

ПОБУДОВА CI/CD ПРОЦЕСУ РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ З ВИКОРИСТАННЯМ TEAMCITY ТА GO CD

© Дубленич Р., Струк Є., 2017

Описано реалізацію процесу неперервної інтеграції та процесу неперервного розгортання з використанням утиліт TeamCity та GoCD.

Ключові слова: автоматизація, неперервна інтеграція, неперервне розгортання.

Described an implementation of Continuous Integration and Continuous Deployment processes using TeamCity and GoCD.

Key words: a Continuous Deployment, Continuous Integration, DevOps, TeamCity, GoCD, Continuous Delivery, deployment, automation.

Вступ

За останні роки процедура розроблення програмного забезпечення набуває все більшої складності. Це зумовлено зростанням комплексності функціональних та нефункціональних вимог до програмного продукту. Саме тому виникла необхідність у застосуванні нових та гнучких методологій розроблення ПЗ. Одним з таких підходів є DevOps.

DevOps (акронім від англ. Development і operations) – методологія розроблення програмного забезпечення, скерована на активну взаємодію та інтеграцію фахівців з розроблення та фахівців з інформаційно-технологічного обслуговування. Базується на ідеї про тісну взаємозалежність розроблення та експлуатації програмного забезпечення для того, щоб допомагати організаціям швидше створювати і оновлювати програмні продукти і сервіси.

Організаціям, яким необхідні часті випуски програмного забезпечення, може знадобитися DevOps підхід. Денний цикл релізів може бути набагато інтенсивнішим у організацій, які випускають кілька різноспрямованих програмних продуктів.

Методологія фокусується на стандартизації середовищ розроблення та тестування з метою сприяння швидкому випуску релізів. В ідеалі системи автоматизації збирання і випуску повинні бути доступні всім розробникам в будь-якому оточенні, і розробники повинні мати контроль над оточенням, а інформаційно-технологічна інфраструктура повинна ставати більш сфокусованою на програмному продукті.

Завдання DevOps – зробити процес розроблення і постачання програмного забезпечення узгодженим з експлуатацією, часто ці завдання вирішують за підтримки автоматичних засобів.

Термін "DevOps" популяризував серією зустрічей "DevOps Days", які почали проходити від 2009 р. у Бельгії [1].

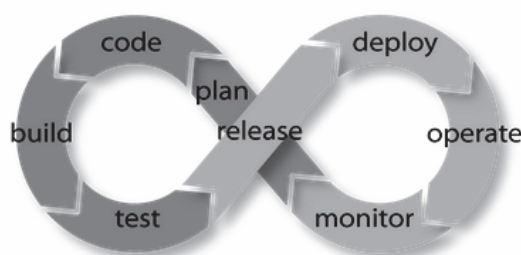


Рис. 1. Завдання, що вирішуються DevOps підходом

Одним з процесів, які мають бути реалізовані за підходом DevOps, є:

- неперервна інтеграція (англ. Continuous Integration) – це практика розроблення програмного забезпечення, яка полягає у виконанні частих автоматизованих складань проекту для якнайшвидшого виявлення та вирішення інтеграційних проблем. У звичайному проекті, де над різними частинами системи розробники працюють незалежно, стадія інтеграції є завершальною. Вона може непередбачувано затримати закінчення робіт. Перехід до неперервної (постійної) інтеграції дає змогу знизити трудомісткість інтеграції і зробити її передбачуванішою внаслідок найбільш раннього виявлення та усунення помилок і суперечностей [2].

- безперервна поставка (CD) являє собою програмний інженерний підхід, за яким команди виробляють програмне забезпечення в коротких циклах, гарантуючи, що програмне забезпечення може бути надійно випущене в будь-який час. Вона спрямована на створення, тестування і випуск програмного забезпечення швидше і частіше. За таким підходом можна знизити витрати, час і ризик доставки зміни, що надає більше додаткових оновлень для додатків у виробництві. Прямолінійний і повторюваний процес розгортання має важливе значення для безперервної доставки [3].

Постановка задачі

Як приклад наведено простий проект клієнт-серверної архітектури, що зображена на рис. 2.

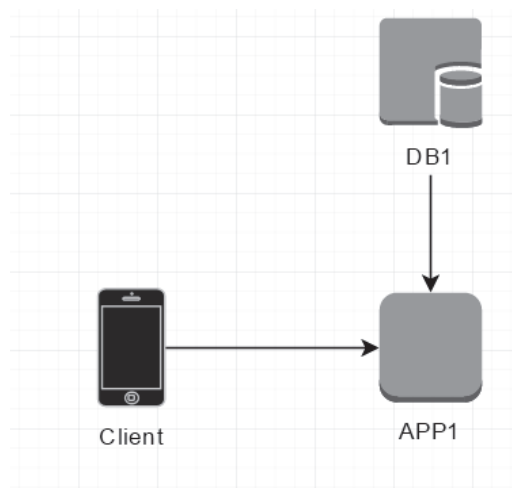


Рис. 2. Архітектура проекту

Основною задачею є побудувати CI/CD процес для такого проекту з урахуванням таких вимог:

- Побудований процес повинен бути простим у своїй реалізації;
- Необхідно реалізувати 3 середовища: Dev (для розробки), Stage (для тестування), Prod (для безпосереднього використання користувачами);
- Вихідні коди програмного продукту повинні компілюватись тільки 1 раз на 1 зміну в коді.

Завдання дослідження

1. Ознайомитись із загальною концепцією DevOps підходу до розроблення та тестування програмного забезпечення.

2. Реалізувати CI/CD процес з використанням TeamCity та GoCD.

Як CI систему було обрано продукт TeamCity від JetBrains. Такий вибір зумовлений:

- умовною безкоштовністю продукту;
- простотою налаштування та використання;
- широким функціоналом;

Як CD-утиліту було обрано доволі новий продукт – GoCD від ThoughtWorks Inc, бо:

- GoCD дає змогу реалізувати концепцію конвеєра;
- з відкритим вихідним кодом;
- працює на ОС Windows, GNU/Linux, OS X;

Continuous Delivery (CD) частина процесу в контексті DevOps підходу

Головними вимогами до процесу неперервного розгортання аплікації є:

• Вихідний код повинен компілюватись і пакуватись тільки 1 раз на 1 версію вихідного коду;

- Зібраний код повинен розгортатись на усіх середовищах тестування та впровадження;
- Конфігурація збірки чи середовища повинна зберігатись в системі контролю версій (SCM);
- Середовища, призначені для розробників та тестувальників повинні бути розділеними.

Отже, потрібно реалізувати 3 середовища:

- Dev – середовище, в якому розробники будуть тестувати свій код;
- Stage – призначене для фінального опробування продукту тестувальниками;
- Prod – середовище, яким користуються безпосередньо користувачі програмного продукту.

Усі 3 середовища абсолютно ідентичні і розгортаються системою керування конфігурацією (Configuration Management system). Середовища зображено на рис. 3.

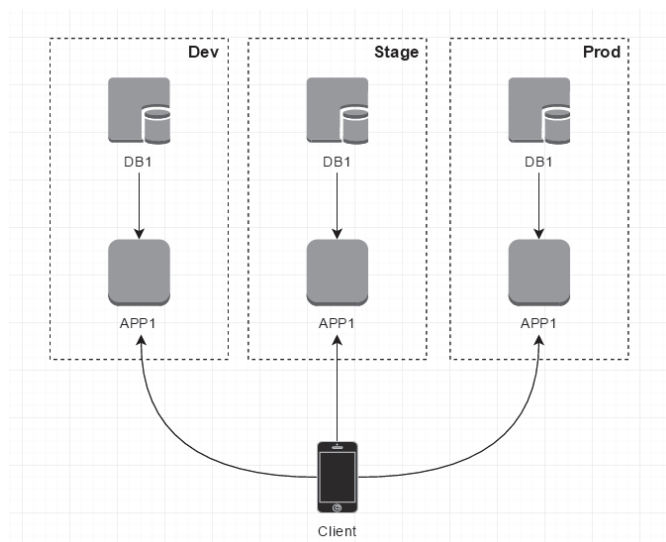


Рис. 3. Середовища впровадження програмного продукту

Потім потрібно налаштувати GoCD. Приклад конвеєра, який б задовільнив вимоги, наведено на рис. 4 та 5.

З поданих рисунків легко бачити, що кожна збірка програмного продукту як мінімум проходить процес розгортання на середовищі Dev. Розгортання відбувається у автоматичному режимі одразу після завершення процесу компілювання та пакування в TeamCity.

Окрім цього, варто звернути увагу на те, що не кожна збірка буде розгорнута на середовищі Prod, а, відповідно, не потрапить у публічне використання користувачами розроблюваного продукту.

На прикладі збірки 0.933 – збірка розгорнута на середовищі Dev, успішно апробована розробниками, після чого розгорнута на середовищі Stage, однак під час ручного тестування було виявлено критичну помилку в роботі аплікації, і тому збірка не буда розгорнута на середовищі Prod для кінцевих користувачів.

Знайдену помилку було виправлено у версії 0.934, і збірка успішно пройшла усі етапи тестування, і зараз знаходиться в розгорнутому і робочому стані на середовищі Prod. Користувачі програмного продукту використовують саме цю версію у цей момент.

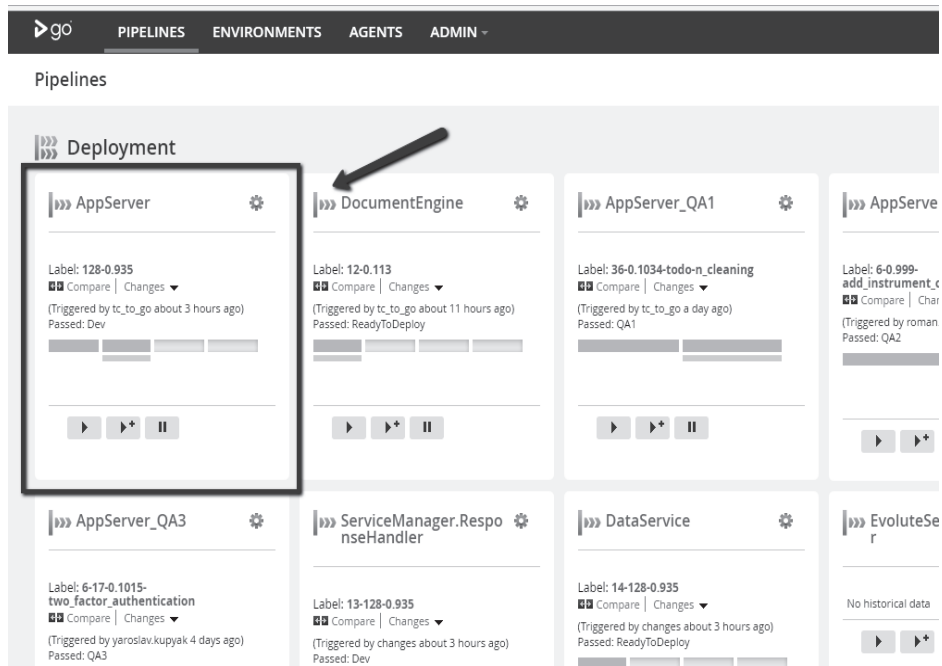


Рис.4 Загальний вигляд конвеєра в GoCD

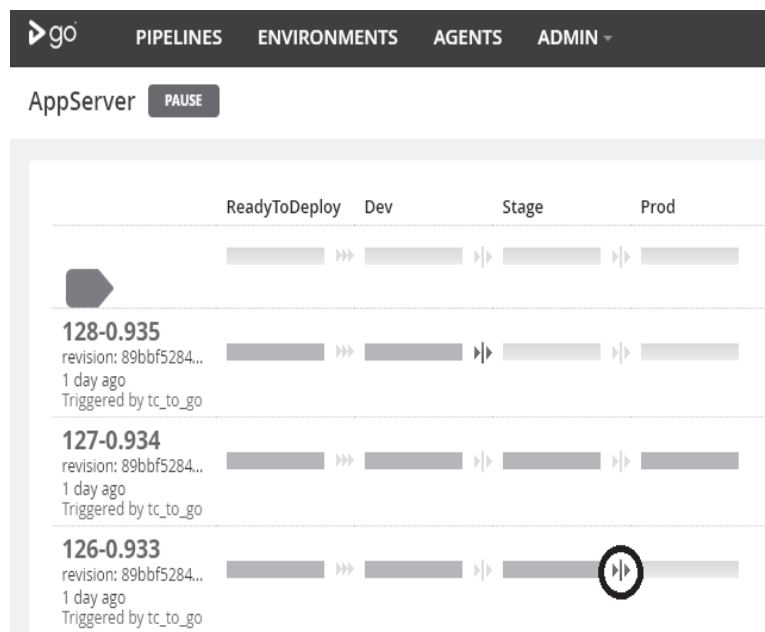


Рис. 5. Розгорнутий вигляд конвеєра в GoCD

Однак, розроблення додатку триває й надалі, і вже з'явилась версія 0.935. Збірка успішно розгорнута на середовищі Dev і зараз проходить базове тестування розробниками продукту. В разі успішного опробування збірка потрапить на середовище Stage для фінального тестування клієнтом та/або тестувальниками.

“Просування” збірок на наступне середовище відбувається одним натисканням на клавішу миші. На рис. 5 цю кнопку обведено. Увесь процес розгортання відбувається за допомогою інсталяційних сценаріїв, написаних мовами PowerShell та Bash. Написані сценарії забезпечують виконання таких операцій, як: зупинка та запуск служб ОС, копіювання файлів, редагування конфігурації, резервування, нотифікація та ін., необхідних для розгортання програмного продукту. Крім цього, інсталяційні сценарії використовують файл з налаштуваннями аплікації для якогось

певного середовища, в якому відбувається розгортання. Тобто збірка не містить жодної конфігурації, що залежить від середовища. Усі необхідні дані отримуються CD-сценаріями під час розгортання.

Перевагою власних сценаріїв CD-процесу є можливість повністю керувати процесом доставки та інсталяції аплікацій.

Continuous Integration (CI) частина процесу в контексті DevOps підходу

Усі збірки проходять процес компіляції та пакування, а також набір базових тестів. Це відбувається за допомогою TeamCity. Загальний вигляд проекту в TeamCity зображено на рис. 6, а конфігурація проекту – на рис. 7.

Build Configuration	Status	Artifacts	Agent	Time
Build_and_Pack_feature	Pending (23)			
todo-n_cleaning #0.1034-todo-n_cleaning	Success	Artifacts	nazariy (2)	one day ago (1m:40s)
add_liquidity_AS-324 #0.1032-add_liquidity_AS-324	Success	Artifacts	nazariy (2)	one day ago (1m:46s)
two_factor_authentication #0.1015-two_factor_authentication	Success	Artifacts	No changes	4 days ago (1m:34s)
todo-n1_cleaning #0.1014-todo-n1_cleaning	Success	Artifacts	Changes (4)	4 days ago (1m:03s)
<default> #0.966	Success	Artifacts	nazariy (2)	11 days ago (57s)
Build_and_Pack_master	Success			
#0.935	Success	Artifacts	neuren (2)	3 hours ago (1m:03s)

Рис. 6. Загальний вигляд проекту в TeamCity

Build Step	Parameters Description	Edit
Nuget	NuGet Installer Solution: integration/AppServer.sln Execute: If all previous steps finished successfully	Edit
Build	Visual Studio (sln) Build file path: integration/AppServer.sln Targets: Build Configuration: Release Platform: x64 Execute: If all previous steps finished successfully	Edit
Trigger Go.CD	PowerShell PowerShell x86 scripts_cdtrigger.ps1 Execute: If all previous steps finished successfully	Edit

Рис. 7. Налаштування процесу компілювання програмного продукту в TeamCity

Як видно з поданих вище рис. 6 та 7, збирають пакети з усіх гілок GIT репозиторія. Сам процес складається з 3-х кроків:

- Nuget – на цьому кроці завантажуються залежності програмного продукту зі сховища пакетів NuGet;
- Build – безпосередньо компіляція та базове автоматичне тестування коду програмного продукту. Здійснюється за допомогою утиліти MSBuild;
- Trigger Go CD – на цьому кроці TeamCity створює новий конвеєр в GoCD, і таким чином щойно створена збірка з'явиться в GoCD та буде автоматично розгорнута у Dev-середовищі. Як спосіб реалізації взаємодії між TeamCity та GoCD було використано API функції GoCD.

Окрім цього, кожен збірку можна завантажити і використати для ручного розгортання та запуску, наприклад, на локальному комп'ютері.

Загальний вигляд реалізації підходу

Як видно із рис. 8, процес неперервної інтеграції та неперервного розгортання здійснюють так:

- вихідний код програмного продукту зберігається у SCM GIT. Цей код потрапляє до TeamCity, як правило, автоматично, майже одразу після внесення змін у код, що міститься в репозиторії;
- TeamCity здійснює процес компілювання та пакування програмного продукту, а також доставляє його в GoCD для подальшого розгортання;
- перед запуском процесу розгортання GoCD оновлює параметри середовища та конвеєра (адреса сервера БД, логін до БД, пароль, і т.п.) і запускає процес розгортання.

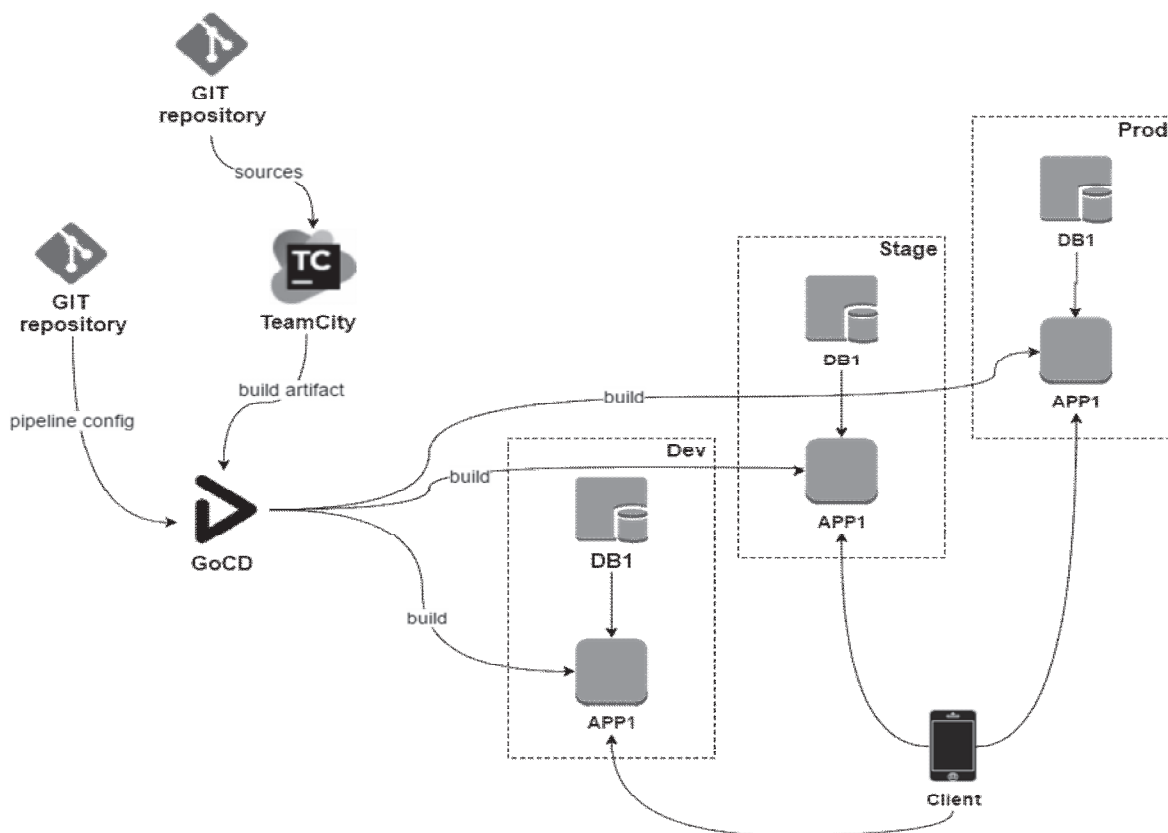


Рис. 8. Загальний вигляд реалізації CI/CD процесу

Окрім цього, слід зауважити, що кожна збірка повинна бути розгорнутою на кожному із середовищ, перш ніж потрапить на останній з них (на публічне використання). Тобто пропустити якийсь з етапів неможливо. Це забезпечує високу надійність та стабільність збірок.

Висновки

Використовуючи розглянутий підхід, було спроектовано та реалізовано процес неперервної інтеграції та розгортання з використанням утиліти TeamCity та GoCD. Процес розгортання відбувається за концепцією pipeline, тобто труби. Це означає, що кожна збірка розроблюваного програмного продукту повинна пройти кожен етап тестування, перш ніж потрапити у публічне використання. Такий підхід дає змогу якісно тестувати програмний продукт, що збільшує його надійність та стабільність.

Система неперервної інтеграції TeamCity та система неперервного розгортання GoCD забезпечують гнучкість, надійність, швидкодію та масштабованість всього процесу розроблення, тестування та впровадження програмного продукту, що добре відображається на якості продукту загалом.

1. *DevOps* // *Википедия – свободная энциклопедия*. 2016. [Електронний ресурс]. URL: <https://ru.wikipedia.org/wiki/DevOps> (дата звернення: 28.12.2016). 2. *Неперервна інтеграція* // *Википедия – вільна енциклопедія*. 2016. [Електронний ресурс]. URL: https://uk.wikipedia.org/wiki/Неперервна_інтеграція (дата звернення: 28.12.2016). 3. *Continuous delivery* // *Wikipedia – The free encyclopedia*. 2016. [Електронний ресурс]. URL: https://en.wikipedia.org/wiki/Continuous_delivery (дата звернення: 28.12.2016).