**ACTIVIDAD 4 – PRUEBAS DE PARTICIONAMIENTO DE BASES DE DATOS NOSQL**

Emily zharith Sepulveda Rayo

Corporación Universitaria Iberoamericana

Ingeniería de software

Base de datos avanzadas

Bogotá D.C

2023

**CASOS DE PRUEBA**

|  |
| --- |
| **Caso de Prueba 1: Escalamiento Horizontal de Datos** |
| Requisito: Verificar la distribución de datos y el estado del clúster mediante sh.status(). |
| Prueba: Ejecutar sh.status() y examinar la distribución de documentos en los shards. |
| Resultado Esperado: Se espera visualizar la distribución de documentos en los shards y el estado actual del clúster. |

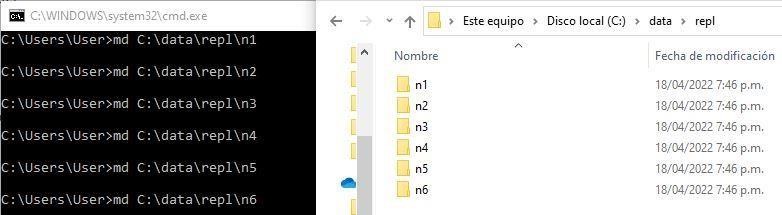
|  |
| --- |
| **Caso de Prueba 2: Fragmentación de Bases de Datos** |
| Requisito: Utilizar getShardDsitribution() para analizar el espacio en memoria asignado a cada shard y sus pesos. |
| Prueba: Ejecutar getShardDsitribution() para verificar el espacio en memoria y los pesos de cada shard. |
| Resultado Esperado: La salida debería mostrar el espacio en memoria asignado y los pesos correspondientes a cada shard. |

|  |
| --- |
| **Caso de Prueba 3: Escritura desde un Solo Servidor** |
| Requisito: Restringir la escritura a un solo servidor, específicamente el nodo primario. |
| Prueba: Intentar escribir en un nodo no primario al azar y validar desde qué nodo se permite la escritura. |
| Resultado Esperado: No se debería permitir la escritura desde un nodo secundario; solo el nodo primario tiene el permiso correspondiente. |

|  |
| --- |
| **Caso de Prueba 4: Grupo de Partición Activo** |
| Requisito: Verificar la actividad de los puertos asignados a servidores, enrutadores y shards. |
| Prueba: Utilizar netstat -a para asegurar que todos los puertos asignados al conjunto de replicación estén activos. |
| Resultado Esperado: Se espera que todos los puertos asignados a servidores, enrutadores y shards estén activos. |

|  |
| --- |
| **Caso de Prueba 5: Particionamiento Basado en Hash** |
| Requisito: Emplear el particionamiento basado en hash. |
| Prueba 1: Verificar los pesos iniciales de los chunks mediante getShardDsitribution(). |
| Resultado Esperado 1: Los pesos de los chunks deben ser uniformes inicialmente. |
| Prueba 2: Adicionar tres documentos de forma sucesiva y revisar los pesos de los chunks. |
| Resultado Esperado 2: Después de la adición de documentos, los pesos de los chunks deben aumentar de manera uniforme. |

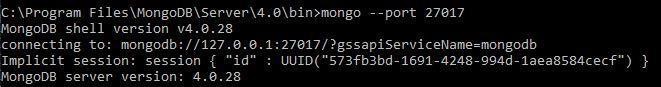
* **Pruebas de conversión de un replica set en un sharder clúster replicado Creación de directorios:**

****

* **Inicio de servidor primario**

****

* **Activación shell servidor primario**

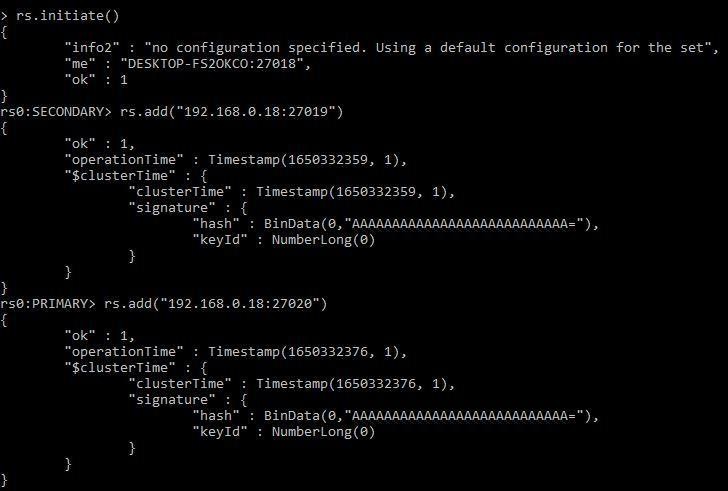
****

* **Inicio de servidores y activación de shell secundarios**

****

****

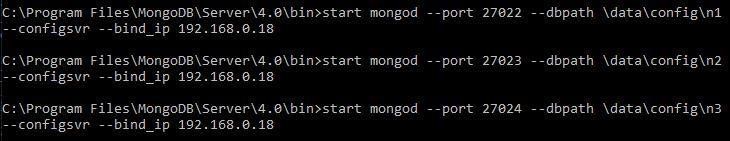
* **Creación de conjunto de replicas**

****

* **Llamado a la base de datos**

****

* **Inicio de servidores de configuración**

****

* **Activación de servidores de configuración**



* **Inicio enrutador**

****

* **Respuesta enrutador**

****

**Link de git https://github.com/esepulveda12/myfootballnine**

**Link del video https://youtu.be/htfFg1be7No**