

1.3 Формула DevOps або чому не буває Junior DevOps

У наш час, коли розробка програмного забезпечення є важливою складовою бізнесу, методологія DevOps стала дуже популярною.

Ця методологія дозволяє розробникам швидко і надійно постачати свої продукти на ринок. Але які компетенції і рівні знань необхідні для роботи в цій галузі та який рівень компенсацій можна очікувати?

Ситуація на ринку DevOps

Сьогодні ринок праці DevOps активно розвивається і є велика кількість вакансій для спеціалістів цієї області. Дедалі більше компаній залучають DevOps-інженерів для автоматизації процесів розробки та розгортання програмного забезпечення.

Про Junior, Middle та Senior

DevOps - це область, в якій є багато можливостей для розвитку кар'єри. Зазвичай рівень знань та досвід DevOps-інженерів поділяється на три рівні: Junior, Middle та Senior.

Рівні компенсацій

Залежно від рівня складності і масштабу проєкту, рівень компенсації для DevOps-інженерів може значно відрізнитися. Порада для джуніорів: якщо ви підходите під вакансії Junior, найімовірніше, ви підходите і під вакансії Middle. Тому вам можуть недоплачувати. Junior DevOps, які отримують умовно ~\$1000, подавайтеся на вакансії Middle і пробуйте отримати у два рази більше.

Кого шукають

Більшість компаній, які шукають DevOps-інженерів, найбільше цінують практичний досвід роботи з інструментами автоматизації та знаннями про процеси розробки програмного забезпечення. Бажано також мати знання з моніторингу, логування та відладки.

Чому "немає Junior DevOps":

Поширена помилкова думка про те, що існують легкі шляхи отримання позиції у DevOps через Junior DevOps. Як правило, це не так. У більшості випадків для

роботи на посаді DevOps необхідні конкретні навички та досвід з технологіями. Без попереднього практичного досвіду не можливо обіймати посаду, навіть Junior DevOps, який передбачає досвід 1-2 роки.

Формула DevOps у п'ять кроків:

Крок 1. Намалюйте символ нескінченності і розділіть його на 7 частин. Позначте кожную частину як

- план,
- код,
- збірка,
- тестування,
- доставка,
- розгортання та
- моніторинг.

Це і є SDLC.

Крок 2. Напишіть програму будь-якою мовою програмування з нуля. Це може бути мікросервіс, моноліт, функція або навіть простий рядок коду. Не копіюйте нічого. Напишіть її самостійно. Відкрийте книгу про мову Go, наприклад, вивчіть основи і почніть кодувати свій вебдодаток. Або ж візьміть цікаву ML-модель на Python і пограйтеся з нею.

Крок 3. Поглиблено пройдіть кожен етап SDLC зі своєю програмою. Саме тут ви починаєте розуміти, що є найбільш важливим для вас. Ви починаєте розвивати свій власний унікальний набір навичок DevOps інженера. DevOps як процес багатогранний, тому не обов'язково бути експертом у всьому. Зосередьтеся на тому, щоб знайти свою нішу і розвиватися в тому напрямку, який вам подобається.

Крок 4. Зміцнюйте свої практичні навички і, головне, починайте набувати досвіду прийняття рішень. Повторіть крок 3, але цього разу використовуйте альтернативний стек. Замініть все, від кодової бази до системи CI/CD, оркестратора, наприклад, Kubernetes, хмари, і зосередьтеся на найкращих практиках захисту вашого додатку та інфраструктури. Чим радикальніші зміни у ваших методах і підходах, тим більший досвід ви отримаєте. На цьому етапі ви вже здобуваєте професіоналізм. Аналіз, прийняття рішень щодо стеку,

практичний досвід роботи з інструментами та методами високо цінуються на серйозних співбесідах.

Крок 5. Розширюйте свою професійну мережу. Всі менеджери з найму та ваші майбутні колеги вже тут. На цьому етапі ви позиціонуєте себе як експерта на ринку, де конкуренція і попит стрімко зростають. Це одразу ж приверне увагу HR. Ось і все. Ви привернули увагу HR і вони запросять вас на співбесіду. Там ви вже будете впевнено розбиратися в питанні, розкриєте себе як перспективного кандидата і почуєте “Ласкаво просимо в команду!”