

Bangun otomatisasi

Otomatisasi build adalah proses mengotomatiskan pembuatan build perangkat lunak dan proses terkait termasuk: mengkompilasi kode sumber komputer menjadi kode biner , mengemas kode biner , dan menjalankan pengujian otomatis .

Ringkasan

Secara historis, otomatisasi pembangunan dicapai melalui makefiles . Saat ini, ada dua kategori umum alat: ^[1]

Utilitas otomatisasi pembangunan

Ini termasuk utilitas seperti Make ,

Rake , CMake , MSBuild , Ant ,

Maven atau Gradle (Java) dll .

Tujuan utamanya adalah

menghasilkan artefak bangunan

melalui aktivitas seperti

mengompilasi dan menautkan kode sumber.

Server otomatisasi pembangunan

Ini adalah alat umum berbasis web yang menjalankan utilitas otomasi pembangunan secara terjadwal atau dipicu; server integrasi berkelanjutan adalah jenis server otomatisasi pembangunan.

Tergantung pada tingkat otomatisasi, klasifikasi berikut dimungkinkan:

- Makefile - tingkat
 - Alat berbasis make
 - Alat yang tidak berbasis merek
- Buat alat pembuatan skrip (atau Makefile).
- Alat integrasi berkelanjutan

- Alat manajemen konfigurasi
- Alat pembuat meta atau pengelola paket
- Lainnya

Daftar perangkat lunak untuk masing-masing dapat ditemukan di daftar perangkat lunak otomatisasi pembangunan .

Utilitas otomatisasi pembangunan

Utilitas build-automation memungkinkan otomatisasi tugas yang sederhana dan dapat diulang. Saat menggunakan alat ini, alat ini akan menghitung cara mencapai tujuan dengan menjalankan tugas dalam urutan yang benar dan spesifik serta menjalankan setiap tugas. Dua cara membangun alat berbeda adalah berorientasi pada tugas vs. berorientasi pada produk. Alat berorientasi tugas menggambarkan ketergantungan jaringan dalam kaitannya dengan serangkaian tugas tertentu dan alat berorientasi produk menggambarkan berbagai hal dalam kaitannya dengan produk yang dihasilkannya. [2]

Server otomatisasi pembangunan

Meskipun server pembangunan sudah ada jauh sebelum server integrasi berkelanjutan, server tersebut umumnya identik dengan server integrasi berkelanjutan, namun server

pembangunan juga dapat dimasukkan ke dalam alat ARA atau alat ALM .

Jenis server

- **Otomatisasi sesuai permintaan** seperti pengguna menjalankan skrip di baris perintah
- **Otomatisasi terjadwal** seperti server integrasi berkelanjutan yang menjalankan build setiap malam
- **Otomatisasi yang dipicu** seperti server integrasi berkelanjutan yang menjalankan build pada setiap komitmen pada sistem kontrol versi .

Otomatisasi pembangunan terdistribusi

Otomatisasi dicapai melalui penggunaan kumpulan kompilasi untuk kompilasi terdistribusi atau pelaksanaan langkah utilitas. ^[3] Proses pembangunan terdistribusi harus memiliki kecerdasan mesin untuk memahami ketergantungan kode sumber untuk menjalankan pembangunan terdistribusi.

Hubungan dengan pengiriman berkelanjutan dan integrasi berkelanjutan

Otomatisasi build dianggap sebagai langkah pertama menuju penerapan budaya pengiriman berkelanjutan dan DevOps . Otomatisasi build yang dikombinasikan dengan integrasi berkelanjutan , penerapan , otomatisasi rilis aplikasi , dan banyak proses lainnya membantu memajukan organisasi dalam menetapkan praktik terbaik pengiriman perangkat lunak. [4].

Keuntungan

Keuntungan otomatisasi pembangunan pada proyek pengembangan perangkat lunak meliputi

- Prakondisi yang diperlukan untuk integrasi berkelanjutan dan pengujian berkelanjutan
- Meningkatkan kualitas produk
- Mempercepat kompilasi dan pemrosesan tautan

- Hilangkan tugas-tugas yang berlebihan
- Minimalkan "bangunan buruk"
- Hilangkan ketergantungan pada personel kunci
- Miliki riwayat pembuatan dan rilis untuk menyelidiki masalah
- Menghemat waktu dan uang - karena alasan yang tercantum di atas. [5].

Lihat juga

- Otomatisasi rilis aplikasi (ARA)
- Otomatisasi konfigurasi berkelanjutan (CCA)

- Integrasi berkelanjutan (CI)
- Pengiriman berkelanjutan (CD)
- Pengujian berkelanjutan
- DevOps
- Daftar perangkat lunak otomatisasi pembangunan
- Rekayasa keluarga produk
- Rekayasa rilis (RE)
- Manajemen konfigurasi perangkat lunak (SCM)
- Pengujian satuan

Referensi

1. *Ceruzzi, Paul E. (2003). Sejarah komputasi modern (<https://archive.or>*

g/details/historyofmodern00ceru_0)
. Pers MIT. ISBN 978-0262532037.

2. *Clark, Mike (2004). Otomatisasi Proyek Pragmatis: Cara Membangun, Menerapkan, dan Memantau Aplikasi Java . Pemrogram Pragmatis. ISBN 978-0974514031.*
3. *Enos, Joe (2013). "Pembuatan Otomatis: Kunci Konsistensi" (<http://www.infoq.com/articles/Automated-Builds>) . InfoQ . C4Media Inc. Diakses pada 16 September 2015 . (<http://www.infoq.com/articles/Automated-Builds>)*
4. *Basan, Shmuel; Bellagio, David E. (2011). Manajemen Item Kerja dengan IBM Rational ClearQuest dan Jazz:*

*Panduan penyesuaian . IBM Pers.
ISBN 978-0137001798.*

5. *"Salinan yang diarsipkan" (https://web.archive.org/web/20081123044304/http://www.denverjug.org/meetings/files/200410_automation.pdf) (PDF) .
Diarsipkan dari asli (http://www.denverjug.org/meetings/files/200410_automation.pdf) (PDF) pada 23-11-2008 .
Diakses tanggal 19-09-2008 . (https://web.archive.org/web/20081123044304/http://www.denverjug.org/meetings/files/200410_automation.pdf) (http://www.denverjug.org/meetings/files/200410_automation.pdf) (http://www.denverjug.org/meetings/files/200410_automation.pdf)*

Retrieved from

["https://en.wikipedia.org/w/index.php?"](https://en.wikipedia.org/w/index.php?)

[title=Build_automation&oldid=1220114444"](#)

WIKIPEDIA

This page was last edited on 21 April 2024, at 22:10 (UTC). •

Konten tersedia di bawah [CC BY-SA 4.0](#) kecuali dinyatakan lain.