Доставка на софтуер - особености и процеси

Тема №2

Подходи при доставката на софтуер

Част 2: DevOps, GitOps, CI/CD

Съдържание

1 Въведение

Във времена на бързи технологични промени и все по-високи изисквания към софтуерните приложения, съвременната софтуерна индустрия се нуждае от иновативни подходи за оптимизиране на разработката, доставката и управлението на софтуера. За тази цел, три ключови концепции се явяват полезни инструменти - DevOps, GitOps и Continuous Integration/Continuous Deployment (CI/CD).

DevOps, като интегриран подход, размива границите между разработчиците и операторите с цел подобряване на процесите, свързани с разработката и поддръжката на софтуера. GitOps разширява този подход, насочвайки се към управлението на инфраструктурата посредством системи за контрол на версиите (най-често Git, от където идва и името). Същевременно, CI/CD практиките автоматизират

сглобяването(assembly), тестването и стъпките от физическата доставка на софтуера, насърчавайки непрекъснатото интегриране и доставка на нови версии на кратки интервали.

В тази лекция ще се фокусираме върху важността и приложението на тези концепции в съвременната софтуерна индустрия. Ще ги разгледаме по-подробно и ще изследваме тяхната роля в процесите на разработка и доставка на софтуер.

2 Понятие за DevOps

2.1 Дефиниция на DevOps

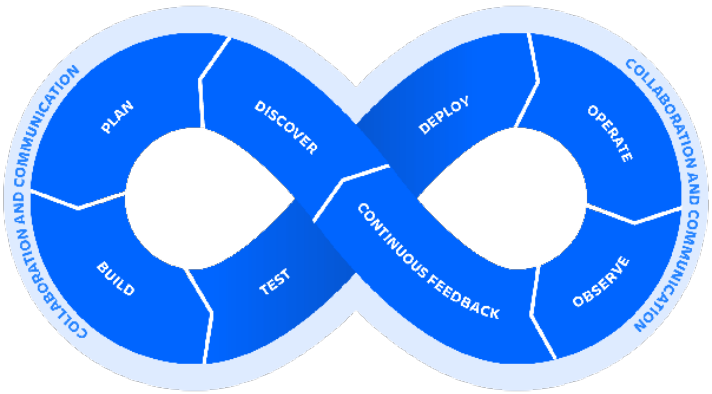
DevOps представлява философия, набор от практики и култура, обединяващи разработчици и оператори с цел подобряване на софтуерната разработка и доставка. Този подход насърчава интегрирането и сътрудничеството между двата традиционно независимите отдела - разработка (Development) и операции (Operations). В резултат на което отговорността за продукта и функционирането му е споделена.

В ядрото на DevOps стои идеята за премахване на "стената" между разработката и операциите. Вместо да бъдат изолирани и да се планират, и изпълняват поотделно, DevOps насърчава тяхното тясно сътрудничество. Това означава, че разработчиците и операторите работят заедно през целия жизнен цикъл на софтуерния продукт - от идеята и проектирането до разработката, тестването, внедряването и поддръжката му.

DevOps съчетава гъвкавите методологии за разработка, с автоматизирани процеси за интеграция и доставка. Той се базира на принципите на "непрекъсната интеграция" (Continuous Integration) и "непрекъсната доставка" (Continuous Delivery), които позволяват на екипите да доставят софтуер на по-чести и по-предсказуеми интервали.

Целта на DevOps е да ускори и оптимизира процесите на разработка и доставка на софтуер, като намали времето изминало от идеята до пускането на продукта на пазара. Това води до по-голяма възможност да се реагира на променящите се изисквания на клиентите и по-голяма конкурентоспособност на организациите.

Ефективно DevOps културата издига изискванията относно оперирането на софтуера на равно с архитектурата и разработката, увеличавайки разбирането и видимостта на потенциални проблеми и трудности.



Фиг. 1: DevOps цикълът

2.2 Защо DevOps е важен в софтуерната индустрия?

В съвременния бизнес свят, където скоростта и качеството на софтуера са от съществено значение, DevOps се явява ключов фактор за успех. Организациите, които приемат и прилагат DevOps практики, са по-гъвкави, конкурентоспособни и способни да доставят стойност на клиентите си по-бързо и ефективно. Малко по-късно в лекцията ще разгледаме принципите на DevOps, които правят методологията полезна и привлекателна за софтуерната индустрия.

2.3 История и еволюция на DevOps

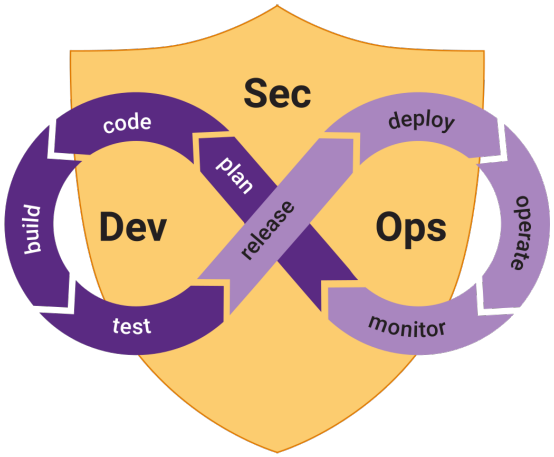
Терминът "DevOps" е първоначално използван от Патрик Дебоа след като гледа важната презентация на Flickr на конференцията O'Reilly Velocity през 2009 г. Джон Олспо (SVP Операции) и Пол Хамънд (Директор на разработката) споделят как увеличеното сътрудничество между отделите за разработка и операции позволява на Flickr да извършва "10+ внедрявания на ден". Това изглежда малко спрямо стотиците на ден, които в момента се постигат от Amazon, Google и Netflix. В онзи момент обаче концепциите, които те описаха, са революционни.

Патрик Дебоа вече работи с Андрю Клей Шейфър и други специалисти, с цел усъвършенстване на гъвкавостта на инфраструктурата. Той решава да организира конференция за сътрудничество свързана с разпространението на тези нововъзникващи идеи и я наименува Devopsdays (която се разраства от едно събитие през 2009 г. до около 80 глобални събития през 2019 г.)

Така се ражда DevOps. Въпреки че името DevOps е ново, действително, в корена на концепцията стои идеята за усъвършенстването на вече съществуващи практики, насочвайки интензивно вниманието към постигане на унифицирано разбиране и редуциране на драстичните различия в компетенциите на членовете на даден екип, с цел постигане на лесна и ефективна колаборация.

2.4 DevSecOps

DevSecOps е методология в областта на софтуерната разработка и операции, която се фокусира върху интегрирането на сигурността във всички аспекти на цикъла на разработка и доставка на софтуера. Тя се стреми да направи сигурността неделима част от процесите на DevOps, като осигурява мерките по сигурността да се прилагат от самото начало на разработката до пазарната експлоатация. Това включва автоматизация на тестове за сигурност, непрекъснат мониторинг на сигурността и обучение на екипите по добри практики, касаещи сигурността. DevSecOps има за цел да намали рисковете от инциденти свързани със сигурността и да гарантира, че софтуерните приложения са сигурни и надеждни.



Фиг. 2: DevSecOps

3 Принципи на DevOps

DevOps се базира на следните ключови принципи, които подчертават важността му в софтуерната индустрия:

3.1 Сътрудничество и интеграция

DevOps насърчава тясното сътрудничество между разработчиците и операторите. Като работят заедно, те подобряват ефективността и качеството на софтуерните продукти. Този принцип акцентира върху нуждата от обединение на знания и усилия за постигане на общи цели.

3.2 Автоматизация на процесите

Автоматизацията е неделима част от DevOps. Този принцип включва използването на инструменти и практики, които автоматизират различни аспекти на разработката и доставката на софтуер. Автоматизираните процеси подпомагат намаляването на човешките грешки и ускоряват разработката и доставката.

3.3 Контрол на версиите и конфигурацията

Важен аспект на DevOps е контролът на версиите на софтуера и конфигурацията на инфраструктурата. Този принцип включва стриктно управление на версиите на софтуера и инфраструктурата, което осигурява стабилност и предсказуемост. Също позволява да се проследяват промени и да се връщат предишни състояния при нужда.

3.4 CI/CD

CI/CD са ключови принципи на DevOps. Те включват автоматизирани процеси за интегриране и доставка на софтуера, удовлетворявайки принципа за автоматизация. Малко по-късно ще обърнем повече внимание на тях.

3.5 Сигурност от началото (DevSecOps)

DevOps не само включва бърза разработка и доставка, но и обръща внимание на сигурността. Принципът DevSecOps отдава нужното внимание на интегрирането на сигурността в целия жизнен цикъл на разработка и доставка на софтуера.

3.6 Прозрачност и комуникация

Добрата комуникация и прозрачност са от съществено значение за успешния DevOps. Ефективната комуникация между всички членове на екипа и заинтересовани страни осигурява разбирателство и ускорява процесите.

3.7 Култура на учене и подобрение

Философията на DevOps се гради върху разбирането за постоянно учене и подобрение. Екипите трябва да бъдат гъвкави, насочени към проактивност и ефективно прилагащи получена обратна връзка в следствие на настъпили пропуски в определени моменти, когато е необходимо. По този начин се осигурява постоянно развитие и иновации. Също така екипите трябва да проявяват интерес към иновациите в индустрията и да мислят за потенциалното им прилагане в екипните процеси.

3.8 Измеримост и метрики

Използването на метрики и анализ на данни играе важна роля в DevOps. Този принцип позволява на екипите да оценяват производителността и да правят подобрения въз основа на факти и данни.

4 Git и GitOps

4.1 Въведение в системата за контрол на версиите Git

Git е изключително мощна система за контрол на версиите, която се използва широко в софтуерната индустрия. Тя позволява на разработчиците да следят, управляват и колаборират по проектите си по ефективен начин. С Git, всяка промяна в кода се записва и се пази в историята, позволявайки лесно връщане към предишни версии и сравняване на промените.

4.2 Как Git се интегрира с DevOps

Git играе ключова роля в DevOps процеса, като осигурява съхранение и управление на изходния код на приложението. В комбинация с автоматизирани инструменти за сглобяване(assembly), тестване и внедряване, Git позволява непрекъсната интеграция и непрекъсната доставка на софтуера.

4.3 Концепцията на GitOps и как работи

GitOps е методология, която използва Git като централен източник на истината за текущото и желаното състояние на инфраструктурата и приложенията. С GitOps, цялата конфигурация на инфраструктурата и процесите за доставка се съхраняват в Git хранилища. Промените в тези хранилища се прилагат автоматично в инфраструктурата, което гарантира възпроизводимост и минимизира вероятността за грешка.

4.4 Предимства на GitOps

GitOps предоставя няколко ключови предимства, включително:

- Структуриран и управляем начин за управление на инфраструктурата.
- Извършване на промени чрез Git позволява преглед и одит на всички промени.
- Възможност за лесно възстановяване и репликация на среди.
- Подобрена сигурност чрез управление на достъпа и автентикацията в Git хранилища.

4.5 Примери за GitOps

За да обясним как работи GitOps, можем да предоставим следните примери:

- **Доставка на приложение:** Когато разработчиците завършат нова версия на приложение, те публикуват кода и конфигурацията в Git хранилище, след което GitOps инструментите автоматично прилагат тези промени в production средата.
- **Скалиране на ресурси:** Ако се налага да се увеличи броя на инстанциите на приложението поради повишен трафик, операторите могат да актуализират желаната конфигурация в Git, и GitOps инструментите ще извършат нужните действия за създаване на нови ресурси.
- **Създаване на нови тестови среди:** Когато екипът за тестове има нужда от нови среди за тестване, те могат да създадат описанието на средата в Git хранилище, и GitOps инструментите ще приложат тази конфигурация, създавайки средата.

Тези примери илюстрират как GitOps улеснява и ускорява управлението на инфраструктурата и приложенията чрез използване на Git като единен източник на истина и автоматизация на процесите.

5 Continuous Integration/Continuous Deployment (CI/CD)

5.1 Основни концепции

Непрекъсната интеграция (Continuous Integration - CI) и непрекъснатото внедряване (Continuous Deployment - CD) са две взаимосвързани, фундаментални практики в областта на DevOps. Те имат за цел да подобрят и ускорят процесите на разработка и доставка на софтуерни приложения, като съчетават автоматизация, тестване и непрекъсната обратна връзка.

5.2 Непрекъсната интеграция (CI)

CI е практиката на редовното и автоматизирано интегриране на кода, написан от различни разработчици, в общото хранилище на проекта. Главната идея зад CI е да се предотвратяват и откриват конфликти и проблеми, които могат да възникнат при сливане на промените в кода към основния код. Когато разработчик завърши работа по определен модул или функционалност, той изпраща своите промени към хранилището. След това се активира автоматизиран процес, който включва компилиране на кода, изпълнение на автоматизирани тестове и генериране на отчети. Ако този процес приключи успешно, измененията могат да бъдат приети и интегрирани в основния код. Ако възникнат проблеми, разработчиците се информират незабавно и ги отстраняват. CI може да включва и внедряване на тестова среда с цел преглед и демонстрация на промените.

Главните предимства на CI включват:

- Бърза и непрекъсната интеграция на нов код.
- Ранно откриване и отстраняване на грешки.
- Увеличена стабилност на приложението.
- Подобрено сътрудничество между разработчиците.

5.3 Непрекъснато внедряване (CD)

CD разширява практиката на CI, като включва автоматизирано внедряване на приложението до продуктивната среда след успешно протичане на CI. Това означава, че всеки път, когато разработчиците внесат промени в кода, тези промени могат автоматично да преминат през процес на тестване и да бъдат пренесени до production среда (или междинна такава, например вътрешна среда), където потребителите могат да използват актуализираната версия на приложението.

Главните предимства на CD включват:

- Бързо и автоматизирано разпространение на нова функционалност до потребителите.
- Минимизиране на риска от човешки грешки при доставката.
- Повишаване на сигурността и стабилността на приложението.

Цикълът на интеграция и доставка съкращава времето от написването на код до доставката на нови функции и подобрения до потребителите. Това е ключов елемент в разработката на модерни софтуерни приложения и поддържа тяхната конкурентоспособност на пазара.

6 Заключение

В тази лекция разгледахме важни концепции и практики, свързани с DevOps, GitOps и CI/CD. DevOps представлява методология, насочена към подобряване на сътрудничеството между разработчиците и операторите, като насърчава автоматизацията, контрола на версиите и непрекъсната интеграция и доставка. Принципите на DevOps включват сътрудничество, автоматизация, контрол на версиите и CI/CD, които се фокусират върху ускоряването и подобряването на разработката и доставката на софтуер.

Git е система за контрол на версиите, която позволява управление на изходния код и сътрудничество между разработчиците. GitOps използва Git като основен източник на конфигурации за управление на инфраструктурата и приложенията. GitOps предоставя структуриран и декларативен начин за управление на инфраструктурата, като използва Git за съхранение на конфигурацията и автоматично прилага промените.

В областта на CI/CD научихме, че непрекъсната интеграция (CI) се отнася до редовното и автоматизирано интегриране на кода, като целта е да се предотвратяват и откриват грешки рано в процеса. Непрекъснатото внедряване (CD) разширява практиката на CI и включва автоматизирана доставка на приложението след успешно интегриране. Този цикъл на интеграция и доставка спомага да се намали времето за доставката на нови функционалности и подобрения до потребителите.