continuous Integration - CI): Tích hợp liên tục mã nguồn từ nhiều nguồn khác nhau và test tự động để đảm bảo tính ổn định của ứng dụng.
- 3. Triển khai liên tục** (Continuous Deployment - CD): Tự động triển khai các phiên bản mới của ứng dụng vào môi trường một cách tự động và an toàn.
- 4. Theo dõi và phản hồi** (Monitoring and Feedback): Theo dõi sự hoạt động của ứng dụng và hệ thống thời gian thực để phát hiện và khắc phục lỗi nhanh chóng.
- 5. Tự phục vụ** (Self-service): Cung cấp các công cụ cho các nhóm phát triển (Dev) và vận hành (Operation) để họ có thể làm việc một cách độc lập và hiệu quả.

Copyright@Linh Nguyen on Udemy
All right reserved

Các nguyên tắc của DevOps

CI/CD là viết tắt của **Continuous Integration** và **Continuous Deployment** (Hoặc **Continuous Delivery**), đây là một phương pháp quy trình phát triển phần mềm để tạo và triển khai ứng dụng một cách tự động và liên tục. Đây là một trong những phần quan trọng của quy trình phát triển phần mềm hiện đại và giúp tối ưu hóa quy trình phát triển, kiểm tra, và triển khai phần mềm.



CI/CD là một khái niệm luôn được nhắc đến khi nói về DevOps. CI/CD chiếm phần lớn nội dung và thời lượng công việc của một DevOps engineer.

Copy

on Udemy
All right reserved

Quy trình phát triển & triển khai Software khi KHÔNG có DevOps

Development Team

Coding on Local machine



Unit Testing

Integration Testing

Security Testing

...

Build Source Code



Artifacts



Operation Team

Prepare needed environment (Server, VM, Database...)

Monitoring

Network maintenance

Version control

Backup & restore

Deploy Artifact to Environment



Receive packages

Install needed tool/middleware

Copyright ©

All rights reserved

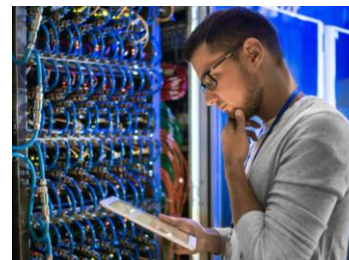
Quy trình phát triển & triển khai Software khi KHÔNG có DevOps

Development Team



Code của tôi chạy ngon ở Local cơ mà?
Rõ ràng là môi trường của anh có vấn đề!!!

Operation Team



Lần release này môi trường vẫn thế mà?
Anh đâu có đưa chỉ dẫn đầy đủ những tool cần thiết phải cài?

Vấn đề là gì?

Thiếu sự cộng tác giữa hai team

Team Development không ý thức được sự khác biệt giữa một software chạy trên môi trường thật và chạy tại máy local.

Team Operation không tham gia ngay từ đầu dự án để có những tư vấn cần thiết cho team Dev

Quy trình làm việc giữa hai team không được định nghĩa một cách rõ ràng và đầy đủ.

Không áp dụng Automation & CICD để tăng hiệu suất công việc.

Copyright@Linh Nguyen
All right reserved

Quy trình phát triển & triển khai Software khi có DevOps



Chia sẻ về kiến trúc, framework của phần mềm dự kiến được phát triển.

Xác định các công việc cần có sự hỗ trợ của team Operation (vd: Software scan, Automation test)

Có các chỉ dẫn cần thiết cho việc deploy trên cơ sở tham khảo các tư vấn từ team Operation.



Tham gia ngay từ đầu trong việc tư vấn và thiết kế hệ thống.

Xác định và lên chiến lược cụ thể cho các thành phần cần triển khai.

Chiến lược Automation, CI/CD & release lên các môi trường.

Phương án Monitor, Backup, Scaling cho các thành phần của hệ thống.

Quy trình phát triển & triển khai Software khi có DevOps



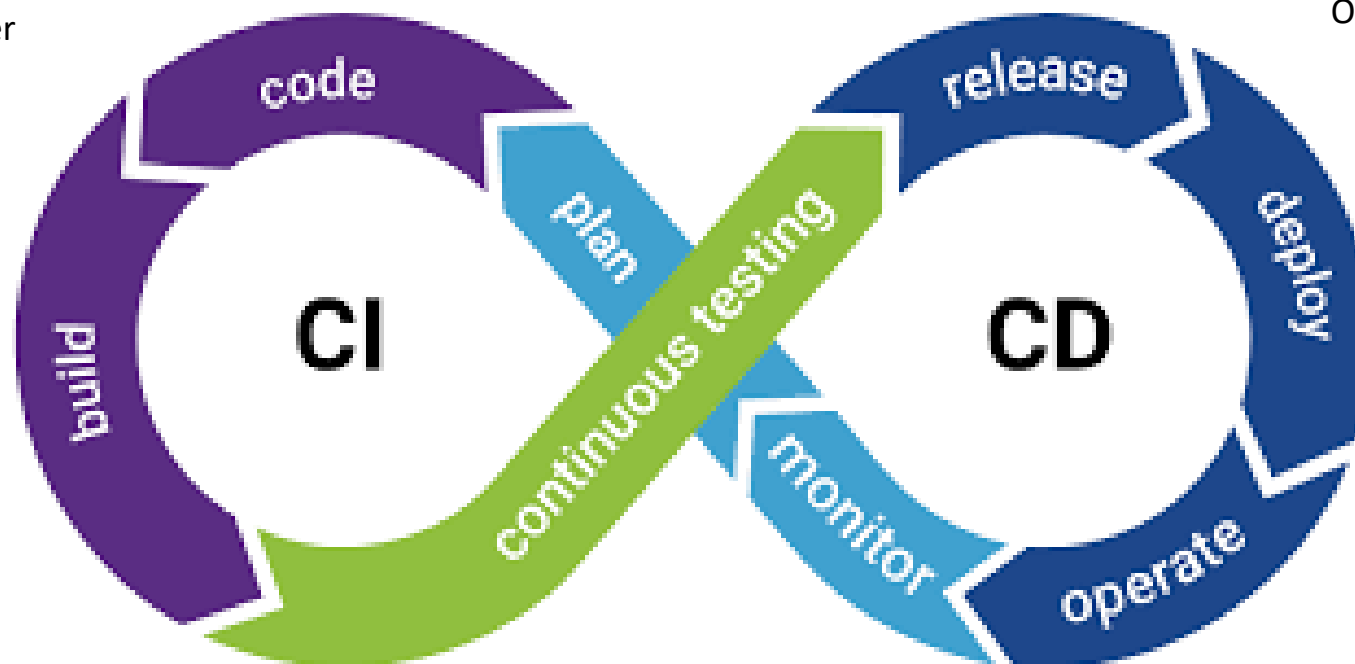
Developer



Operator



Project Manager



Tester

Copyright @ Linh Nguyen
All right reserved
on Udemy

Vai trò của DevOps Engineer trong dự án

Vai trò của DevOps Engineer trong dự án phát triển và vận hành phần mềm là quan trọng và đa dạng.

Người này đóng vai trò gắn kết giữa nhóm **phát triển** và nhóm **vận hành**, đảm bảo rằng quy trình phát triển và triển khai phần mềm diễn ra một cách suôn sẻ và hiệu quả.

Copyright@Linh Nguyen on Udemy
All right reserved

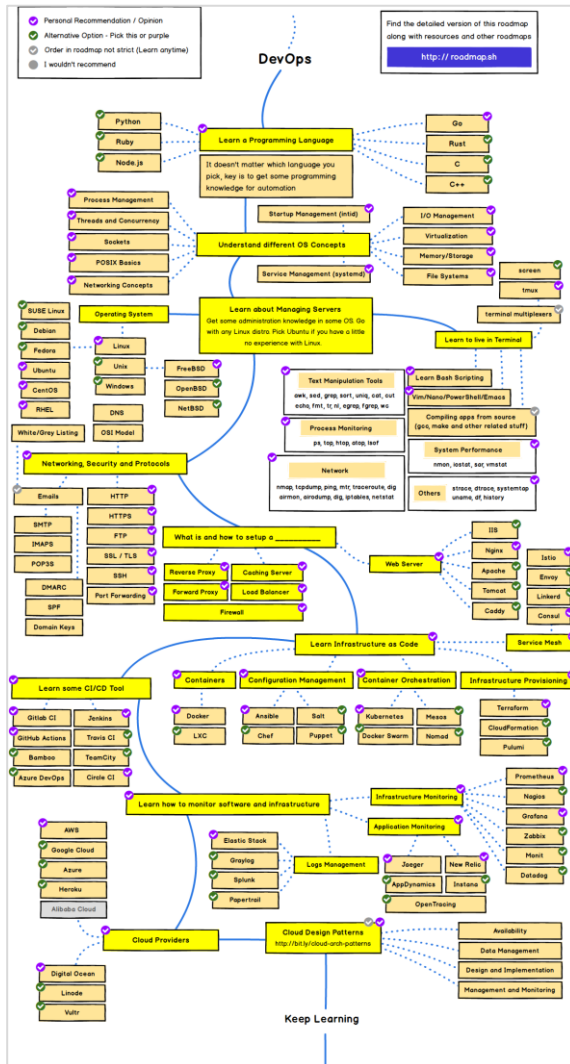
Vai trò của DevOps và DevOps engineer trong dự án

1. **Tự động hóa quy trình:** chịu trách nhiệm thiết lập và duy trì các công cụ và quy trình tự động hóa. Bao gồm việc xây dựng infra tự động (IaC), Automation test, CI/CD pipelines.
2. **Quản lý hệ thống và môi trường:** DevOps engineer giám sát, quản lý và tối ưu hóa hệ thống và môi trường làm việc. Điều này bao gồm việc cấu hình và triển khai Server, DB, và đảm bảo tính sẵn sàng và bảo mật của hệ thống.
3. **Tích hợp liên tục (Continuous Integration):** xây dựng và duy trì hệ thống Integration liên tục để kiểm tra source code và automation test để đảm bảo tính ổn định của ứng dụng.
4. **Triển khai liên tục (Continuous Deployment):** xây dựng và duy trì quy trình tự động để triển khai phiên bản mới của ứng dụng vào các môi trường (Test, Staging, Production).
5. **Bảo mật và giám sát:** DevOps engineer thực hiện các biện pháp bảo mật để bảo vệ hệ thống và dữ liệu. Họ cũng xây dựng hệ thống Monitoring để theo dõi hiệu suất và sự hoạt động của ứng dụng, đồng thời cảnh báo về các sự cố.
6. **Hỗ trợ và sửa lỗi:** Khi có sự cố hoặc lỗi xảy ra trong quá trình triển khai hoặc vận hành, DevOps engineer thường chịu trách nhiệm tìm ra nguyên nhân và sửa chữa chúng một cách nhanh chóng.
7. **Hợp tác và giao tiếp:** DevOps engineer phải tương tác chặt chẽ với các thành viên khác trong dự án, bao gồm nhóm Developer, PM, Tester, QA, System Admin, Operator, để đảm bảo sự hợp tác mạnh mẽ và đạt được mục tiêu chung của dự án.

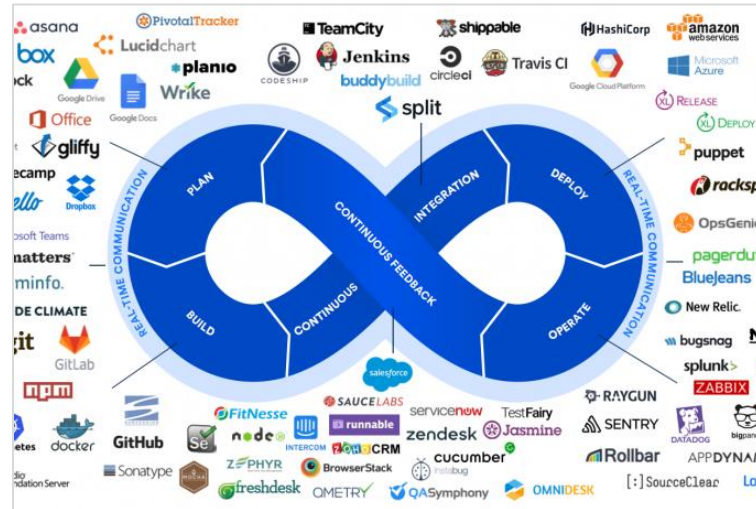
Copyright @ Linh Nguyen on Udemy
All right reserved

Ma trận DevOps và CI/CD

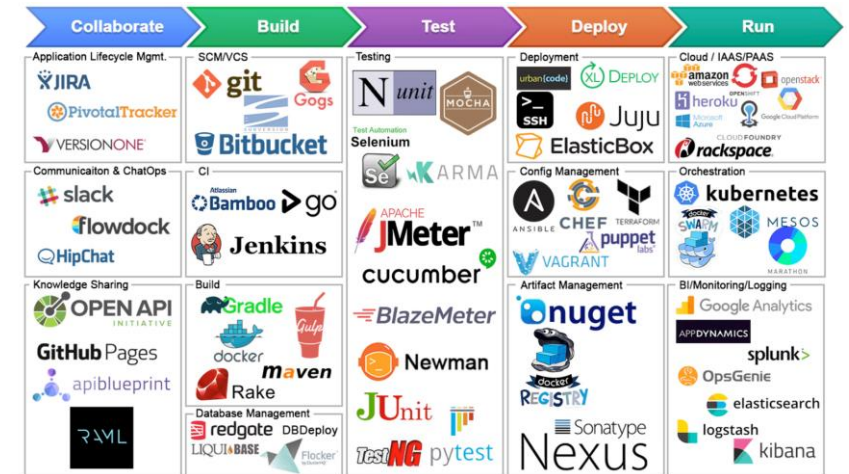
“DevOps roadmap”



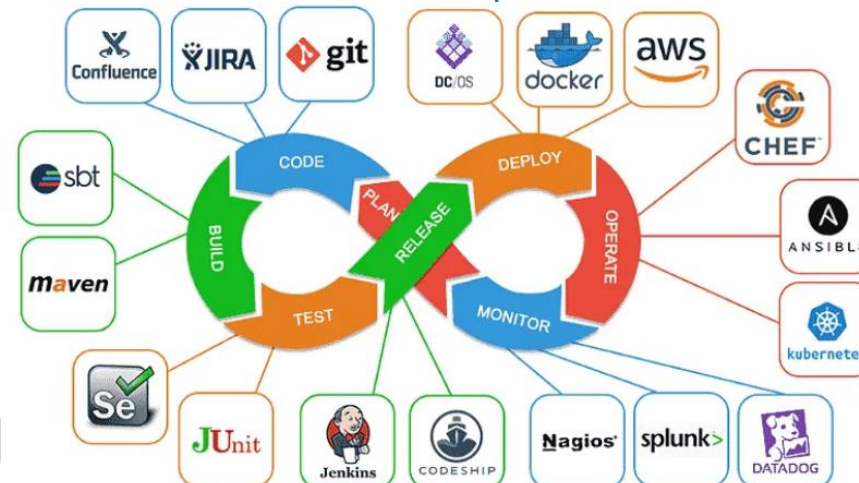
“DevOps tools”



“CI/CD”



“CI/CD concept”



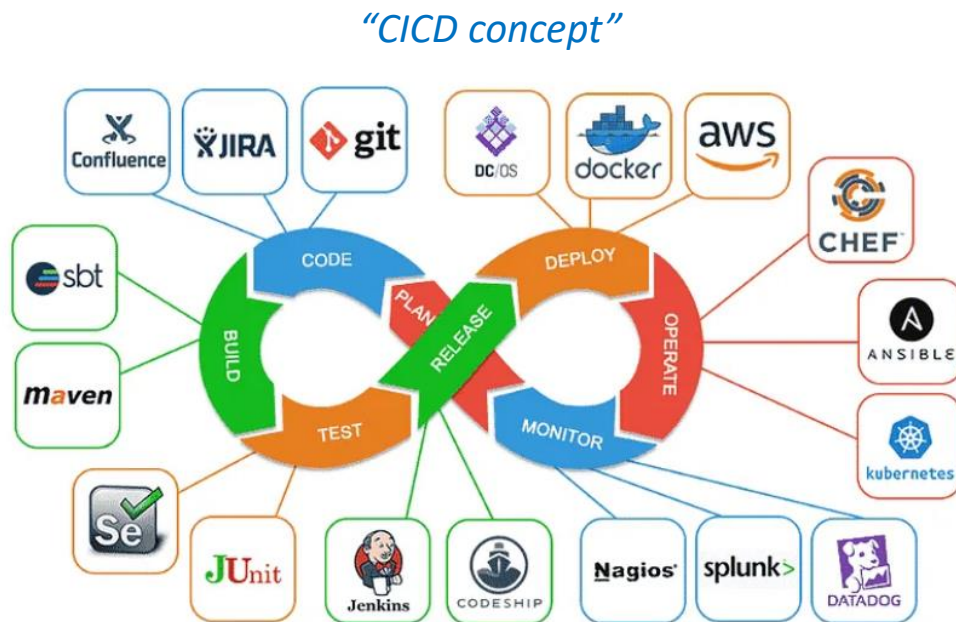
Copy!

on Udemy reserved

Ma trận DevOps và CICD

Bạn không cần phải là người biết sử dụng tất cả mọi tool để trở thành một DevOps Engineer!

***Nhắc lại:** DevOps là một nét văn hoá, là triết lý, không phải tools hay công nghệ!*



Việc lựa chọn tool nào để apply vào các step của CICD hoàn toàn dựa vào kinh nghiệm của đội dự án và các yếu tố khác như phí license. Hoàn toàn không có quy định kiểu tool A phải đi với tool B.

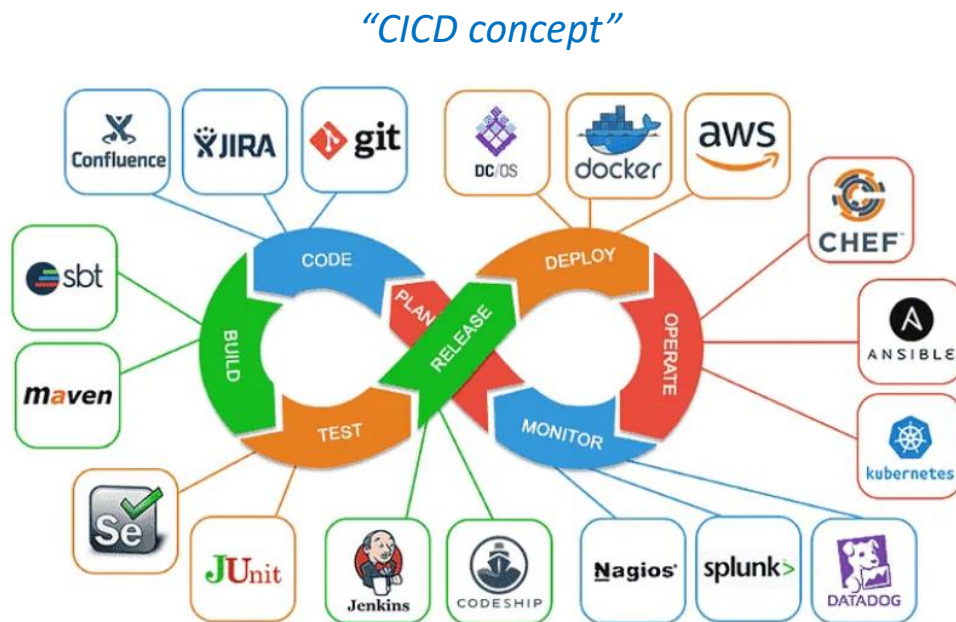
“Vô chiêu thắng hữu chiêu”

“DevOps engineer giỏi không nhất thiết là người biết nhiều tools”

Copyright @ Linh Nguyen on Udemy
All right reserved

Ma trận DevOps và CI/CD

Liệu có một giải pháp tối ưu & toàn diện cho CI/CD để apply cho tất cả các dự án?



CI/CD là “may đo” (tailor). Sẽ không có one solution for all problem. Việc apply bao nhiêu quy trình, tools, phương pháp đơn giản hay phức tạp phụ thuộc hoàn toàn vào yêu cầu của project, thời gian, ngân sách, kinh nghiệm v.v...

Copyright@Linh Nguyen on Udemy
All right reserved

Ma trận DevOps và CICD

Liệu có một giải pháp tối ưu & toàn diện cho CICD để apply cho tất cả các dự án?



Trang bị kiến thức gì để trở thành DevOps engineer?

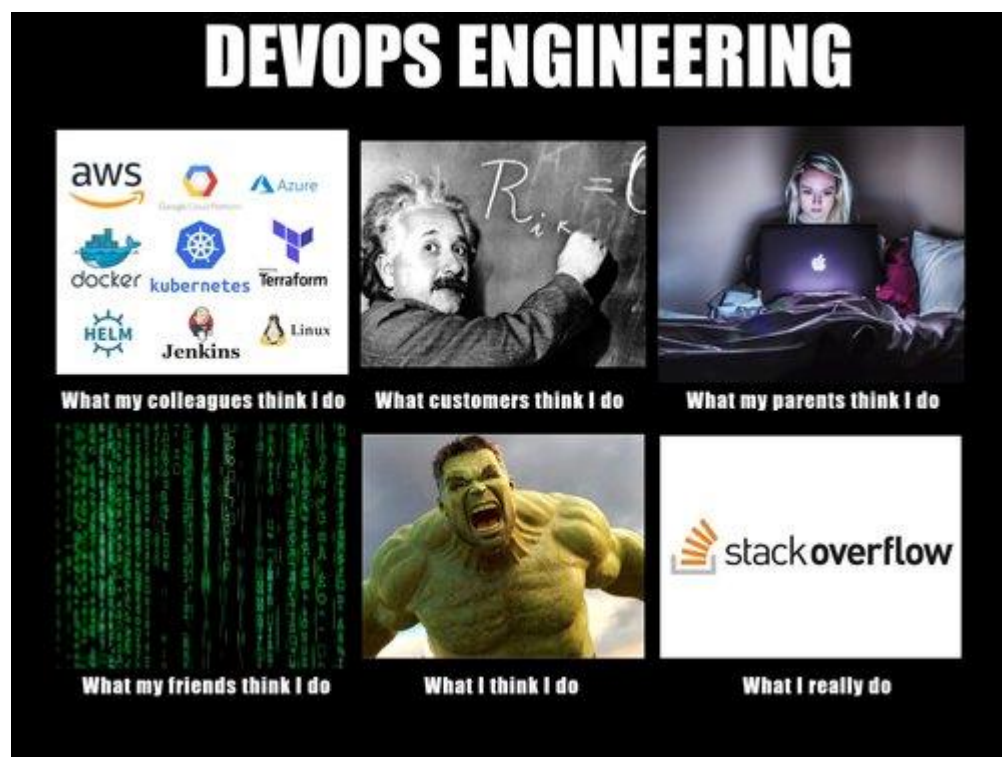
- Cloud Computing (AWS, Azure, GCP...)
- Operating System (Windows Server, Linux, Redhat, CentOS)
- Version Control System (Github, Gitlab, Bitbucket)
- Project management tool (Jira, Trello)
- Networking, Database, Security & other Cloud's managed services
- Automation and CI/CD tools (Terraform, Ansible, Jenkins, GitlabCI, Github CI, Code Pipeline, CircleCI...)
- Framework (Angular, Nodejs, Spring Boot, .NET, ...)
- Docker and Kubernetes
- Monitoring, Alert and Notification tools
- Other: Problem solving, Communication skill

Copyright @ Linh Nguyen on Udemy
All right reserved

Trang bị kiến thức gì để trở thành DevOps engineer?

QUAN TRỌNG NHẤT:

Giữ được thái độ bình tĩnh trong mọi tình huống để không delete server!!!

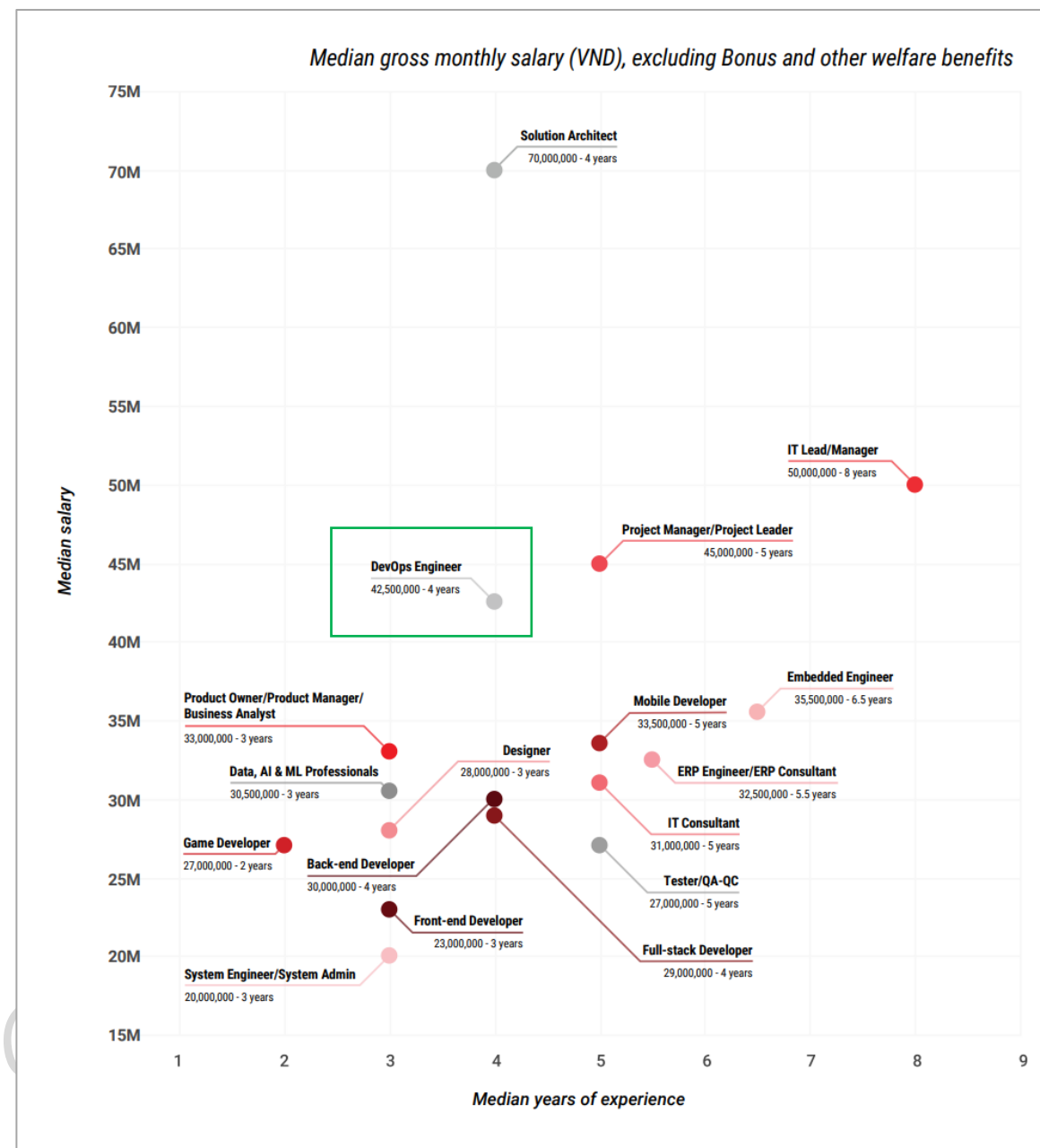


Copyright @ All rights reserved

Cơ hội việc làm cho DevOps engineer

MEDIAN SALARY BY IT POSITION & RANGE OF EXPERIENCE

	Under 1 year	From 1 to less than 3 years	From 3 to less than 5 years	From 5 to 8 years	Over 8 years
Software Development					
Back-end Developer	12,000,000	18,000,000	29,000,000	37,000,000	49,000,000
Front-end Developer	11,000,000	15,000,000	30,000,000	40,000,000	70,000,000
Full-stack Developer	17,000,000	19,000,000	25,000,000	37,000,000	39,500,000
Mobile Developer	12,000,000	19,000,000	27,000,000	42,000,000	57,500,000
Game Developer	19,000,000	22,000,000	28,500,000	32,000,000	51,000,000
Embedded Engineer	15,000,000	37,000,000	N/A	41,000,000	N/A
Product Management & Business Analysis					
Product Owner/Product Manager/Business Analyst	12,500,000	28,000,000	33,000,000	39,000,000	40,000,000
Management & Consulting					
Project Manager/Project Leader	19,500,000	35,000,000	44,500,000	N/A	58,000,000
IT Lead/Manager	N/A	N/A	39,500,000	42,500,000	61,000,000
IT Consultant	N/A	N/A	27,500,000	34,000,000	N/A
Designing					
Designer	N/A	25,000,000	27,000,000	N/A	50,000,000
Testing					
Tester or QA-QC	11,500,000	15,000,000	20,000,000	30,000,000	46,000,000
Infrastructure					
System Engineer/System Admin	13,000,000	17,000,000	20,000,000	26,000,000	28,000,000
DevOps Engineer	N/A	31,000,000	40,000,000	61,000,000	65,000,000
Data, AI & Machine Learning					
Data, AI & ML Professionals	12,000,000	24,000,000	38,000,000	40,000,000	42,500,000



Thanks you and see you in next chapter!

Copyright@Linh Nguyen on Udemy
All right reserved