**Практична робота №11-2. DevOps – методики розробки і безперервної доставки версій програмного продукту**

**Мета роботи**: ознайомлення з методиками розробки і безперервної доставки версій програмного продукту

**Завдання:**

1. Ознайомитися з теоретичною частиною.
2. Надати письмові відповіді на запитання.
3. Для власного проекту визначте обов‘язкові дії інженерів DevOps.

**КОНТРОЛЬНІ ПИТАННЯ:**

1. Які навички у розробці, системному адмініструванні, управлінні інфраструктурою ( ***hard skills)*** повинен мати інженер DevOps?
2. Як ми можемо визначити Soft skills ?
3. Які інженер DevOps повинен мати софт скіли ?
4. Які практики входять в поняття DevOps і навіщо вони потрібні?
5. Чим займаються DevOps-інженери і що вони повинні вміти?
6. **В чому саме полягає суть DevOps-культури?**
7. **Які бувають DevOps-практики?**
8. **Чим саме займається DevOps-інженер?**
9. **Що повинен знати DevOps-інженер?**

**Теоретичні відомості**

**DevOps** (Development and Operations) –  це методика, спрямована на забезпечення ефективної взаємодії розробників та кінцевих користувачів продукту та оптимізацію всіх процесів життєвого циклу ПЗ.

**DevOps – це набір практик,** що поєднує **розробку програмного забезпечення** з його обслуговуванням і експлуатацією. Його назва відображає ці дві частини: **Development and Operations** (розробка та операції). DevOps виник із колекції попередніх практик. Серед них – система **Agile development, Toyota Way та Lean manufacturing.** Термін DevOps став широко відомий на початку 2010-х років.

**Основна мета DevOps – скоротити час** між тим, коли **внесли зміни у код** і тим, коли **клієнт ці зміни відчув,** не впливаючи при цьому на надійність продукту. Фахівець прагне узгодити цілі розвитку з потребами організації для створення цінності бізнесу.

**Інженер DevOps – це** ІТ-спеціаліст **широкого профілю,** який повинен мати грунтовні знання як у розробці, так і в різних операціях, включаючи **кодування, управління інфраструктурою, системне адміністрування та ланцюжки інструментів DevOps**. Інженери DevOps також повинні володіти навичками міжособистісного спілкування, оскільки вони працюють з різними командами компанії, щоб створити середовище для більш ефективної співпраці.

DevOps -інженери повинні добре **розуміти загальну архітектуру системи,** ініціалізацію та адміністрування. Також DevOps-інженер повинен мати досвід роботи з **традиційним набором інструментів** розробника та практиками, такими як:

* використовувати систему **контролю первісного коду (Sourse Code Control System);**
* надавати та отримувати огляду коду **(Code Review);**
* писати **модульні тести;**
* бути знайомим з **принципами Аgile.**

**Існує думка, що** у цій професії **не потрібно технічних знань**!

Роль інженера DevOps буде **відрізнятися від компанії до компанії,** але незмінно передбачає, що фахівець буде **поєднувати розробку релізів, забезпечувати інфраструктуру та управляти нею; бути системним адміністратором, забезпечувати безпеку всього, що є на проєкті, та просувати філософію DevOps.**

**DevOps Engineer** – це спеціаліст, який **бере участь у всіх етапах життєвого циклу продукту.** Цей фахівець забезпечує тісну **кооперацію між** командами розробників**,** щоб  оптимізувати їхні робочі процеси і, відповідно, **допомагає скоротити час, за який програмний продукт потрапить до кінцевого користувача.**

Коли ми говоримо про DevOps-інженера потрібно розуміти, що **цей фахівець не є якимось конкретним вузькопрофільним спеціалістом.** Ці фахівці приходять на посаду з різних професій. Наприклад, **інженер DevOps може почати як розробник програмного забезпечення, який контролює аспекти ІТ-операцій.** З іншого боку, інженер DevOps може бути в минулому системним адміністратором, оскільки він отримав знання про кодування, створення сценаріїв, інтеграцію та тестування. Але саме до обов’язків інженера DevOps входить зміна бізнес-процесів таким чином, щоб вирішити організаційні проблеми і покращати бізнес-результати.

Незважаючи на широкий і різноманітний діапазон ролей DevOps, **є деякі загальні навички та вміння, на які вам потрібно звернути увагу, якщо ви націлені стати "девопсом", то ви повинні мати:**

* досвід адміністрування ОС, таких як **Linux і Windows;**
* значний досвід роботи з рядом **інструментів автоматизації** та керування конфігураціями, такими як традиційні сценарії, а також більш специфічними інструментами**;**
* чітке розуміння програмування та створення сценаріїв поширеними **мовами програмування, такими як PHP, Python, Perl і Ruby;**
* знання принаймні однієї **основної мови кодування, такої як C++ або Java;**
* здатність ідентифікувати, оцінювати та **інтегрувати різні технології** з відкритим кодом , а також хмарні сервіси;
* серйозний досвід ІТ-апаратного забезпечення та роботи з практичним досвідом **встановлення серверів,** сховищ і мережевих пристроїв, ініціалізації та моніторингу. Це зазвичай доповнюється знаннями найкращих ІТ-практик для відмовостійких операцій з високою доступністю;
* здібності до **підтримки команд та співпраці з ними,** а також підтверджені навички взаємодії з клієнтами;
* грунтовне знання **технологій віртуалізації**, таких як **VMware vSphere** для віртуальних машин, а також досвід роботи з контейнерними технологіями, такими як **Docker і Kubernetes;**
* перевірений **досвід роботи з інструментами CI/CD, де** CI (Continuous Integration) — це неперервна інтеграція, а CD (Continuous Delivery) — неперервна доставка. Цей набір методик дозволяє розробникам частіше і надійніше розгортати зміни в програмному забезпеченні.**). До таких інструментів відносяться Microsoft GitHub, Atlassian Jira та Confluence, Red Hat Ansible, Prometheus та Jenkins;**
* грунтовний практичний досвід роботи з **публічними хмарними ресурсами** та службами, такими як **AWS, Microsoft Azure і Google Cloud;**
* досвід роботи з різноманітними інструментами ІТ-моніторингу та управління, такими як **Cloudflare і Datadog;**
* знання про те, як усунути несправності та вирішити технічні проблеми в тестовому та робочому середовищах.

## Роль DevOps-інженера та його/її обов’язки

На сьогодні DevOps інженер – одна з найнеобхідніших позицій в компаніях, де розробляються продукти. Чому так? Щоб відповісти на це питання, для початку нам необхідно розібратися із самою методологією і DevOps інженерами.

## Що таке DevOps?

DevOps або ж Development Operations – це не просто професія, а певна методика для розв’язання проблем у взаємодії між командами експлуатації та розробки. До виникнення DevOps була наступна ситуація: розробники запускали код в експлуатацію, при виникненні будь-яких проблем у команди експлуатації, відповідальність перекладалася на першу команду. Це призводило до того, що багато продуктів випускалися з запізненням або ж з вадами. Також на кінцевий продукт впливало і велика кількість змін, яких важко було відстежити в процесі супроводження.

Методологія DevOps була створена для розв’язання подібних проблем. Іншими словами, DevOps – це міст між командами розробки та експлуатації.

## Тоді хто такі DevOps інженери?

DevOps інженери – це фахівці, які розбираються з усіма можливими процесами циклу розробки й експлуатації. До них відносяться: розробка, тестування, архітектура продукту, оцінка ризиків захисту, різні способи автоматизації, підтримка продукту як до, так і після випуску. Більш того, фахівець розуміє, як працюють обидві команди розробників і експлуататорів.

DevOps умовно ділиться на кілька субролей:

* Build Engineer – професіонал, який відповідає не тільки за збірку коду, але і за розбір конфліктів і підтягування різних залежностей;
* Release Engineer – людина, яка відповідає за доставлення коду від команди розробників в експлуатацію, за фінальну гілку для тестування та ін.
* Automation Engineer – фахівець, який працює над автоматизацією всіх продуктів.
* Security Engineer – людина, яка відповідає за коректність тестів з безпеки, перебування слабких місць в компонентах продукту.

На ділі ж всі ці субролі виконує одна людина або ж кілька ролей інший фахівець. Буває так, що розробник може виконувати завдання Build Engineer. А з автоматизацією працюють системні адміністратори. В результаті DevOps інженер тільки автоматизує та робить остаточне опрацювання та автоматизація виробництва та доставки коду в експлуатацію.

Значення широковживаних термінів, які варто знати кожному DevOps інженеру.

## Документація

Документація – це важливий інструмент DevOps команди, оскільки допомагає переконатися, що всі поділяють спільну мету і працюють на її досягнення. Документація може бути основним інструментом для стандартизації, співпраці, обміну знаннями й адаптації нових працівників.

## Контейнер

Контейнер – це легкий портативний пакет, який містить усе необхідне для роботи програми, включно з її кодом і залежностями, такими як бібліотеки та системні інструменти. Уявіть собі невелике автономне середовище, яке може працювати будь-де без встановлення чи налаштування.

## Мікросервіс

Мікросервіс – це невеличкий незалежний програмний компонент, який виконує визначену функцію в межах більшої програми. Він як модульний будівельний блок, який можна використати для будівництва складніших застосунків. Кожен мікросервіс є самодостатнім, має свій код, базу даних та API та зв’язується з іншими мікросервісами за допомогою чітко визначених інтерфейсів.

## Автоматизація

Автоматизація в DevOps передбачає використання технологій, щоб упорядкувати та спростити повторювані, трудомісткі завдання, які є частиною процесу розроблення та розгортання програмного забезпечення. Це допомагає скоротити час і витрати, пов’язані з ручним виконанням завдань, покращує якість і надійність і дає змогу командам зосередитися на завданнях вищого рівня, таких як інновації та розв’язання проблем.

## Безсерверна архітектура

Безсерверною називають таку модель розгортання застосунків, де хмарний провайдер керує інфраструктурою і автоматично виділяє ресурси за потреби, без залучення користувача до налаштування чи підтримання серверів. Ці архітектури проєктуються так, щоб вони мали високий рівень масштабованості, ефективності та рентабельності.

## Розподілювач навантаження

Розподілювач навантаження – це інструмент, який розподіляє вхідний трафік між кількома серверами або віртуальними машинами (інстансами), щоб підвищити продуктивність, надійність та доступність застосунків.

## Логування та моніторинг

Логування та моніторинг застосовуються, щоб збирати та аналізувати дані від застосунків або інфраструктури, діагностувати проблеми та мати уявлення про продуктивність та шаблони використання.

## CI/CD (Безперервна інтеграція/Безперервне постачання)

CI/CD охоплює практики та інструменти, що автоматизують розроблення та постачання програмного забезпечення, від змін у коді до розгортання у робочому середовищі. Мета CI/CD – надати командам змогу швидше та надійніше постачати високоякісні програмні застосунки, одночасно зменшуючи ризики помилок і затримок.

## Конвеєр

Конвеєр – це набір автоматизованих кроків для створення, тестування та розгортання програми чи системи. Конвеєр зазвичай включає серію етапів або фаз, кожна з яких виконує певне завдання або набір функцій.

## Agile

Agile – це принципи й методи розроблення програмного забезпечення, які ґрунтуються на співпраці, гнучкості та постійному вдосконаленні. Методології Agile наголошують на важливості тісної взаємодії між розробниками, тестувальниками та іншими стейкхолдерами протягом усього життєвого циклу розроблення програмного забезпечення, зосереджуючись на внесенні невеликих поступових змін у застосунок, які можна швидко протестувати та перевірити.