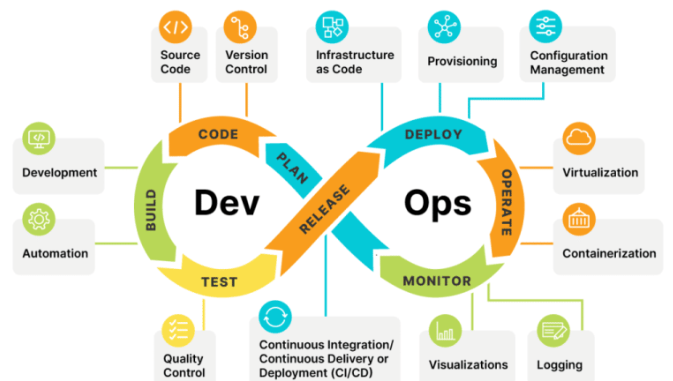
**Teori**

1. Devops ( Devlopment Operation ) adalah Sebuah kultur dimana tim Developer dan tim Operation bekerja sama dalam proses development aplikasi agar lebih cepat dan efisien. dibawah adalah gamabar yang saya dapat dari google untuk ilustrasi dari devops sendiri



Dari gambar diatas kita ketahui untuk bagian devops mencakup itu semua yang dimulai **Plan** yang disusaikena dengan requirement dari business yang dibuat lalu **Code** disini kita menerima SCM yang nanti nya akan dipasang CI/CD agar developer dengan tugas nya membuat code dan push ke repo nya, lalu **Build** disini build menggunakan docker karena kita butuh image untuk dipakai membuat container dengan kubernetes, lalu **Test** disini code akan di test apakah fitur yang ditambahkan oleh developer berjalan atau tidak lalu **Realease** ketika semua dpastikan berjlan akan release, lalu aplikasi akan di **Deploy** ke server yang dibuat dengan menngukan terraform, lalu **Operate** disini kita melakukan hemat bubget dalam pembuatan vm yang biasa saya gunakan aws untuk server, lalu **Monitoring** dengan menggunkan Grapana dan Promotheus

Dengan menggunakan tools diatas yang sabeutkan proses ini akan berulang secara otomatis ketika ada pengembangan/penambahan fitur dengan code yang dibuat lalu di deploy ke server sampai live bisa dinikmati pengguna, hampir semua akan berjalan serba otomatis dengan menggunakan CI/CD jadi gambar diatas akan terulang terus dengan otomatis.

2. Devops digunakan untuk meningkatkan produktivitas dalam mendevelop sebagai penambah nilai business

3. Beberapa Tools yang saya gunakan adalah

* docker yang digunakan untuk membuat image yang akan di push repositoty di docker registry agar kita bisa leluasa memakan di pepmbuatan container
* Orchestra container yaitu kubernetes berfungsi untuk membuat container dengan menggunakan berbagai keunggulan di kubernetes seperti replicas dan juga rollback.
* HELM untuk menyimpan repository yang dibuat yaml yang dibuat agar nanti ingin mendeloy ulang tinggak menngunakan helm install
* Cloud saya menggunakan AWS untuk deploy ke server
* CI/CD dengan jenkins untuk mengotomatiskan step step diatas agar lebih mudah dari developer jika ada perubahan maka akan otomatis terdeploy langsung ketika semua proses dah selesai
* Grapana dan prmotheus untuk monitoring

4. Untuk menjaga dan memelihara server dev, staging, dan production kalau saya lebih ke Monitoring untuk mengecek healty/kesehatan dari sever secara berskala dan juga memantau cpu,ram dan juga disk dan semua ini mesti ada setting ke email

Dan juga backup data agar ketika server rusak kita bisa membuat server baru dengan data yang sudah kita backup dan juga dengan backup data ini berguna untuk versioning agar bisa rollback ke sebelumnya

5. container -> running diatas container engin dan share kernel dengan host

VM -> running diatas hypervisor dan tidak share kernel dengan host

6. Tunnel -> suatu cara agar suatu jaringan dapat berkomunikasi secara private tanpa menggunakan akses public

NAT -> Digunakan untuk suatu jaringan yang bersifat private bisa berkomunikasi dengan internet tanpa terkoneksi secara langsung. Dengan menggunakan NAT, public ip yang dieksposnya hanyalah sesuai dengan host NAT saja.

VPN -> Jaringan virtual yang digunakan untuk berkomunikasi secara private

7. Router -> suatu device yang bekerja pada layer 3 osi layer yang berfungsi untuk meneruskan IP Address

Gateway -> ibarat suatu gerbang atau pintu, yang secara umum terdapat pada suatu jaringan untuk menconvert suatu protocol tertentu atau meloloskan.

Proxy -> gateway antara client dan internet

8. Alamat IP 172.16.0.36/23 adalah alamat IP kelas B yang digunakan untuk mengidentifikasi sebuah host dalam sebuah jaringan. Dari alamat tersebut, bisa dapet perincian seperti ini:

Subnet: 172.16.0.0

Netmask: 255.255.254.0

Broadcast: 172.16.1.255

dalam jaringan dengan alamat IP 172.16.0.36/23, terdapat 510 host yang tersedia dengan

range host yang dapat digunakan dari 172.16.0.1 hingga 172.16.1.254.

9. Untuk kasus ini sebenernya tergantung kebutuhan dari rumah sakit seperti apa tapi dikarenakan rumah sait pasti memiliki lanatai yang banyak dan juga gedung jadi saya merekomendasikan Menggunakan Topologi Mesh karena setiap perangkat bisa terhubung dengan banyak perangkat ini menguntungkan karena bisa berbagi dari dari lantai ke lantai lain jadi terintegrasi

10. menggunakan NAT gateway untuk berkomunikasi dengan internet atau memanafaatkan VPN

11. menggunakan VPN client atau di aws menggunakan site-to-site vpn. Sebagai tambahan jika memerlukan koneksi yang memliki bandwidth tinggi dan latency yg minim bisa menggunakan aws direct connect (cost dan waktu untuk established akan meningkat)

**Untuk Praktek ada di Github**

<https://github.com/Mdzikri/test-panintihub>