Jonathan Roland

WEB 335 Introduction to NoSQL

Instructor: Prof. Krasso

12/10/2020

Discussion 9.1

1. **Explain the process for setting up authentication and user accounts.**

MongoDB Community edition supports the use of SCRAM (Salted Challenge Response Authentication Mechanism) and x.509 authentication standards. The Enterprise edition supports Kerberos and LDAP proxy authentication. Using x.509 as an example, authentication is set up in the following steps:

* Enable authentication with the –auth option for the mongod command in the MongoDB console, or set security.authorization to “enabled” in the MongoDB config file.
* Establish a trusted Certificate Authority (CA) either on your own resources or through a TLS/SSL vendor. A third party is preferred for most cases beyond an isolated network.
* Generate a root CA. This is the top-level authority. It includes such variables as a list of server hosts, client hosts, and port number.
* Generate a public and private key pair.
* Create an OpenSSL config file.
* Use the openssl req command with the -x509 option to create a root certificate.
* Create an intermediate CA for signing. This is a certificate signed by the root CA. It acts as a division of concerns and a buffer against compromised certificates compromising the whole structure.
* Concatenate the root and intermediate CA into a .pem file.
* Generate and sign member certificates. These are used by cluster members to verify membership to each other.
* Generate and sign client certificates.
* Create an admin user.
* Restart the replica set with authentication and authorization enabled.

MongoDB does not automatically make use of authentication or authorization features. This must be set up first by creating an admin user. The admin can then set up more users with specified roles and grants that enable or disable their ability to perform certain operations. Some of the available roles are:

* read
* readWrite
* dbAdmin
* userAdmin
* dbOwner
* clusterManager
* clusterMonitor
* hostManager
* clusterAdmin
* backup
* restore
* readAnyDatabase
* readWriteAnyDatabase
* userAdminAnyDatabase
* adAdminAnyDatabase
* root

1. **What is the system profiler and why is it important?**

The system profiler records system operations in the system.profile collection. It can help you find slow operations, but at a cost of an overall system performance decrease due to the added record keeping overhead. Best practice is to only enable it for short periods to gather data to evaluate. Even setting a slow heartbeat to the profiler will still result in significant slowdown to the whole system.

1. **What is mongostat and what is it used for?**

Mongostat is a command-line tool that updates server-wide stats on a user customizable time interval, defaulted to 1 second. It displays a count of each of these fields:

* insert, query, update, delete, getmore, command
  + Counts of each of these operations.
* flushes
  + How many times mongod flushed data to disk.
* mapped
  + The amount of memory mongod has mapped.
* vsize
  + The amount of virtual memory mongodb is using.
* res
  + The amount of non-virtual memory mongod is using.
* locked db
  + The database that spent the most time locked since the last report.
* idx miss %
  + The amount of index accesses that had to go to disk because it was not available in memory.
* qr|qw
  + How many reads and writes are waiting to process.
* ar|aw
  + How many clients are performing reads and writes.
* netIn
  + The number of network bytes coming in to MongoDB.
* netOut
  + The number of network bytes going out of MongoDB.
* conn
  + The total number of incoming and outgoing connections.
* time
  + The exact time these stats were taken.

**Reference:**

Bradshaw, S., Brazil, E., & Chodorow, K. (2019). *MongoDB: The Definitive Guide: Powerful and Scalable Data Storage* (3rd ed.). O’Reilly Media.