DEVOPS PETA JALAN

oleh TechWorldwithNana

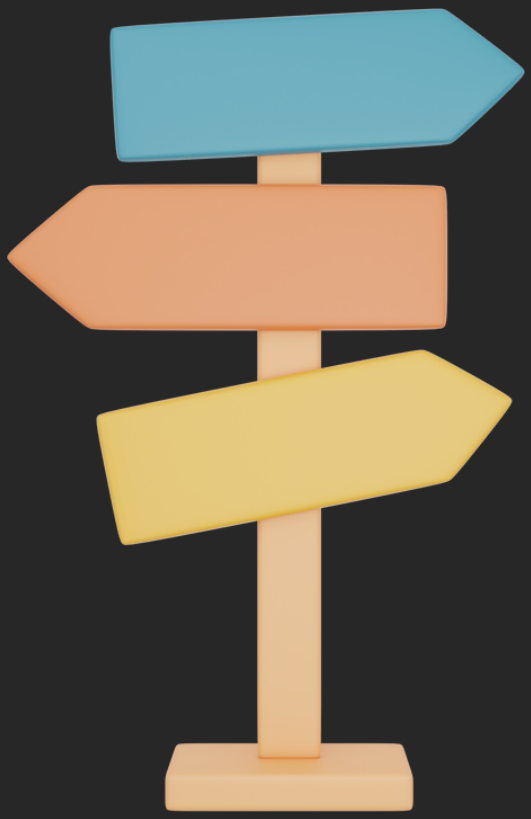
TABEL O F CO NTENDA

Mulai di sini

Ini adalah sebuah**jalur langkah demi langkah yang akan saya ambil sebagai profesional dan pendidik DevOps**, kalau saya mulai dari nol lagi. Menunjukkan kepada Anda jalur apa yang paling efisien untuk menjadi insinyur DevOps berdasarkan pengetahuan yang saya miliki sekarang.

Saya harap dengan ini saya dapat membantu Anda dalam perjalanan yang sangat bermanfaat namun menantang ini menuju DevOps.

Dan saya ingin membuatnya**lebih individual**untuk Anda berdasarkan latar belakang tempat Anda beralih ke DevOps:



- administrator sistem
- pengembang perangkat lunak
- uji otomatisasi insinyur
- insinyur jaringan
- seseorang dengan nol atau sangat sedikit pengetahuan TI

Jadi setelah peta jalan DevOps, Anda akan menemukan informasi tentang cara memulai di DevOps yang memiliki latar belakang ini.

Satu hal penting sebelumnya:
Karena DevOps mencakup seluruh siklus hidup pengembangan perangkat lunak, itu berarti Anda bekerja dengan banyak teknologi. Plus DevOps masih berkembang dan ada banyak alat baru yang dikembangkan setiap saat.

Jadi kamu harus begitu**nyaman dengan terus belajar**dan memperluas pengetahuan Anda, bahkan setelah Anda menjadi insinyur DevOps.

TECHWORLD WITH NANA

01

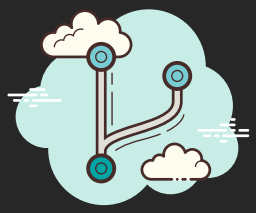
TECHWORLD WITH NANA

Konsep Perangkat Lunak Perkembangan

Sebagai insinyur DevOps, Anda tidak akan memprogram aplikasi, tetapi karena Anda bekerja sama dengan tim pengembangan untuk meningkatkan dan mengotomatiskan tugas untuk mereka, Anda perlu melakukannya **memahami konsep-konsep**dari:

Bagaimana pengembang bekerja dan berkolaborasi (Agile, alur kerja Jira)

Apa alur kerja Git mereka menggunakan



Bagaimana aplikasinya dikonfigurasi (Bangun & Alat pengemasan)

Pengujian otomatis dan lingkup pengujian

Dan umumnya mengerti apa keseluruhannya **siklus hidup pengembangan perangkat lunak** mencakup dari ide ke kode, hingga merilisnya ke pengguna akhir!

www.techworld-with-nana.com/devops-roadmap

02

Dasar OS & Linux

Sebagai insinyur DevOps, Anda bertanggung jawab untuk menyiapkan dan memelihara infrastruktur (server) tempat aplikasi diterapkan.

Jadi, Anda perlu mengetahui dasar-dasar cara mengelola server dan menginstal berbagai alat di dalamnya.

Konsep Dasar Sistem Operasi Anda perlu mengerti**:**

- Perintah Shell
- Sistem File & Izin Linux
- Manajemen Kunci SSH
- Virtualisasi

Karena sebagian besar server menggunakan OS Linux, Anda perlu mengetahui dan merasa nyaman menggunakannya**Linux,**terutama Komandonya Antarmuka Garis.

Anda juga perlu mengetahui**dasar-dasar Jaringan & Keamanan**untuk mengkonfigurasi infrastruktur, seperti:

- Konfigurasikan Firewall untuk
- mengamankan akses Pahami cara kerja alamat IP, port, dan DNS
- Load Balancer
- Proksi
- HTTP/HTTPS

Namun, untuk menarik garis di sini antara DevOps dan Operasi TI: Anda tidak perlu menjadi SysAdmin. Jadi**tidak ada pengetahuan lanjutan**administrasi server diperlukan di sini. Cukup mengetahui dasar-dasarnya. Ada profesi sendiri seperti SysAdmins, Networking atau Security Professionals untuk kasus penggunaan yang lebih lanjut.

03

TECHWORLD WITH NANA

Kontainerisasi - Docker

Karena wadah telah menjadi standar baru pengemasan perangkat lunak, kemungkinan besar Anda akan menjalankan aplikasi Anda sebagai wadah.

Ini berarti Anda perlu memahami secara umum:

- konsep virtualisasi konsep
- containerisasi cara mengelola
- aplikasi dalam container di server.

Wadah adalah satuan standar dari perangkat lunak itu mengemas kode dan semua dependensinya sehingga aplikasi berjalan dengan cepat dan andal di komputasi apa pun lingkungan.

Docker sejauh ini adalah yang paling populer teknologi kontainer!

Beberapa hal yang harus Anda ketahui:

- Jalankan kontainer
- Periksa kontainer aktif
- Docker Networking
- Pertahankan data dengan aplikasi Docker
- Volumes Dockerize menggunakan Dockerfiles
- Jalankan beberapa kontainer menggunakan Docker-Compose
- Bekerja dengan Repositori Docker

Wadah dan mesin virtual memiliki manfaat isolasi dan alokasi sumber daya yang serupa, tetapi fungsinya berbeda.**VM memvirtualisasikan seluruh OS. Kontainer hanya memvirtualisasikan level aplikasi OS.**Oleh karena itu, wadah lebih ringan dan lebih cepat.

04

Pipa CI/CD

TECHWORLD WITH NANA

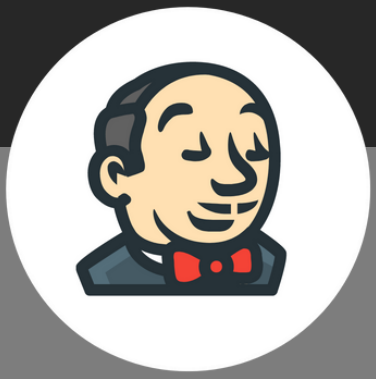
CI/CD adalah jantung dari DevOps.

Di DevOps, semua perubahan kode, seperti fitur baru atau perbaikan bug, harus dilakukan **terintegrasi** dalam aplikasi yang ada dan **dikerahkan**untuk pengguna akhir**terus menerus** dan dalam sebuah **otomatis**jalan.

Oleh karena itu istilah:
Cterus menerus**Saya**integrasi dan**C**terus menerus **D**pekerjaan (CI/CD)

Saat fitur atau perbaikan bug selesai, saluran pipa yang berjalan di server CI (mis. Jenkins) harus dipicu secara otomatis, yang:

1. menjalankan tes
2. paket aplikasi
3. membangun Gambar wadah
4. push wadah Gambar ke repositori gambar
5. menyebarkan versi baru ke server



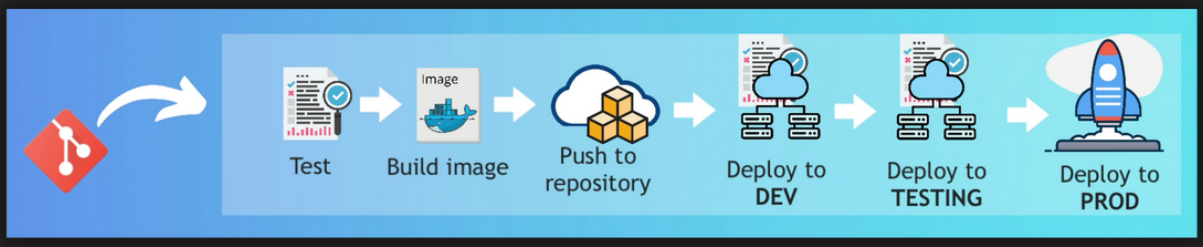
Ada banyak platform CI/CD di luar sana. Yang paling populer saat ini adalah Jenkins

---

Yang populer lainnya: GitLab, GitHub Tindakan, Travis CI, Bambu

Keterampilan yang perlu Anda pelajari di sini:

- Menyiapkan server CI/CD Integrasikan
- repositori kode untuk memicu pipeline secara otomatis
- Build Tools & Package Manager Tools untuk menjalankan tes dan mengemas aplikasi
- Mengonfigurasi repositori artefak (seperti Nexus) dan berintegrasi dengan saluran pipa



05

TECHWORLD WITH NANA

Pelajari satu Penyedia Cloud

Saat ini banyak perusahaan yang menggunakan **infrastruktur virtual di cloud**, alih-alih mengelola infrastruktur mereka sendiri. Ini adalah platform Infrastructure as a Service (IaaS), yang menawarkan beragam
layanan tambahan, seperti pencadangan, keamanan, penyeimbangan muatan, dll.

Layanan ini**khusus platform**. Jadi, Anda perlu mempelajari layanan dari platform khusus tersebut dan mempelajari cara mengelola keseluruhan penerapan
infrastruktur di atasnya.

Misalnya untuk AWS, Anda harus mengetahui dasar-dasar:

- Layanan IAM - mengelola pengguna dan izin
- Layanan VPC - layanan EC2 jaringan
- pribadi Anda - server virtual

aws

AWS adalah platform IaaS yang paling kuat dan paling banyak digunakan, tetapi juga a yang sulit.

Yang populer lainnya: Microsoft Azure, Google Cloud

AWS memiliki banyak layanan, tetapi Anda hanya perlu mempelajari layanan yang benar-benar dibutuhkan oleh Anda/perusahaan Anda. Misalnya ketika kluster K8s berjalan di AWS, Anda perlu mempelajari layanan EKS juga.

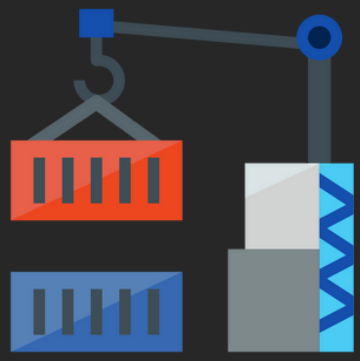
Setelah Anda mempelajari satu platform IaaS, mudah untuk mempelajari yang lain

06

Orkestrasi Kontainer - Kubernetes

Karena kontainer populer dan mudah digunakan, banyak perusahaan menjalankan ratusan atau ribuan kontainer di banyak server. Ini berarti wadah ini perlu dikelola bagaimanapun.

Untuk tujuan ini ada alat orkestrasi wadah.



Alat orkestrasi kontainer seperti Kubernetes, **mengotomatisasikan**penyebaran, penskalaan, dan pengelolaan aplikasi kemas.

Kubernetes (juga dikenal sebagai K8s) adalah orkestrasi kontainer yang paling populer alat

Jadi, Anda perlu belajar:

- Cara kerja Kubernetes
- Bagaimana mengelola dan mengelola kluster K8s
- Cara menyebarkan aplikasi di K8s

Diperlukan pengetahuan khusus K8:

- Pelajari komponen inti seperti, Deployment, Service, ConfigMap, Secret, StatefulSet, Ingress
- Kubernetes CLI (Kubectl)
- Data tetap dengan Ruang Nama
- Volume K8

07

TECHWORLD WITH NANA

Pemantauan & Observabilitas

Setelah perangkat lunak dalam produksi, penting untuk memantaunya**melacak kinerja, menemukan masalah**dalam infrastruktur dan aplikasi Anda.

Jadi salah satu tanggung jawab Anda sebagai insinyur DevOps adalah:

- mengatur pemantauan perangkat lunak
- menyiapkan pemantauan infrastruktur, misalnya untuk kluster Kubernetes dan server yang mendasarinya
- memvisualisasikan data

Prometheus:
Sebuah pemantauan populer dan alat peringatan

Grafana:
Analitik dan alat visualisasi interaktif

Anda juga harus memahami bagaimana sistem dapat mengumpulkan dan mengagregasi data dengan tujuan menggunakannya untuk memecahkan masalah, mendapatkan wawasan bisnis, dll.

Tumpukan ELK:
Catatan populer tumpukan manajemen

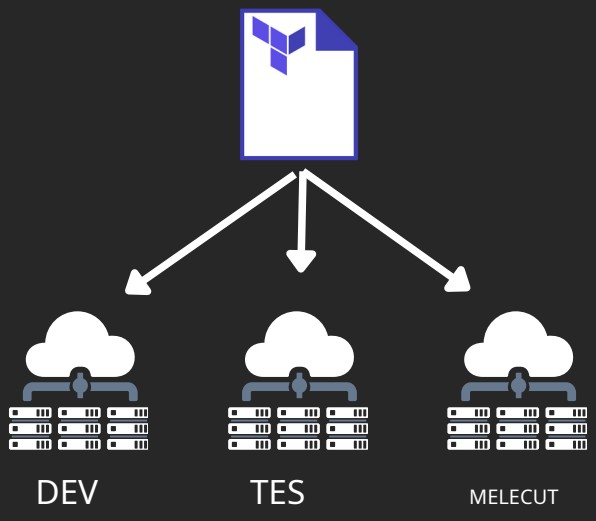
08

TECHWORLD WITH NANA

Infrastruktur sebagai Kode

Membuat dan memelihara infrastruktur secara manual memakan waktu dan rawan kesalahan. Terutama saat Anda membutuhkannya **mereplikasi infrastruktur**, misalnya untuk lingkungan Pengembangan, Pengujian, dan Produksi.

Di DevOps, kami ingin mengotomatiskan sebanyak mungkin dan di situlah Infrastruktur sebagai Kode muncul.



Dengan IAC kami**gunakan kode untuk membuat dan mengonfigurasi infrastruktur**dan ada 2 jenis alat IAC yang perlu anda ketahui :

1. Penyediaan infrastruktur
2. Manajemen konfigurasi

Terraform adalah infrastruktur paling populer alat penyediaan

Ansible adalah konfigurasi yang paling populer alat manajemen

Manfaat memiliki segalanya sebagai kode :

- Mendorong kerjasama dalam tim
- Mendokumentasikan perubahan pada infrastruktur
- Transparansi infrastruktur negara
- Aksesibilitas ke informasi itu di tempat terpusat versus tersebar di mesin lokal orang dalam bentuk beberapa skrip.

09

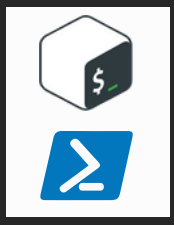
Bahasa Skrip

Karena Anda bekerja erat dengan pengembang dan administrator sistem untuk juga mengotomatiskan tugas pengembangan dan pengoperasian, Anda perlu menulis skrip dan aplikasi kecil untuk mengotomatiskannya.

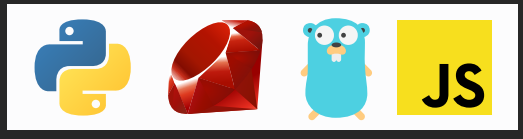
Untuk itu, Anda memerlukan beberapa keterampilan scripting atau pemrograman dasar.

Contoh: skrip utilitas seperti membersihkan cache, memulai build, dan penyebaran dll.

Ini bisa menjadi**bahasa skrip khusus OS** seperti bash atau Powershell.



Tapi yang lebih dituntut adalah **OSindependen**bahasa seperti Python, Ruby atau Go.



Bahasa-bahasa ini lebih kuat dan fleksibel. Jika Anda mengetahui salah satunya, itu akan membuat Anda jauh lebih berharga sebagai insinyur DevOps.

Python adalah salah satu bahasa pemrograman yang paling populer dan mudah digunakan

mempelajari

Ada banyak bahasa pemrograman, tetapi saya akan merekomendasikan untuk memulai dengan Python. Python digunakan secara luas, mudah dipelajari dan digunakan untuk banyak kasus penggunaan yang berbeda, terutama di DevOps.

Anda tidak memerlukan level yang sama dengan pengembang perangkat lunak. Mempelajari cara menulis skrip dengan Python sudah cukup.

Dan hal baiknya adalah, konsep pemrograman tetap sama, jadi saat Anda mempelajari satu bahasa dengan baik, Anda dapat dengan mudah mempelajari bahasa baru dengan cukup cepat.

10

Kontrol Versi - Git

Anda menulis semua logika otomasi sebagai kode. Dan hanya kode aplikasi, kode otomasi juga harus**dikelola dan dihosting** pada suatu **alat kontrol versi**, seperti Git.

File konfigurasi K8s

Skrip python

e

Git adalah**Alat CLI**, yang Anda instal secara lokal. Ini memungkinkan pelacakan perubahan dalam kode sumber dan memungkinkan lebih baik kolaborasi pada kode.

Jadi, Anda perlu belajar:

- Perintah inti Git, seperti git clone, git branch, git pull/Push, git merge dll
- Tetapi juga bagaimana berkolaborasi dalam suatu proyek, seperti membuat permintaan tarik, ulasan kode, percabangan

Anda perlu belajar Git. Itu yang paling versi populer dan banyak digunakan alat kontrol

File Anda disimpan secara terpusat di repositori Git jarak jauh di web. Repositori Git yang paling populer adalah GitHub dan GitLab.

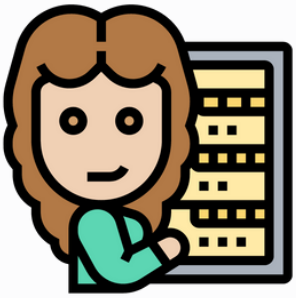
Repositori Git

JANGAN simpan rahasia dan kata sandi di repositori Git Anda

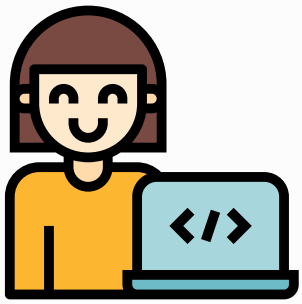
Mulai dari..

Memiliki keterampilan DevOps tersebut adalah tujuan akhir, tetapi banyak dari Anda memulai perjalanan DevOps Anda**berbagai latar belakang yang berbeda**.

Jadi titik awalnya berbeda untuk Anda semua. Anda mungkin memulai sebagai administrator sistem atau insinyur perangkat lunak atau insinyur otomasi pengujian, dll. Atau mungkin tidak memiliki latar belakang TI sama sekali dan ingin beralih ke DevOps:



Administrator Sistem



Pengembang perangkat lunak

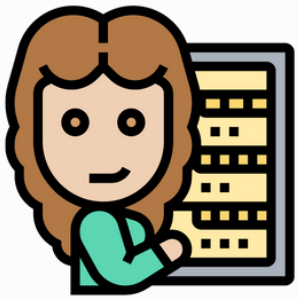
Insinyur Otomasi Uji

Insinyur Jaringan

TECHWORLD WITH NANA

Dimulai sebagai...

Administrator Sistem

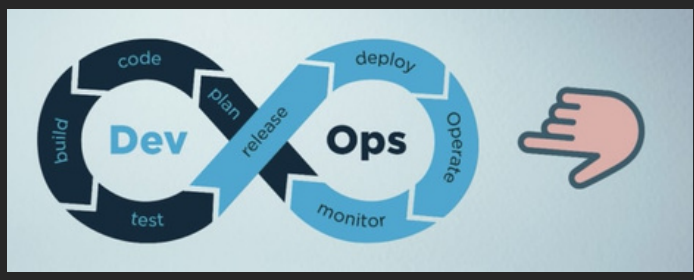


Anda tahu cara mengelola server dan sistem lainnya. Jadi Anda sudah memiliki beberapa keterampilan dalam:

- menyiapkan infrastruktur
- mengonfigurasi dan menyiapkannya untuk penerapan
- bekerja dengan sistem operasi, menginstal dan menjalankan perangkat lunak
- keamanan, konfigurasi jaringan

Beberapa tugas lain yang mungkin Anda lakukan adalah hal-hal seperti:

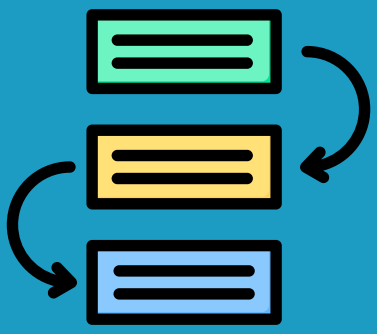
- sistem pemantauan,
- kesehatan, pencadangan dan pemulihan
- bencana, administrasi basis data,
- administrasi jaringan atau
- administrasi keamanan



Jadi Anda sudah punya banyak**keterampilan yang dapat Anda gunakan di sisi penerapan dan operasi DevOps.**

START HERE

Bagian besar**hilang**di sini untuk memulai di DevOps sedang mempelajari**dasar pengembangan perangkat lunak**:



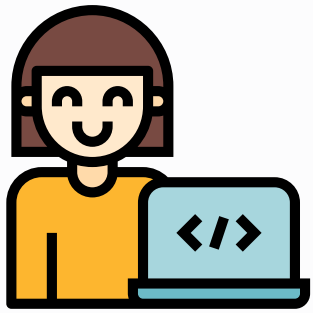
- Memahami alur kerja Git Bagaimana
- pengembang bekerja
- Cara membuat pipa CI/CD

www.techworld-with-nana.com/devops-roadmap

TECHWORLD WITH NANA

Dimulai sebagai...

Pengembang perangkat lunak



Pengembang perangkat lunak

Jika Anda seorang pengembang perangkat lunak, Anda memiliki latar belakang yang cukup bagus, karena Anda sudah mengetahui bagian penting dari DevOps, yaitu alur kerja pengembangan perangkat lunak dan pipa rilis.

Keterampilan pemrograman Anda juga akan sangat membantu dalam menulis skrip otomatis untuk berbagai bagian dari proses pengembangan dan penerapan aplikasi.

START HERE

Tapi kamu**keterampilan yang hilang dalam manajemen server**. Jadi,
Anda harus mulai dengan mempelajari tentang:

- Linux, dasar-dasar OS dan mesin virtual
- membuat dan mengonfigurasi server
- mengonfigurasi keamanan infrastruktur, jaringan, dll.

Dan karena sebagian besar aplikasi modern berjalan di cloud, Anda perlu mempelajari cara melakukan semua ini di infrastruktur cloud.
Jadi itu akan menjadi titik awal Anda saat mempelajari DevOps sebagai pengembang perangkat lunak.

Dan begitu Anda memiliki fondasi itu, Anda dapat membangunnya dengan mempelajari tentang cara kerja container di atas mesin virtual dan cara menjalankan aplikasi dalam container dan cara menjalankan container di platform seperti Kubernetes, dll.

www.techworld-with-nana.com/devops-roadmap

TECHWORLD WITH NANA

Dimulai sebagai...

Insinyur Otomasi Uji

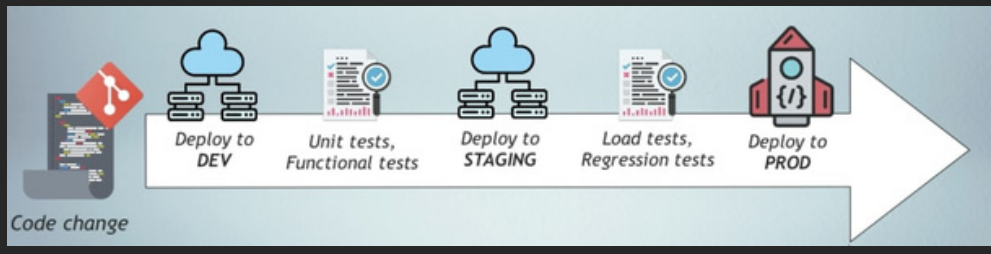
Insinyur Otomasi Uji

Latar belakang umum lainnya yang dimiliki orang ketika bertransisi ke DevOps adalah rekayasa otomasi pengujian.

Di sini Anda mungkin memiliki**sedikit lebih mengejar ketinggalan**untuk melakukan dan lebih banyak keterampilan untuk dipelajari dibandingkan dengan pengembang atau administrator sistem, tetapi Anda pasti dapat menggunakan kembali banyak keterampilan Anda di DevOps.

Anda kemungkinan besar tahu cara kerja pengembang, seperti proses gesit, alur kerja Jira, dan sebagainya. Dan sebagai bagian dari pengetahuan otomatisasi pengujian, Anda memahami cakupan pengujian yang berbeda.

Anda juga mengerti**cara menguji berbagai aspek aplikasi**dan pengetahuan tersebut sangat membantu untuk menyiapkan pipeline CI/CD otomatis, karena untuk mengotomatisasi dan merampingkan penyampaian perubahan aplikasi Anda hingga ke lingkungan produksi, Anda memerlukan pengujian otomatis yang ekstensif:



START HERE

Seperti pengembang, Anda**keterampilan yang hilang dalam manajemen server**. Jadi, Anda harus mulai dengan mempelajari tentang:

- mesin virtual dan dasar-dasar Linux
- membuat dan mengkonfigurasi server
- mengkonfigurasi keamanan infrastruktur, jaringan dll.

www.techworld-with-nana.com/devops-roadmap

TECHWORLD WITH NANA

Dimulai sebagai...

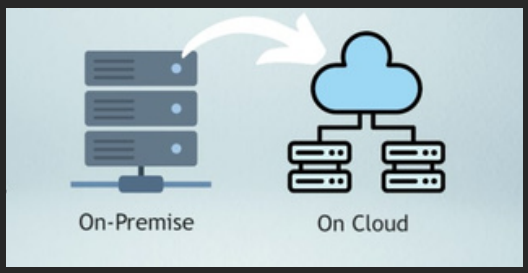
Insinyur Jaringan

Insinyur Jaringan

Latar belakang umum lainnya yang dimiliki orang saat bertransisi ke DevOps adalah jaringan rekayasa. Ini mungkin yang paling jauh dari DevOps dibandingkan dengan tiga lainnya, tetapi Anda masih memiliki beberapa keterampilan yang dapat Anda bawa ke DevOps sebagai insinyur jaringan.

Sebagai insinyur jaringan, Anda tahu cara mengonfigurasi perangkat dan jaringan antar perangkat. Jadi, Anda memiliki pengetahuan berharga dalam mengonfigurasi jaringan untuk infrastruktur di tempat.

Transisi ke Insinyur Jaringan Cloud Tetapi karena sebagian besar perusahaan memindahkan infrastruktur mereka ke cloud, banyak insinyur jaringan beralih ke rekayasa jaringan cloud, mengonfigurasi **maya**rute, switch, dll.



Dengan pengetahuan ini, Anda memiliki keuntungan untuk memahami jaringan dalam wadah dan Kubernetes, yang merupakan cara sebagian besar aplikasi modern berjalan. Jaringan dalam wadah dan K8 cukup rumit, terutama saat kita perlu mengamankan dan memecahkan masalah jaringan tersebut.

Beberapa insinyur jaringan bahkan tahu scripting di bash atau python misalnya, yang merupakan keterampilan lain yang berguna ketika datang ke bagian otomatisasi DevOps.

START HERE

Jadi titik awal yang baik untuk insinyur jaringan adalah beralih ke rekayasa cloud terlebih dahulu, lalu beralih ke wadah dan Kubernetes

www.techworld-with-nana.com/devops-roadmap

Dimulai dengan...

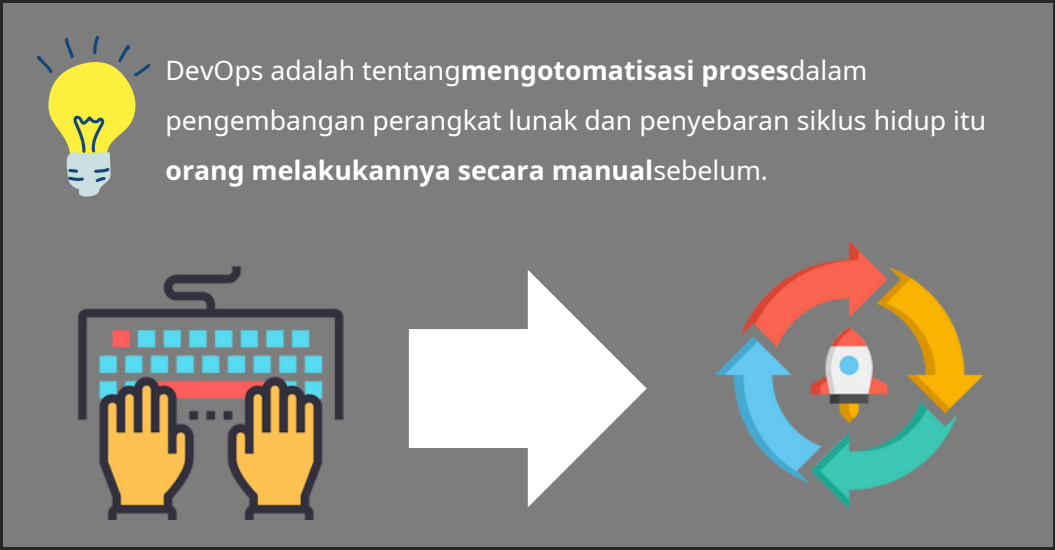
Tidak ada atau sedikit Latar Belakang TI

Latar Belakang Non IT

Beberapa orang ingin masuk ke DevOps **sangat sedikit atau tidak ada latar belakang TI**. Ini berarti mungkin ada beberapa dari Anda yang membaca ini, berpikir untuk masuk ke DevOps tanpa banyak pengetahuan IT sebelumnya dan ingin tahu apa jalan menuju DevOps.

Sekarang ini sangat rumit, karena**DevOps BUKAN profesi tingkat pemula**di dalamnya. Ini bukan hal pertama yang Anda pelajari saat ingin masuk ke bidang TI.

Sekarang kenapa begitu?



Artinya, sebelum Anda mengotomatiskan proses dan tugas yang dilakukan secara manual, Anda **pertama-tama perlu memahami apa proses dan tugas itu di tempat pertama**. Jika Anda tidak memahaminya, Anda tidak akan tahu apa yang Anda otomatisasi atau mengapa Anda membutuhkan DevOps.

TECHWORLD WITH NANA

Dimulai dengan...

Tidak ada atau sedikit Latar Belakang TI

START HERE

1 - Memahami perangkat lunak lengkap siklus hidup pengembangan

Temukan beberapa proyek contoh, tempat Anda membuat aplikasi web super sederhana dan pelajari cara menerapkannya ke server virtual. Dalam proses ini, Anda akan mempelajari langkah-langkah mengembangkan, mengemas, mungkin menguji secara otomatis, lalu menerapkan contoh aplikasi web di server Linux di platform cloud.

Ini akan**mengajari Anda dasar-dasar siklus hidup pengembangan perangkat lunak yang lengkap**, tetapi yang paling penting itu akan membuat Anda memahami setiap langkah dalam alur kerja yang lengkap dan apa yang terjadi di dalamnya.

2 - Bagaimana tim pengembangan perangkat lunak berkolaborasi

Setelah itu, lanjutkan dan tonton beberapa tutorial tentang metode agile dan scrum serta bagaimana tim pengembangan perangkat lunak berkolaborasi dan bekerja dalam proyek TI.

3 - Prasyarat DevOps dan 4 - Keterampilan DevOps

Keterampilan ini sebenarnya akan cukup untuk mulai mempelajari DevOps dengan Bootcamp DevOps kami misalnya, karena di bootcamp kami, Anda benar-benar mempelajari Linux, Git, dan semua alat dasar ini dari awal.

Tetapi sekali lagi Anda perlu memahami alur kerja tersebut terlebih dahulu untuk memahami, mengapa kami menggunakan Git, mengapa kami membutuhkan Jenkins, mengapa kami mempelajari Linux dan skrip, kami menggunakan wadah, dll.

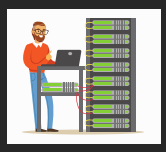
www.techworld-with-nana.com/devops-roadmap

Ringkasan
Peta Jalan DevOps

1 - Mendapatkan prasyarat dengan benar

Langkah pertama adalah memperbaiki prasyarat DevOps. Jadi, bergantung pada latar belakang dan prapengetahuan yang Anda miliki, pertama-tama Anda harus memastikan untuk mendapatkan pengetahuan prasyarat yang hilang.

Jadi sebagai administrator sistem atau insinyur jaringan, pelajari alur kerja pengembangan perangkat lunak. Sebagai pengembang, pelajari dasar-dasar infrastruktur, server virtual, dll.



Tentu saja dengan latar belakang nol TI, Anda harus mendapatkan semua pengetahuan prasyarat ini dari administrasi server hingga pengembangan terlebih dahulu. Jadi Anda memiliki entri yang lebih sulit, tetapi mungkin jika Anda tahu apa yang harus dipelajari.

2 - Cloud, Docker, K8s

Setelah mempelajari prasyarat, Anda sudah dapat memulai dengan keterampilan penting DevOps dalam bekerja dengan container dan alat orkestrasi container. Jadi pada dasarnya mempelajari Docker dan Kubernetes untuk membantu tim Anda menerapkan dan menjalankan aplikasi secara efisien.

Dan karena sebagian besar aplikasi modern dan kluster Kubernetes berjalan di cloud, Anda perlu mempelajari infrastruktur cloud, cara bekerja dengan infrastruktur cloud, cara mengonfigurasinya, cara menskalakannya, dan sebagainya.

3 - Otomasi

Sebagai keterampilan otomasi profesional DevOps adalah salah satu yang paling penting. Dan sebagai inti dari DevOps, belajar membangun pipeline CI/CD adalah keterampilan yang penting.

Terakhir, Anda akan mempelajari cara mengotomatiskan bagian dari proses DevOps lengkap satu per satu menggunakan konsep dan alat yang disebut X sebagai kode: IaC, Konfigurasi sebagai Kode, Keamanan sebagai Kode, Kebijakan sebagai Kode, dan seterusnya, yang pada dasarnya berarti hanya mengotomatiskan semuanya dalam bentuk kode!

4 - Pergi dari sana. Terus belajar

DevOps berkembang dan alat-alat baru sedang dikembangkan setiap saat. Jadi sebagai DevO, Anda harus mempelajari cara mengevaluasi dan menguji banyak alat baru, selalu dengan proses yang sama dan mengotomatiskan yang ada serta membuatnya lebih efisien.

www.techworld-with-nana.com/devops-roadmap

TECHWORLD WITH NANA

Good luck pada Anda **perjalanan DevOps**!

Jika Anda ingin melakukan perjalanan yang rumit**masuk ke DevOps jauh lebih mudah**, Periksa**"TechWorld dengan Nana"**sumber daya

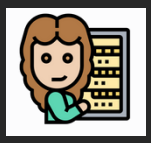
www.techworld-with-nana.com/devops-roadmap

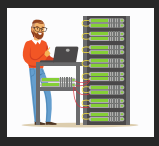
TECHWORLD WITH NANA

https://www.techworld-with-nana.com

Bootcamp DevOps kami

Untuk siapa program pendidikan ini?



Keterangan

Kami mempertimbangkan semua latar belakang yang disebutkan di atas saat membuat bootcamp DevOps kami. Ini adalah program 6 bulan untuk memulai karir Anda sebagai insinyur DevOps. Seluruh bootcamp dibuat dengan fokus memberikan contoh proyek kehidupan nyata, membuat Anda siap kerja dan mampu melakukan tugas DevOps dalam pekerjaan nyata.

Kursus Pemula TI

Karena kami mendapatkan banyak permintaan untuk Bootcamp DevOps kami dari para pemula TI, kami memutuskan untuk membuat kursus prasyarat bootcamp yang lengkap.

Untuk siapa kursus ini?

Orang, yang tidak memiliki atau sedikit latar belakang TI

Topik yang dibahas

- Dapatkan pemahaman yang sangat baik tentang cara kerja pengembangan perangkat lunak lengkap dan alur kerja rilis dengan mengambil setiap peran satu per satu Mampu
- menulis aplikasi web sederhana dengan frontend, backend, dan database dan menyebarkannya di server virtual pada infrastruktur cloud
- Memahami konsep skema versi, sumber terbuka, kerangka kerja, pustaka, alat pengelola paket