Nagios checks for Mongo

Below are a list of health checks that can be executed in Nagios via a python plugin called Nagios-MongoDB which is located here.

The original python script and instructions on executing from Nagios can be found here.

An example for configuring Nagios to run python scripts can be viewed here.

Check Percentage of Open Connections

This is a test that will check the percentage of free connections left on the Mongo server. In the following example it will send out an warning if the connection pool is 70% used and a critical error if it is 80% used.

Host: MongoDB server

Action: connections

Warning: 70 percent

Critical: 80 percent

Port: 55551

```
./check_mongodb.py -H c-test1-mongodb1.k12.int -A connections -P 55551 -W 70 -C 80
```

```
OK - 0 percent (45 of 51200 connections) used
```

Check Connections

This check will issue a warning if the connection to the server takes 2 seconds and a critical error if it takes over 4 seconds

Host: MongoDB server

Action: connections

Warning: 2 seconds

Critical: 4 seconds

Port: 55551

```
./check_mongodb.py -H c-test1-mongodb3.k12.int -A connect -P 55551 -W 2 -C 4
```

```
OK - Connection took 0 seconds
```

Check Replication Lag Percentage

This is a test that will test the replication lag percentage of Mongo servers. It will send out a warning if the lag is over 50 percents and a critical error if it's over 70 percents. Please note that this check gets oplog timeDiff from primary and compares it to replication lag. When this check reaches 100 percent full resync is needed.

Host: MongoDB server (Secondary servers ONLY)

Action: replication_lag_percent

Warning: 50 percent

Critical: 70 percent

Port: 55551

```
./check_mongodb.py -H eqc-test6-mongodb3.k12.com -A replication_lag_percent -P 55551 -W 50 -C 70
```

```
OK - Lag is 0 percents
```

```
./check_mongodb.py -H c-test1-mongodb2.k12.int -A replication_lag_percent -P 55551 -W 50 -C 70
```

```
CRITICAL - Lag is 1426 percents
```

DO NOT USE the Primary server (Example below)

```
./check_mongodb.py -H eqc-test6-mongodb1.k12.com -A replication_lag_percent -P 55551 -W 50 -C 70
```

```
OK - This is the primary.
```

Check Memory Usage

This is a test that will test the memory usage of Mongo server. In my example my Mongo servers have 32 gigs of memory so I'll trigger a warning if Mongo uses over 20 gigs of ram and a error if Mongo uses over 28 gigs of memory.

Host: Any MongoDB server

Action: memory

Warning: 20 GBs of memory used

Critical: 28 GBs of memory used

Port: 55551

```
./check_mongodb.py -H c-test1-mongodb3.k12.int -A memory -P 55551 -W 20 -C 28
```

```
OK - Memory Usage: 4.85GB resident, 6.97GB virtual, 0.00GB mapped, 0.00GB mappedWithJournal
```

Check Status of MongoDB Replicaset

This is a test that will check the status of nodes within a replicaset. Depending which status occurs, it sends a Warning during status 0, 3 and 5, critical if the status is 4, 6 or 8 and a ok with status 1, 2 and 7.

Note the trailing 2 0's keep those 0's as the check doesn't compare to anything.. So those values need to be there for the check to work.

Host: Any MongoDB server within the replica set

Action: replset_state

Warning: 0 - default value

Critical: 0 - default value

Port: 55551

```
./check_mongodb.py -H c-test1-mongodb1.k12.int -A replset_state -P 55551 -W 0 -C 0
```

```
OK -  c-test1-mongodb3.k12.int:55551: 1 (Primary) c-test1-mongodb2.k12.int:55551: 3 (Recovering)
c-test1-mongodb1.k12.int:55551: 3 (Recovering)
```