

Nama: Perdana Putro Harwanto

Program: Intro To Node JS

SUMMARY MELAKUKAN DEPLOYMENT DENGAN NODE JS Deployment adalah kegiatan yang bertujuan untuk menyebarkan aplikasi atau mengunggah ke internet yang telah dikerjakan oleh para developer. Cara penyebarannya pun sangat beragam, tergantung dari jenis aplikasinya. Jika Anda pilih aplikasi Web, maka Anda akan di hosting pada server. Sedangkan jika aplikasi mobile, akan terdapat dua deployment. Yang pertama adalah deployment untuk aplikasi ke Playstore atau Appstore, dan yang kedua adalah deployment API (backend) ke server.

Heroku. Heroku adalah web hosting berbasis cloud digunakan untuk melakukan pengembangan atau tester aplikasi dengan support banyak bahasa pemrograman. sehingga bagi anda yang ingin melakukan deploy aplikasi ke heroku cukup hanya dengan melakukan konfigurasi aplikasi yang ingin di deploy dan menyediakan platform yang memungkinkan pelanggan untuk mengembangkan, menjalankan, dan mengelola aplikasi tanpa kompleksitas membangun dan memelihara infrastruktur yang biasanya terkait dengan pengembangan dan peluncuran aplikasi. Environment dan konfigurasi untuk melakukan deployment menggunakan heroku adalah yang pertama membuat akun heroku terlebih dahulu pada situs resmi heroku <https://www.heroku.com>, lalu setelah itu menginstall heroku cli pada local komputer, dan kita hubungkan project kita ke heroku dengan menggunakan git. setelah itu kita dapat melakukan deployment jika sudah selesai menghubungkan project ke heroku dengan menggunakan tools heroku cli tadi. Untuk deployment sendiri kita menggunakan perintah-perintah git pertama lakukan initialization git 'git init', kemudian pada terminal lakukan login heroku dengan perintah 'heroku login' lalu masukkan email dan password akun heroku yang sudah dibuat, setelah itu buat instance heroku dengan perintah 'heroku create <name-app>' maka setelah itu secara otomatis heroku akan menambahkan git remote url reponya untuk memastikan itu jalankan 'git remote -v' dan lihat apakah remote url heroku sudah ada. lalu tahap terakhir lakukan perintah git pada umumnya untuk push pertama 'git add .' lalu 'git commit -m "pesan commit"' dan terakhir project ke heroku dengan 'git push heroku master' maka deployment selesai dan aplikasi sudah diunggah ke internet.

AWS Amazon Web Service, didalam aws terdapat beberapa macam service computing salah satunya adalah elastic compute cloud, didalamnya kita dapat menyimpan sebuah instans atau sebuah virtual machine dimana kita dapat memiliki sebuah server yang layaknya server sungguhan didalamnya kita dapat menyimpan resource dan juga database, kita kita menggunakan service EC2 software yang

dibangun dapat memiliki tingkat scalability yang tinggi, karena kita dapat menentukan atau mengkonfigurasi ram, cpu dan storage di dalam virtual server sesuai dengan kebutuhan, serta kelebihan lainnya kita hanya akan mengeluarkan biaya sesuai dengan apa yang kita butuhkan. mengatur environment aws. Pertama kita perlu membuat akun aws didalam website resmi aws <https://aws.amazon.com/>, setelah membuat akun lakukan konfigurasi untuk membuat instans, untuk membuat instans pertama menentukan machine image, setelah memilih machine image, kita tentukan instance type, dan klik launch maka instans akan terbuat. setelah selesai membuat instans EC2 buat konfigurasi untuk connect dengan ssh agar supaya instans dapat dioperasikan dengan terminal.

setelah selesai mengkoneksikan instans dengan ssh dan dapat diakses melalui terminal, untuk melakukan deploy project node js pertama install nvm pada instans melalui terminal yang sudah dihubungkan dengan EC2 atau instans agar dapat melakukan perintah-perintah node. setelah itu install node js, install git untuk melakukan clone pada repository project yang akan di deploy, setelah melakukan clone repository masuk ke dalam folder repo dan jalankan aplikasi. lalu pada instans lakukan perubahan pada security groups untuk mengganti port agar dapat diakses oleh public dan setelah itu maka aplikasi sudah ter deploy dan dapat diakses oleh public menggunakan public IP DNS yang diberikan oleh instans. Agar aplikasi dapat terus running saat kita mematikan server pada aplikasi dapat menggunakan package pm2, setelah itu jalan kan server kembali dengan perintah pm2 start "fileServerName.js", lalu jalankan perintah pm2 startup agar aplikasi dapat berjalan secara otomatis ketika instans dijalankan, dan simpan perubahan menggunakan pm2 -save. mengkonfigurasi port agar dapat menggunakan port public, dengan menginstall nginx (engine next) pada instans, lalu lakukan perubahan kembali pada security groups, dan tambah inbound rules dengan port 80 port default dari nginx, setelah itu masukkan project ke dalam port 80 yaitu port nginx dengan cara lakukan perubahan pada konfigurasi file nginx.conf. pada location tambahkan proxy_pass http://127.0.0.1:{{portPadaProjectYangDitentukan}}; proxy_http_version 1.1; proxy_set_header Upgrade \$http_upgrade; proxy_set_header Connection 'upgrade'; proxy_set_header Host \$host; proxy_cache_bypass \$http_upgrade; setelah itu save file dan restart nginx terakhir kita konfigurasi nginx agar otomatis running ketika instans dijalankan atau direstart dengan perintah sudo chkconfig nginx on

Google Cloud Platform, Google Cloud merupakan salah satu provider layanan Cloud Computing terbesar di dunia yang mengandalkan infrastruktur dari Google. didalam Google Cloud platform juga terdapat layanan IaaS yang memiliki kegunaan sama seperti Amazon EC2 yang bernama compute engine dimana berbagai kebutuhan

pengembang bisa dilakukan dengan mesin ini karena memang layanan ini menawarkan ketangguhan penyimpanan disk dan kinerja yang konsisten.

Mengatur environment Google Cloud Platform untuk mendeploy aplikasi node js yang sudah dibuat. pertama adalah kita sign in ke gcp menggunakan akun email google, lalu masuk ke menu console untuk masuk ke dalam dashboard dari GCP dan pilih new project. setelah project selesai dibuat, lakukan pembuatan instance baru pada menu compute engine, untuk melakukan konfigurasi nantinya, dapat menggunakan ssh dengan memilih menu ssh pada website, dan terminal instance engine siap digunakan.

Mendeploy aplikasi nodejs menggunakan instance vm engine dengan GCP yang telah dibuat. Pertama install NodeJS terlebih dahulu, setelah itu install git untuk melakukan clone pada repository project yang akan di deploy, setelah melakukan clone repository masuk ke dalam folder repo dan jalankan aplikasi. lalu pada instance lakukan perubahan pada VPC network dan pilih firewall rules untuk mengganti port agar dapat diakses oleh public dan setelah itu maka aplikasi sudah terdeploy dan dapat diakses oleh public menggunakan public IP yang diberikan oleh instance. Agar aplikasi dapat terus running saat kita mematikan server pada aplikasi dapat menggunakan package pm2, setelah itu jalan kan server kembali dengan perintah pm2 start "fileServerName.js", lalu jalankan perintah pm2 startup agar aplikasi dapat berjalan secara otomatis ketika instans dijalankan, dan simpan perubahan menggunakan pm2 -save. mengkonfigurasi port agar dapat menggunakan port public, dengan menginstall nginx (engine next) pada instans, lalu lakukan perubahan kembali pada security groups, dan tambah inbound rules dengan port 80 port default dari nginx, setelah itu masukkan project kedalam port 80 yaitu port nginx dengan cara lakukan perubahan pada konfigurasi file nginx.conf. pada location tambahkan proxy_pass http://127.0.0.1:{{portPada project yang ditentukan}}; proxy_http_version 1.1; proxy_set_header Upgrade \$http_upgrade; proxy_set_header Connection 'upgrade'; proxy_set_header Host \$host; proxy_cache_bypass \$http_upgrade; setelah itu save file dan restart nginx terakhir kita konfigurasi nginx agar otomatis running ketika instans dijalankan atau direstart dengan perintah sudo chkconfig nginx on