Continuous delivery & Live monitoring

George Popa

ThoughtWorks®

TABLE OF CONTENTS

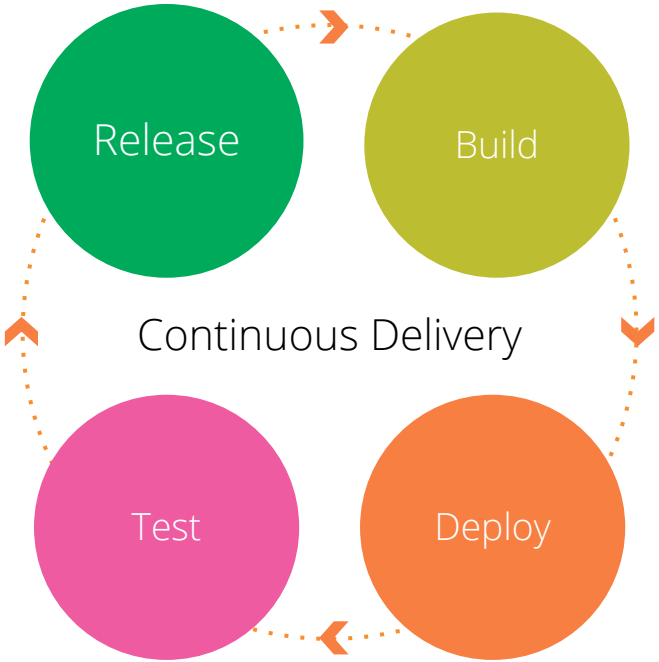
Continuous delivery

Definitie:

Tehnica software de lucru in cicluri scurte, cu scopul de a avea in orice moment o aplicatie stabila, pregatita pentru deployment in productie.

Caracteristici:

- Se dezvolta aplicatia, se testeaza si se publica in productie in ritm crescut;
- Capacitate de a lucra pe mai multe “ramuri” (branches) (functionalitati noi sau bug fixes), izolate si derivate din trunk ;
- Optiunea de a publica in productie numai anumite modificari, intr-un timp scurt.



Continuous delivery - build

- Dezvoltarea aplicatiei se realizeaza in cicluri scurte, de ordinul zilelor - saptamani, la sfarsitul carora se poate efectua un deployment sau un release in productie.
- Se folosesc feature branches pentru fiecare noua functionalitate sau bug fix, divergente din trunk, astfel incat acestea sunt izolate, si se pot testa si accepta individual.
- La nivelul de build se executa testele unitare, de validare a codului, ideal printr-un code coverage cat mai apropiat de 100%.
- Procesul de testare unitara poate fi automat prin solutii de continuous integration, precum Jenkins CI.

Continuous delivery

Continuous deployment - automatizeaza procesul de deployment, astfel incat fiecare membru al echipei, imediat ce a terminat munca in legatura cu o noua functionalitate sau bug fix, declanseaza deployment pe o masina de acceptanta, urmand se fie executate teste de integrare, sau teste exploratorii, inainte de a fi publicate in productie.

Continuous delivery - build

Deployment - instalarea, configurarea și controlul unei aplicații software pe o mașină țintă, locală sau la distanță, într-o manieră similară cu un instrument de build, sau bazându-se pe un asemenea instrument.

Necesitatea sistemelor de deployment:

- Automatizează instalarea și configurarea aplicației;
- Permite accesul la distanță pe mediul de lucru;
- Asigură continuarea funcționalității (minimal downtime);
- Permite revenirea la versiuni anterioare a aplicației (capistrano);
- Minimizarea impactului asupra utilizatorilor, menținând serverele în stare de funcționare în timpul deploymentului (capistrano).

Exemple: capistrano, chef, Docker

Scenarii de deployment

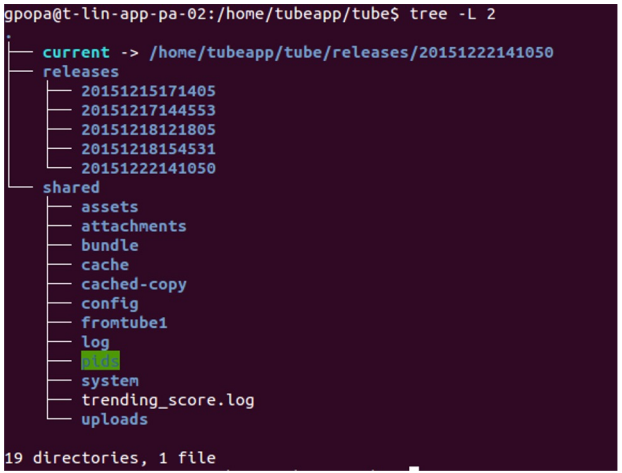
- Continuous delivery (CD) - publicare si testare automata in urma modificarii codului sursa (Exemplu: Jenkins CI)
- Nightly builds - publicare si testare periodica a aplicatiei
- Actualizarea dependentelor
- Push-button builds
- Rollback deployment

Instrumente de deployment

- Capistrano - http://capistranorb.com
- Chef - https://www.chef.io/products/chef-infra
- Octopus Deploy - https://octopus.com
- Docker - https://www.docker.com

Capistrano

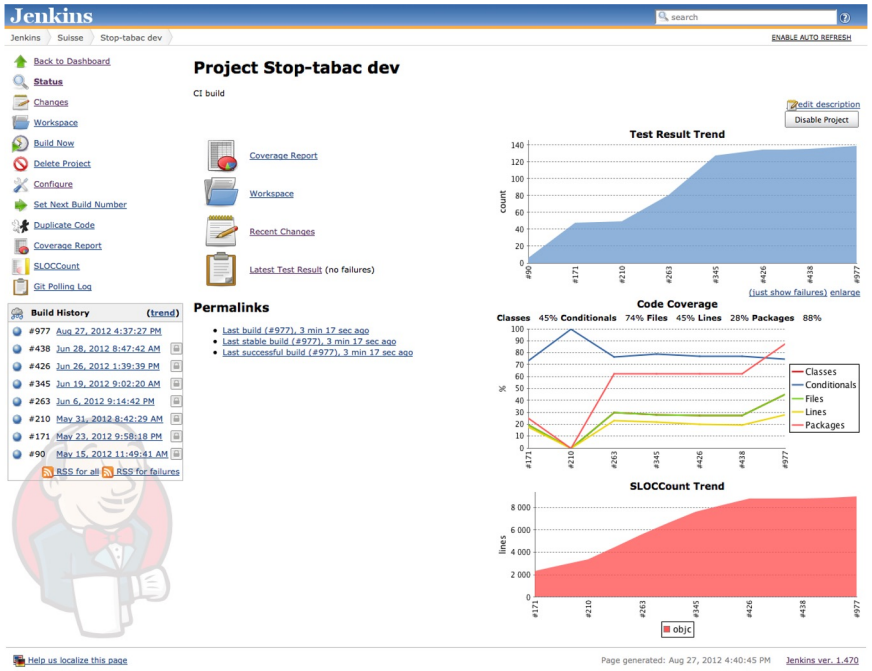
- Capistrano RB este un instrument universal de execuție a scripturilor pe un server la distanță scris în limbajul Ruby, utilizând protocolul SSH.
- Scripturile sunt deregulă task-uri de tip rake, ce urmează a fi executate pe mașini ținta target, însă se pot executa remote orice fel de comenzi de sistem sau scripturi bash.
- Aceleași scripturi vor fi executate parametrizat, în funcție de mașina țintă (mediu de dezvoltare, testare sau producție).



Continuous delivery - testing

Jenkins CI:

- Reproduce un mediu de testare automata (teste unitare, de integrare, de performanta);
- Se declanseaza manual, programat periodic, sau la schimbari in codul sursa;
- Publica rezultatele testelor, code coverage), semnaleaza erori.



Continuous delivery - release

Instrumente de deployment se ocupa de:

- Deployment in mediul de productie;
- Compunere “Release notes” automat, pe baza noilor functionalitati / bug fixes incluse;
- Calculeaza numerele de versiune;
- Trimite notificari.

Live monitoring software

Sisteme software ce analizează condițiile de execuție ale aplicațiilor, atât în medii de dezvoltare sau testare, cât mai ales în producție.

- New Relic - http://newrelic.com
- LiveAction - http://liveaction.com

Caracteristici:

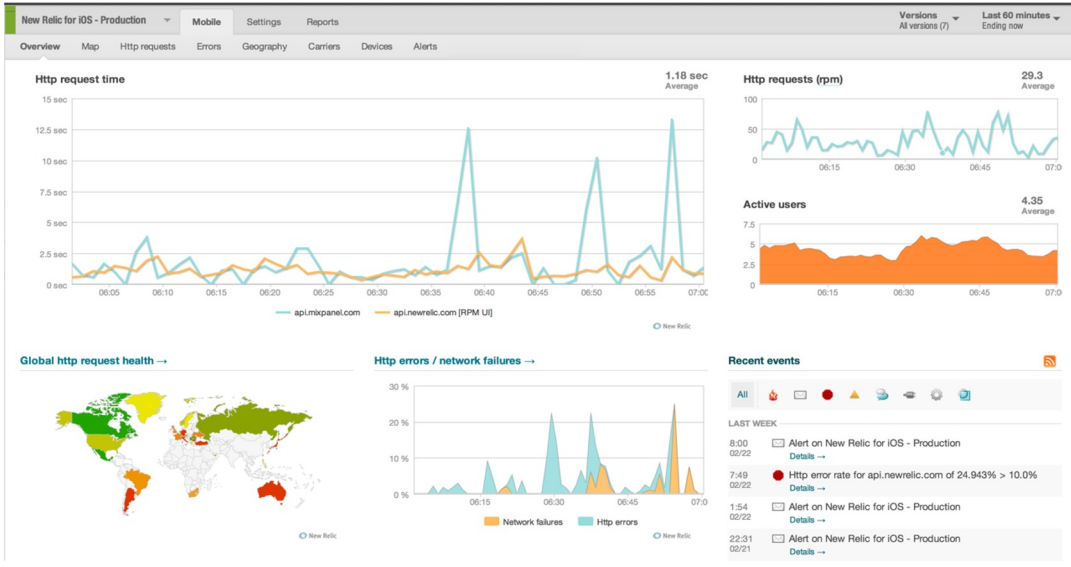
- Instrumentează aplicațiile țintă, prin programe / plug-ins adiționale cu overhead minim.
- Colectează date de intrare, ieșire sau performață ale aplicațiilor, punându-le la dispoziția utilizatorilor într-o interfață prietenoasă, de regulă configurabilă.

New relic monitoring

Gestioneaza si monitorizează:

- Aplicații web;
- Servere dedicate (application server, baze de date, ...etc.);
- Sisteme hardware

...prin intermediul unor plugin-uri ce se instalează pe mașina țintă, sau prin requests periodice (ping, polling) către aceasta mașină.



New relic monitoring

Sumarizeaza informatii despre executia aplicatiilor (similar cu profilerele sau instrumentele de stress testing):

- Timpii medii de executie, timpi de acces, timpi de raspuns;
- Nivelul de incarcare a masinilor tinta (CPU, memorie, disk);
- Traficul pe internet, localizare geografica a clientilor;
- Erori de executie, cauza erorilor identificate in cod.