

## Task 5 - Налаштування реплікації та перевірка відмовостійкості MongoDB

Ознайомтесь з реплікацію даних в MongoDB

<http://docs.mongodb.org/manual/core/replication-introduction/>

### Завдання:

- 1) Налаштувати реплікацію в конфігурації: Primary with Two Secondary Members (всі ноди можуть бути запущені як окремі процеси або у Docker контейнерах)  
- <http://docs.mongodb.org/manual/core/replica-set-architecture-three-members/>
  - Deploy a Replica Set for Testing and Development-  
<http://docs.mongodb.org/manual/tutorial/deploy-replica-set-for-testing/>
  - <http://www.tugberkugurlu.com/archive/setting-up-a-mongodb-replica-set-with-docker-and-connecting-to-it-with-a-net-core-app>
- 2) Продемонструвати запис даних на *primary node* з різними Write Concern Levels (<http://docs.mongodb.org/manual/core/write-concern/>):
  - Unacknowledged
  - Acknowledged
  - Journalled
  - AcknowledgedReplica  
(<http://docs.mongodb.org/manual/core/replica-set-write-concern/>)
- 3) Продемонструвати Read Preference Modes: читання з *primary* і *secondary* node (<http://docs.mongodb.org/manual/core/read-preference/>)
- 4) Спробувати зробити запис з однією відключеною ногою та *write concern* рівнім 3 та нескінченім таймаутом. Спробувати під час таймаута включити відключену ноду
- 5) Аналогічно попередньому пункту, але задати скінченний таймаут та дочекатись його закінчення. Перевірити чи данні записались і чи доступні на читання з рівнем *readConcern: "majority"*
- 6) Продемонстрував перевибори primary node в відключивши поточний primary (Replica Set Elections) -  
<http://docs.mongodb.org/manual/core/replica-set-elections/>
  - і що після відновлення роботи старої primary на неї реплікуються нові дані, які з'явилися під час її простою

7) Привести кластер до неконсистентного стану користуючись моментом часу коли *primary node* не відразу помічає відсутність *secondary node*

- відключивши дві *secondary node* протягом 5 сек. на мастер записати значення (з w:1) і перевірити, що воно записалось
- спробувати зчитати це значення з різними рівнями *read concern* -  
`readConcern: {level: <"majority" | "local" | "linearizable">}`
- включити дві інші ноди таким чином, щоб вони не бачили попереднього мастера (його можна відключити) і дочекатись поки вони оберуть нового мастера
- підключити (включити) попередню *primary*-ноду до кластеру і подивитись, що сталося зі значенням яке було на ній записано

8) Земувати *eventual consistency* за допомогою установки затримки реплікації для репліки

<https://docs.mongodb.com/manual/tutorial/configure-a-delayed-replica-set-member/>

9) Лишити *primary* та *secondary* для якої налаштована затримка реплікації.

Записати декілька значень. Спробувати прочитати значення з `readConcern: {level: "linearizable"}`

Має бути затримка поки значення не реплікуються на більшість нод

Опис додаткових команд Replication Reference -

<http://docs.mongodb.org/manual/reference/replication/>

### **Вимогу до оформлення протоколу:**

Завдання здається особисто без протоколу, або надсилається протокол який має містити:

- команди та результати їх виконання