

УДК 004.9:330.341.13

ОСОБЛИВОСТІ РОЗГОРТАННЯ ВЕБ-СЕРВІСУ ДЛЯ ДИСТАНЦІЙНОЇ ОСВІТИ

Мацибурки П. Т., Пасєка М. С.

*Івано-Франківський Національний Технічний Університет Нафти і Газу
Україна, 76000, м. Івано-Франківськ, вул. Карпатська, 15, pete.matsy@gmail.com*

***Анотація.** В статті розглянуті особливості розгортання веб-сервісу для дистанційної освіти засобами інформаційно-телекомунікаційних технологій.*

***Abstract.** The article describes the features of the deploy web service for distance education by means of information and telecommunication technologies..*

Вступ. Розгортання – важливий етап життєвого циклу будь-якого веб-сервісу. На цьому етапі необхідно розробити і впровадити архітектуру розгортання яка б забезпечувала безперебійну роботу усіх його компонентів. Надійність і масштабованість є основними вимогами до хорошої архітектури розгортання веб-сервісу.

Припустимо що у нас є веб-сервіс для надання послуг з дистанційної освіти, який складається з таких компонентів: веб-застосунок; база даних; двигун повнотекстового пошуку; сховище контенту.

Почнімо з розгортання веб-застосунка. Для його роботи потрібно забезпечити відповідне оточення на віддаленому сервері. Наприклад, для Ruby on Rails додатка необхідна unix-подібна ОС з встановленими інтерпретаторами ruby, nodejs, а також gcc для компіляції нативних розширень. Автоматизувати процес налаштування оточення можна за допомогою системи керуванням конфігураціями Puppet.

Puppet складається з серверної частини, що виконується на віддаленому сервері, і клієнтської – розміщується на стороні розробника. Все, що необхідно – це встановити puppet-server, і надати доступ до нього клієнту puppet. Після цього розробник зможе здійснювати конфігурацію оточення за допомогою puppet маніфестів одночасно на усіх віддалених серверах [1].

Керування версіями веб-застосунка будемо здійснювати за допомогою інструменту Capistrano. Він дозволить розгортати нові версії за допомогою системи керування версіями Git, використовуючи ssh протокол [2].

Отже, на даному етапі, у нас розгорнуто N серверів з робочою версією веб-застосунка. Тепер необхідно налаштувати проксі сервер для розподілення запитів користувачів між ними. Для цього використаємо балансувальник навантаження на Proxu. Його базова конфігурація зводиться до визначення IP адрес серверів [3].

На наступному етапі розгорнемо кластери БД. Реалізуємо master-slave реплікацію: на першому сервері розгорнемо slave кластер PostgreSQL, на наступному – master кластер і налаштуємо потокову реплікацію на slave кластер. Третій сервер використаємо як балансувальник навантаження

– встановимо pgpool-II і направимо операції запису на master, а зчитування – на master і slave[4]. Таким чином ми зможемо подвоїти кількість оброблюваних операцій зчитування даних з бази за рахунок відносно невеликої затримки при записі.

Тепер розгорнемо двигун повнотекстового пошуку Solr на віддаленому сервері. Solr запускається як окремий веб-додаток. Комунікація з ним відбувається по http протоколу, тому масштабування можна здійснити за допомогою збільшення кількості серверів і розподілення навантаження між ними через HAProxy.

На останньому етапі розгорнемо сховище даних Redis яке буде використовуватися для кешування.

Для зберігання контенту використаємо існуюче рішення – Amazon S3. Цей сервіс надає можливість для зберігання і отримання будь-якого обсягу даних, у будь-який час з будь-якої точки мережі. За допомогою Amazon S3 досягається висока масштабованість, надійність, висока швидкість, недорога інфраструктура зберігання даних.

Також використаємо інші сторонні сервіси: Amazon CloudFront CDN – хостинг статичного контенту; Amazon Route 53 – DNS сервіс; Amazon SES – email сервіс; Logentries – логування; New Relic – моніторинг продуктивності; Rollbar – відслідковування помилок.

Описана архітектура розгортання покликає надати хорошу надійність і продуктивність веб-сервісу, а також забезпечити легку масштабованість і ефективне використання ресурсів.

Висновки. Розгортання веб-сервісу для дистанційної освіти є ключовим елементом надання доступу до освітніх послуг, оскільки час знаходження необхідного контенту суттєво залежить від його архітектури.

Використані літературні джерела:

1. Puppet: установка і настройка puppet-сервера і puppet-агента [Електронний ресурс] Режим доступу: <https://rtfm.co.ua/centos-ustanovka-i-nastrojka-puppet-servera-i-puppet/>
2. Deploy приложения на RoR 4 с помощью Capistrano [Електронний ресурс] Режим доступу: <https://habrahabr.ru/post/213269/>
3. Load Balancing with HAProxy [Електронний ресурс] Режим доступу: <https://serversforhackers.com/load-balancing-with-haproxy>
4. Отказоустойчивый кластер Master-Slave на PostgreSQL [Електронний ресурс] Режим доступу: <https://habrahabr.ru/post/188096/>