ZABBIX MEETUP ONLINE '21

Приключения команды интеграции в стране чудес или mongoDB и все-все-все

VADIMS IPATOVS un JŪLIJA ČUKINA
ІНЖЕНЕРИ КОМАНДИ ІНТЕГРАЦІЇ
ZABBIX

ZABBIX MEETUP ONLINE '21

Терминология

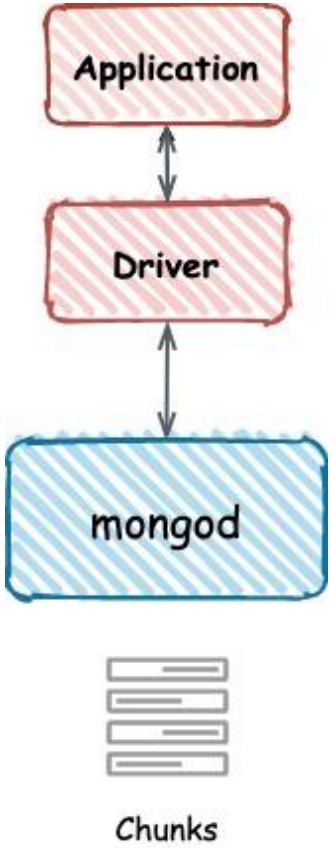
Терминология

RDBMS		MongoDB
Table, View	→	Collection
Row	→	Document
Index	→	Index
Join	→	Embedded Document
Foreign Key	→	Reference
Partition	→	Shard

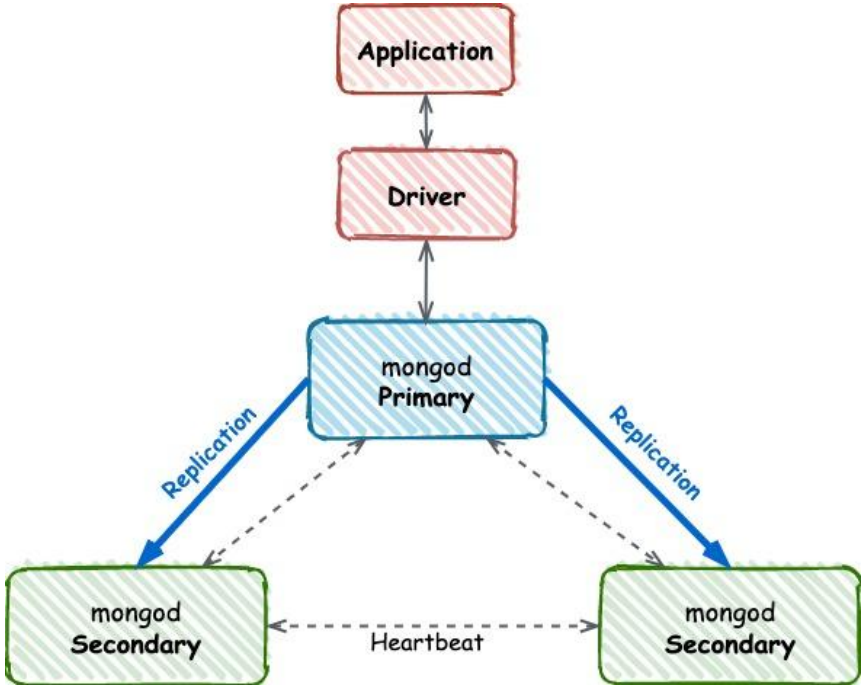
Standalone

mongod — прием запросов, их обработка и выполнение.

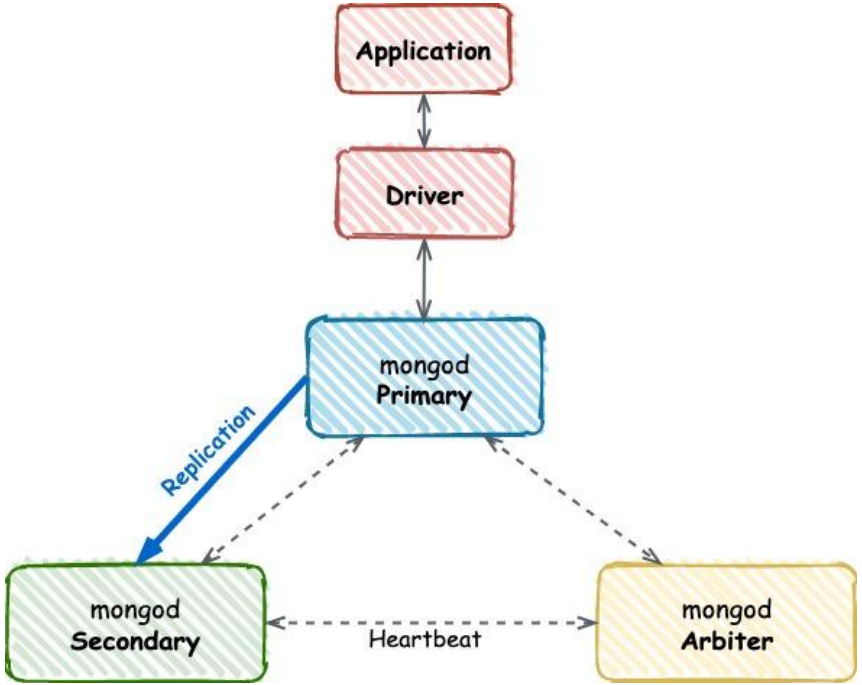
Данные хранятся в **chunks**.



Replica Set



Replica Set

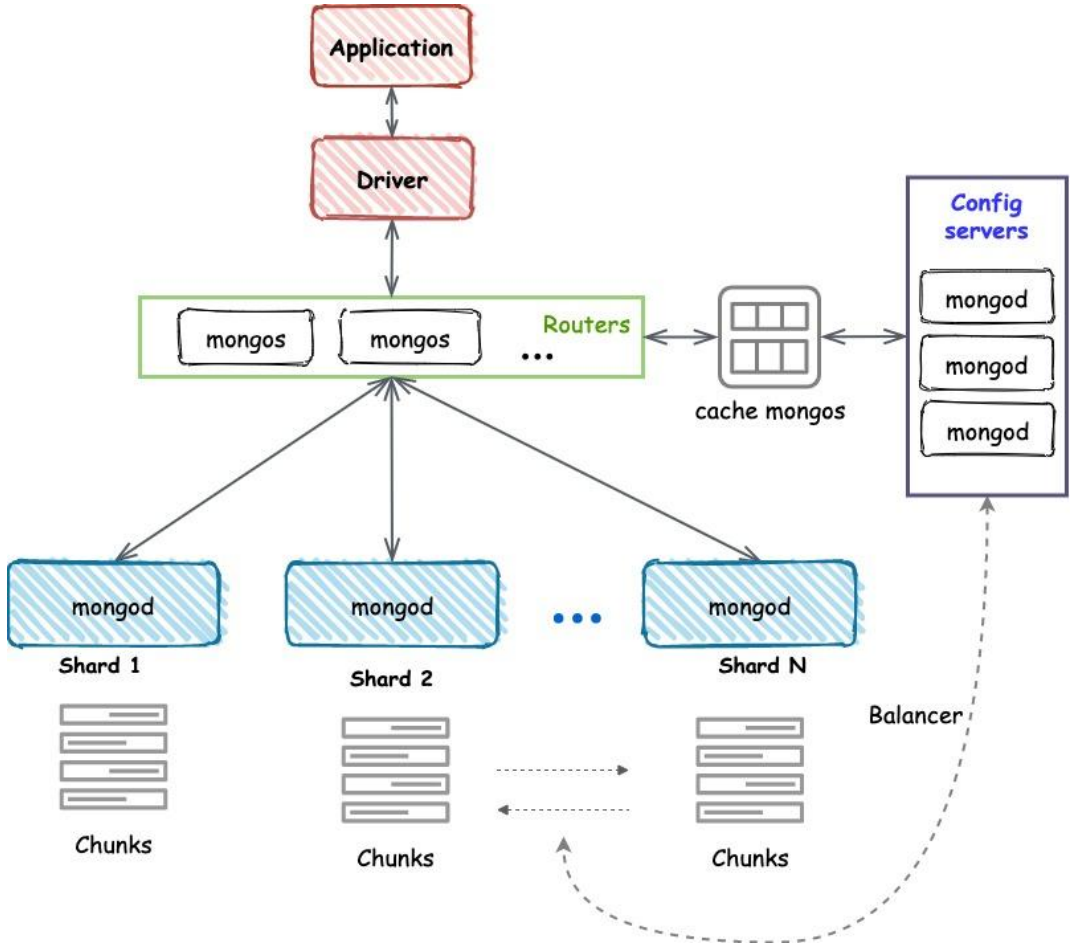


Sharded cluster

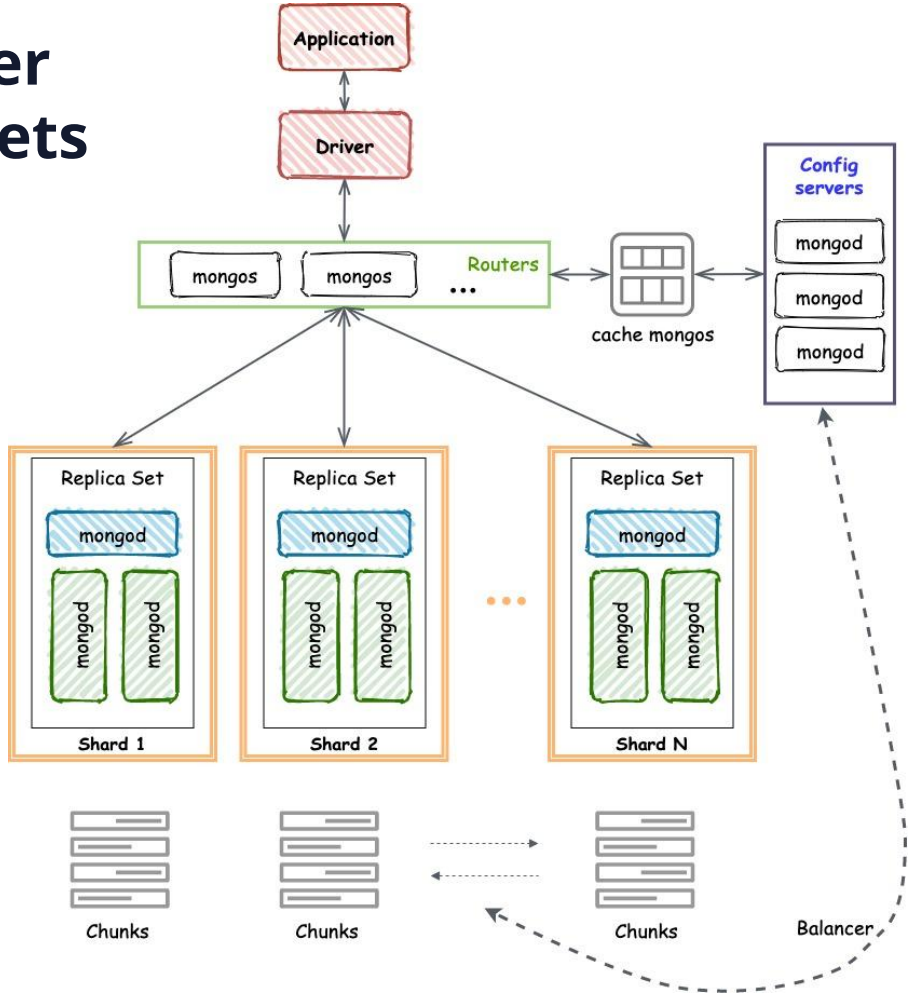
Sharding — это метод для распределения данных по нескольким машинам.

Config Server — хранилище метаданных.

Router (mongos) — маршрутизация запросов, кэширование метаданных и запуск балансера.



Sharded cluster with Replica Sets



ZABBIX MEETUP ONLINE '21

mongoDB plugin для Zabbix agent 2

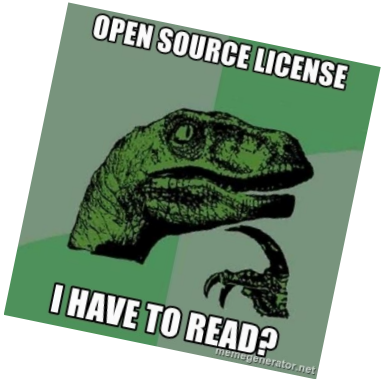
Возможности

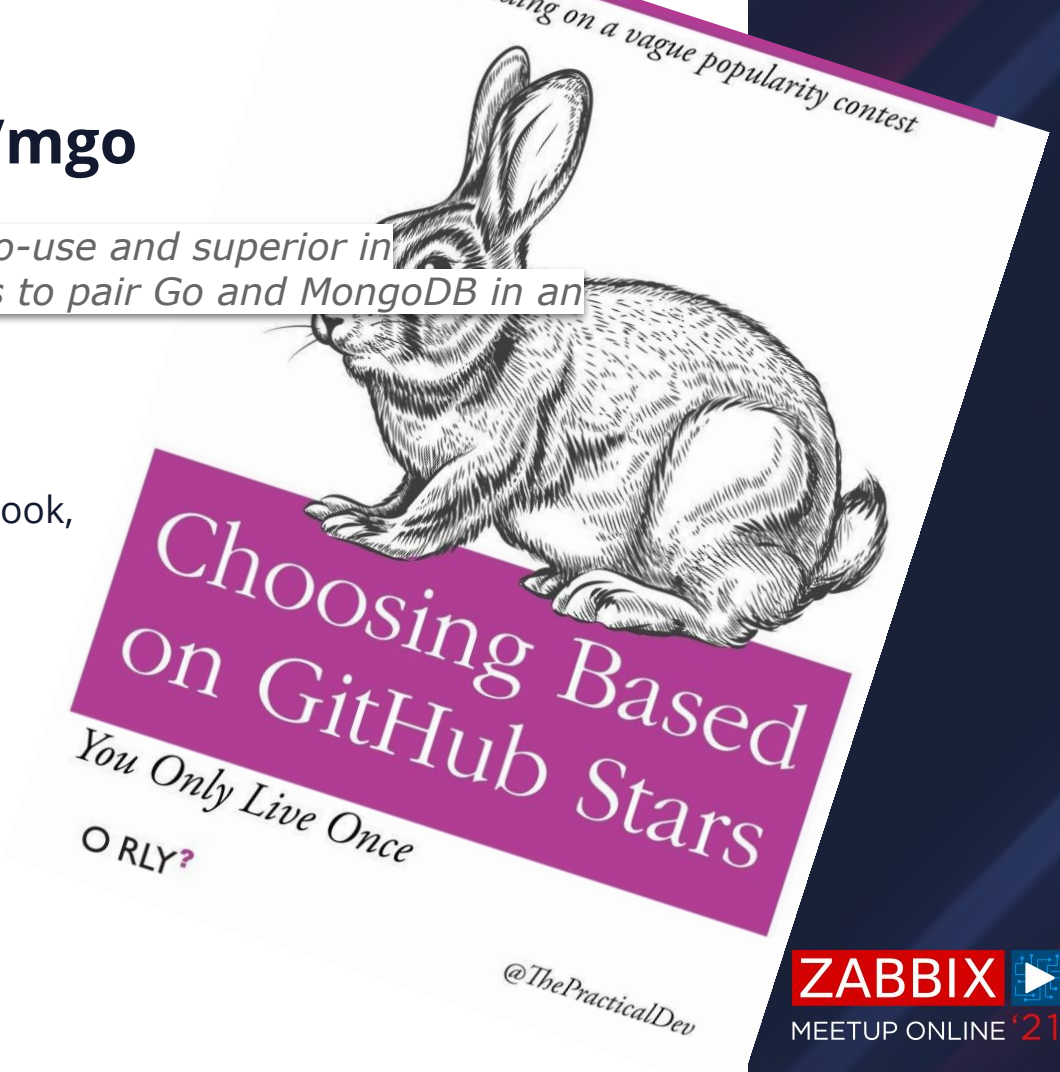
- Написан на Go
- Нет сторонних зависимостей
- mongoDB versions 4.4, 4.2, 4.0 & 3.6
- Standalone, Sharding, ReplicaSet
- Автоматическое обнаружение БД, коллекций, шардов и конфиг серверов
- Один плагин ⇔ несколько инстансов (как локально, так и удалённо)
- Постоянные соединения
- Метрики сгруппированы в JSON, отдаются пачками
- Поддержка IPv6
- Пассивный и активный режим агента
- Гибкие интервалы опроса
- Работает на Linux, MacOS и Windows
- Доступен в Zabbix 5.0.10+

Драйвер — globalsign/mgo

"mgo is the dream driver: it's easy-to-use and superior in performance all the while allowing us to pair Go and MongoDB in an idiomatic fashion"
— Naitik Shah, Parse team at Facebook

- Simplified BSD License
- Использовался в Canonical, Facebook,
  в тулинге самой Mongo





Конфигурация

- Timeout
- Поле maxTimeMS и метод SetMaxTime()
- KeepAlive
- Sessions

Конфигурация: именованные сессии

- Timeout
- Поле maxTimeMS и метод SetMaxTime()
- KeepAlive
- Sessions

Plugins.Mongo.Sessions.**Prod**.Uri=tcp://192.168.1.1:27017
Plugins.Mongo.Sessions.**Prod**.User=<UserForProd>
Plugins.Mongo.Sessions.**Prod**.Password=<PasswordForProd>

Plugins.Mongo.Sessions.**Test**.Uri=tcp://192.168.0.1:27017
Plugins.Mongo.Sessions.**Test**.User=<UserForTest>
Plugins.Mongo.Sessions.**Test**.Password=<PasswordForTest>

Конфигурация: именованные сессии

- Timeout
- Поле maxTimeMS и метод SetMaxTime()
- KeepAlive
- Sessions

Plugins.Mongo.Sessions.**Prod**.Uri=tcp://192.168.1.1:27017
Plugins.Mongo.Sessions.**Prod**.User=<UserForProd>
Plugins.Mongo.Sessions.**Prod**.Password=<PasswordForProd>

Plugins.Mongo.Sessions.**Test**.Uri=tcp://192.168.0.1:27017
Plugins.Mongo.Sessions.**Test**.User=<UserForTest>
Plugins.Mongo.Sessions.**Test**.Password=<PasswordForTest>

mongodb.ping[192.168.1.1:27017,UserForProd,PasswordForProd]
mongodb.ping[192.168.0.1:27017,UserForTest,PasswordForTest]

mongodb.ping[**Prod**]
mongodb.ping[**Test**]

Конфигурация: mongoDB

- **Standalone**: для каждой отдельной ноды mongoDB
- **ReplicaSet**: на Primary ноде
- **Sharded cluster:**
  - на каждом шарде кластера (в случае ReplicaSet — достаточно на Primary)
  - на mongos роутере (автоматически распространится на config серверы)

```
use admin

db.auth("admin", "<ADMIN_PASSWORD>")

db.createUser({
  "user": "zabbix",
  "pwd": "<PASSWORD>",
  "roles": [
    { role: "readAnyDatabase", db: "admin" },
    { role: "clusterMonitor", db: "admin" },
  ]
})
```

Отладка

DebugLevel=5

```
021/03/10 22:45:11.490482 [Mongo] Socket 0xc00040a620 to 192.168.7.189:27133: serializing op: &mgo.getMoreOp{collection:"test_db.$cmd.l
rId:2938487257334988488, replyFunc:(mgo.replyFunc)(0x4417420)}
021/03/10 22:45:11.490488 [Mongo] Socket 0xc00040a620 to 192.168.7.189:27133: sending 1 op(s) (61 bytes)
021/03/10 22:45:11.490500 [Mongo] Socket 0xc00040a620 to 192.168.7.189:27133: updated write deadline to 3s ahead (2021-03-10 22:45:14.4
7)
021/03/10 22:45:11.490528 [Mongo] Socket 0xc00040a620 to 192.168.7.189:27133: updated read deadline to 3s ahead (2021-03-10 22:45:14.49
)
021/03/10 22:45:11.491206 [Mongo] Socket 0xc00040a620 to 192.168.7.189:27133: got reply (1332 bytes)
021/03/10 22:45:11.491263 [Mongo] Socket 0xc00040a620 to 192.168.7.189:27133: received document: bson.M{"idIndex":bson.M{"key":bson.M{"
 "info":bson.M{"readOnly":false, "uuid":bson.Binary{Kind:0x4, Data:[]uint8{0x11, 0x68, 0xd9, 0x23, 0x4, 0xc8, 0x45, 0x12, 0xb3, 0x9b, 0
2a}}}, "name":"HISTORY", "options":bson.M{}, "type":"collection"}
021/03/10 22:45:11.491271 [Mongo] Iter 0xc000016d20 received reply document 1/8 (cursor=0)
021/03/10 22:45:11.491329 [Mongo] Socket 0xc00040a620 to 192.168.7.189:27133: received document: bson.M{"idIndex":bson.M{"key":bson.M{"
 "info":bson.M{"readOnly":false, "uuid":bson.Binary{Kind:0x4, Data:[]uint8{0x2d, 0x67, 0x96, 0x69, 0xe5, 0x1, 0x43, 0x1, 0xaf, 0x83, 0x
}}}, "name":"STOCK", "options":bson.M{}, "type":"collection"}
021/03/10 22:45:11.491341 [Mongo] Iter 0xc000016d20 document unmarshaled: &struct { Name string }{Name:"HISTORY"}
021/03/10 22:45:11.491347 [Mongo] Iter 0xc000016d20 received reply document 2/8 (cursor=0)
021/03/10 22:45:11.491360 [Mongo] Iter 0xc000016d20 document unmarshaled: &struct { Name string }{Name:"STOCK"}
021/03/10 22:45:11.491385 [Mongo] Socket 0xc00040a620 to 192.168.7.189:27133: received document: bson.M{"idIndex":bson.M{"key":bson.M{"
 "info":bson.M{"readOnly":false, "uuid":bson.Binary{Kind:0x4, Data:[]uint8{0x49, 0x9c, 0xdb, 0x74, 0xc2, 0x97, 0x41, 0xb9, 0xbe, 0xba,
x9}}}, "name":"ITEM", "options":bson.M{}, "type":"collection"}
021/03/10 22:45:11.491392 [Mongo] Iter 0xc000016d20 received reply document 3/8 (cursor=0)
021/03/10 22:45:11.491426 [Mongo] Socket 0xc00040a620 to 192.168.7.189:27133: received document: bson.M{"idIndex":bson.M{"key":bson.M{"
 "info":bson.M{"readOnly":false, "uuid":bson.Binary{Kind:0x4, Data:[]uint8{0x5a, 0x2f, 0x3, 0x4a, 0xea, 0x38, 0x44, 0x39, 0xb5, 0xb7, 0
6a}}}, "name":"NEW_ORDER", "options":bson.M{}, "type":"collection"}
021/03/10 22:45:11.491442 [Mongo] Iter 0xc000016d20 document unmarshaled: &struct { Name string }{Name:"ITEM"}
021/03/10 22:45:11.491448 [Mongo] Iter 0xc000016d20 received reply document 4/8 (cursor=0)
021/03/10 22:45:11.491460 [Mongo] Iter 0xc000016d20 document unmarshaled: &struct { Name string }{Name:"NEW_ORDER"}
021/03/10 22:45:11.491482 [Mongo] Socket 0xc00040a620 to 192.168.7.189:27133: received document: bson.M{"idIndex":bson.M{"key":bson.M{"
 "info":bson.M{"readOnly":false, "uuid":bson.Binary{Kind:0x4, Data:[]uint8{0x7a, 0x53, 0xdf, 0x7a, 0xe3, 0x25, 0x41, 0xf4, 0xac, 0x34,
6d}}}, "name":"ORDERS", "options":bson.M{}, "type":"collection"}
021/03/10 22:45:11.491489 [Mongo] Iter 0xc000016d20 received reply document 5/8 (cursor=0)
```

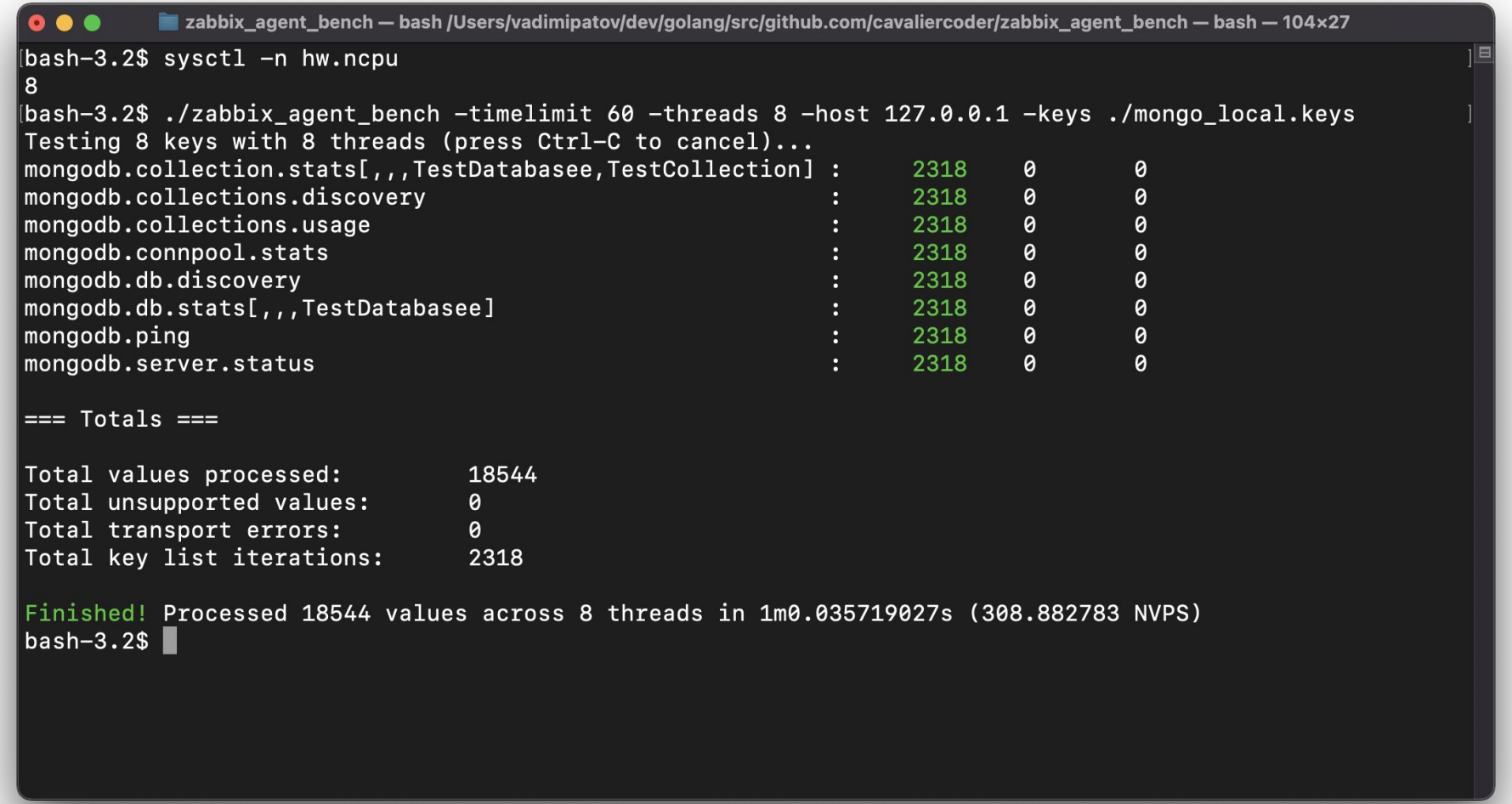
ZABBIX MEETUP ONLINE '21

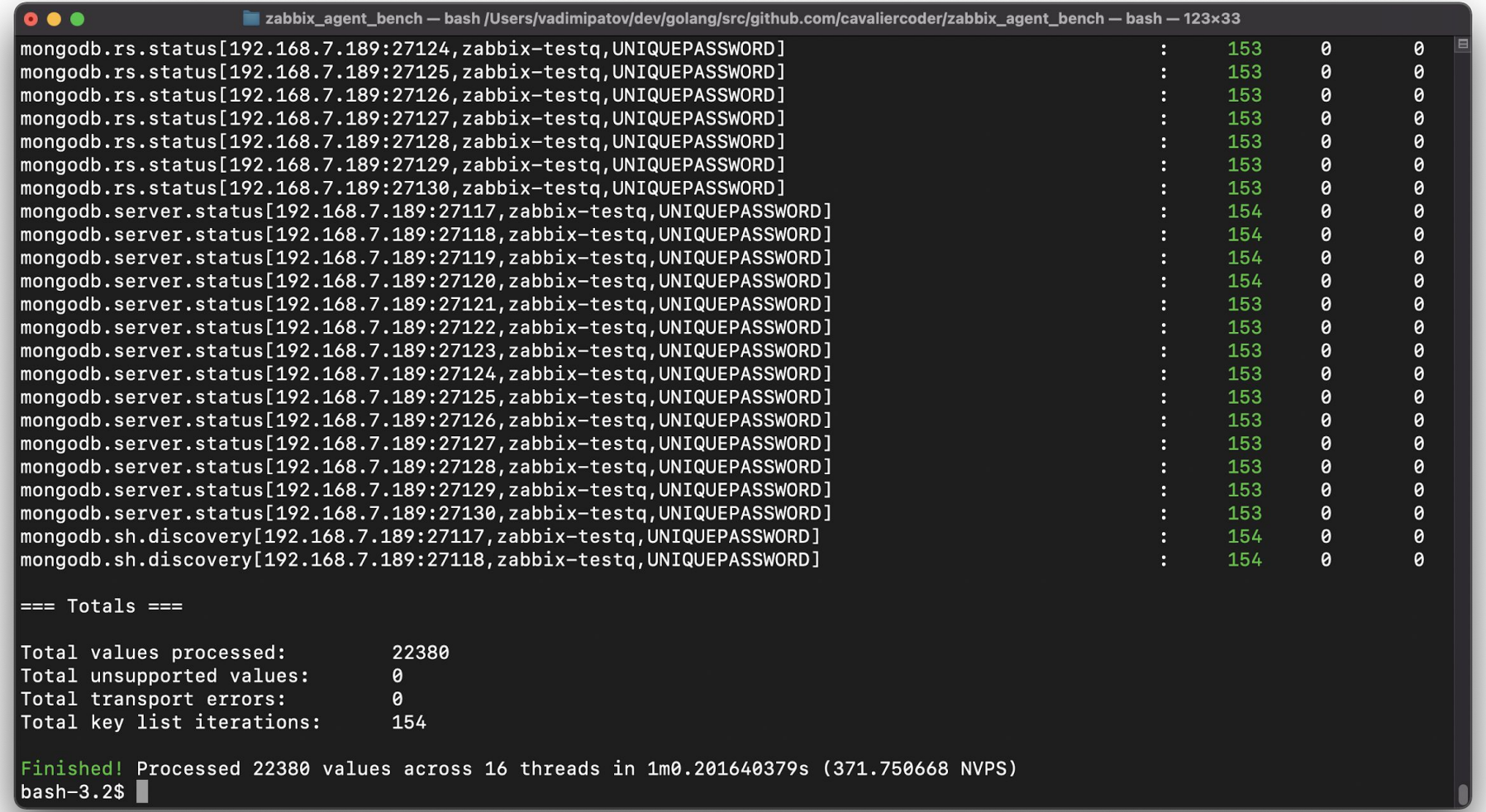
Производительность и бенчмарки

Нагрузочное тестирование

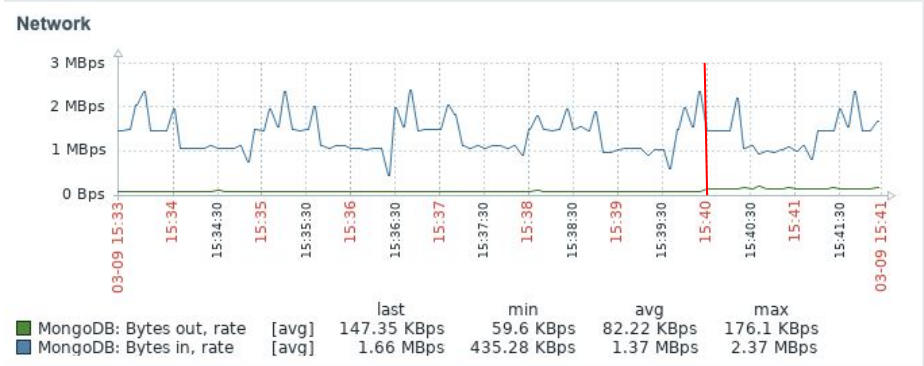
Локальный сетап

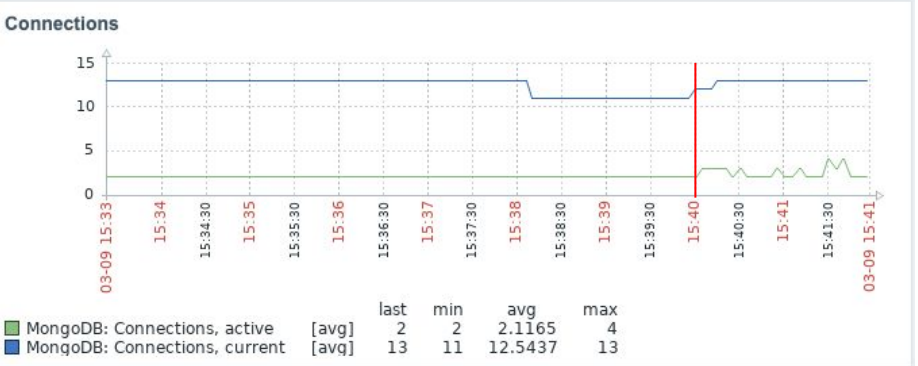
- MacBook Pro 2016
- 2,6 GHz Quad-Core Intel Core i7
- 16GB RAM
- MongoDB 4.4 в Docker
- Бенчмарк
  https://github.com/cavaliercoder/zabbix_agent_bench by Ryan Armstrong


zabbix_agent_bench — bash /Users/vadimipatov/dev/golang/src/github.com/cavaliercoder/zabbix_agent_bench — bash — 104×27
bash-3.2$ sysctl -n hw.ncpu
8
bash-3.2$ ./zabbix_agent_bench -timelimit 60 -threads 8 -host 127.0.0.1 -keys ./mongo_local.keys
Testing 8 keys with 8 threads (press Ctrl-C to cancel)...
mongodb.collection.stats[,,,TestDatabasee,TestCollection] : 2318 0 0
mongodb.collections.discovery : 2318 0 0
mongodb.collections.usage : 2318 0 0
mongodb.connpool.stats : 2318 0 0
mongodb.db.discovery : 2318 0 0
mongodb.db.stats[,,,TestDatabasee] : 2318 0 0
mongodb.ping : 2318 0 0
mongodb.server.status : 2318 0 0
=== Totals ===
Total values processed: 18544
Total unsupported values: 0
Total transport errors: 0
Total key list iterations: 2318
Finished! Processed 18544 values across 8 threads in 1m0.035719027s (308.882783 NVPS)
bash-3.2$

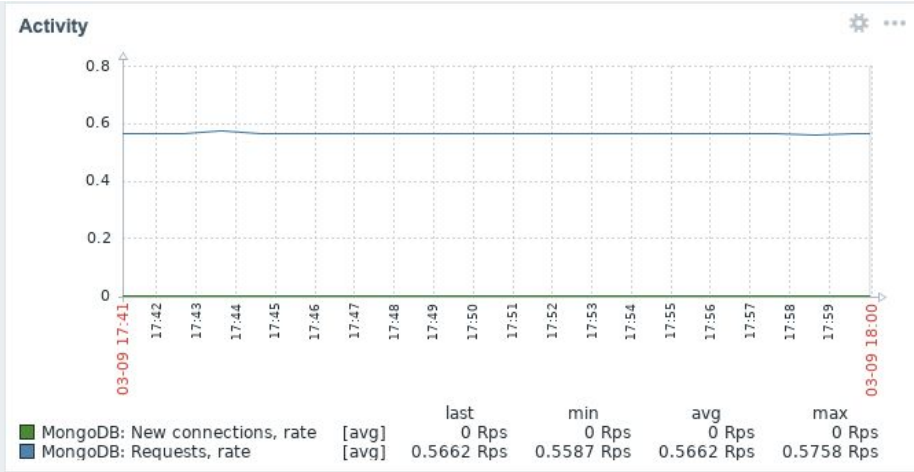

zabbix_agent_bench — bash /Users/vadimipatov/dev/golang/src/github.com/cavaliercoder/zabbix_agent_bench — bash — 123×33
mongodb.rs.status[192.168.7.189:27124,zabbix-testq,UNIQUEPASSWORD] : 153 0 0
mongodb.rs.status[192.168.7.189:27125,zabbix-testq,UNIQUEPASSWORD] : 153 0 0
mongodb.rs.status[192.168.7.189:27126,zabbix-testq,UNIQUEPASSWORD] : 153 0 0
mongodb.rs.status[192.168.7.189:27127,zabbix-testq,UNIQUEPASSWORD] : 153 0 0
mongodb.rs.status[192.168.7.189:27128,zabbix-testq,UNIQUEPASSWORD] : 153 0 0
mongodb.rs.status[192.168.7.189:27129,zabbix-testq,UNIQUEPASSWORD] : 153 0 0
mongodb.rs.status[192.168.7.189:27130,zabbix-testq,UNIQUEPASSWORD] : 153 0 0
mongodb.server.status[192.168.7.189:27117,zabbix-testq,UNIQUEPASSWORD] : 154 0 0
mongodb.server.status[192.168.7.189:27118,zabbix-testq,UNIQUEPASSWORD] : 154 0 0
mongodb.server.status[192.168.7.189:27119,zabbix-testq,UNIQUEPASSWORD] : 154 0 0
mongodb.server.status[192.168.7.189:27120,zabbix-testq,UNIQUEPASSWORD] : 154 0 0
mongodb.server.status[192.168.7.189:27121,zabbix-testq,UNIQUEPASSWORD] : 153 0 0
mongodb.server.status[192.168.7.189:27122,zabbix-testq,UNIQUEPASSWORD] : 153 0 0
mongodb.server.status[192.168.7.189:27123,zabbix-testq,UNIQUEPASSWORD] : 153 0 0
mongodb.server.status[192.168.7.189:27124,zabbix-testq,UNIQUEPASSWORD] : 153 0 0
mongodb.server.status[192.168.7.189:27125,zabbix-testq,UNIQUEPASSWORD] : 153 0 0
mongodb.server.status[192.168.7.189:27126,zabbix-testq,UNIQUEPASSWORD] : 153 0 0
mongodb.server.status[192.168.7.189:27127,zabbix-testq,UNIQUEPASSWORD] : 153 0 0
mongodb.server.status[192.168.7.189:27128,zabbix-testq,UNIQUEPASSWORD] : 153 0 0
mongodb.server.status[192.168.7.189:27129,zabbix-testq,UNIQUEPASSWORD] : 153 0 0
mongodb.server.status[192.168.7.189:27130,zabbix-testq,UNIQUEPASSWORD] : 153 0 0
mongodb.sh.discovery[192.168.7.189:27117,zabbix-testq,UNIQUEPASSWORD] : 154 0 0
mongodb.sh.discovery[192.168.7.189:27118,zabbix-testq,UNIQUEPASSWORD] : 154 0 0
=== Totals ===
Total values processed: 22380
Total unsupported values: 0
Total transport errors: 0
Total key list iterations: 154
Finished! Processed 22380 values across 16 threads in 1m0.201640379s (371.750668 NVPS)
bash-3.2$

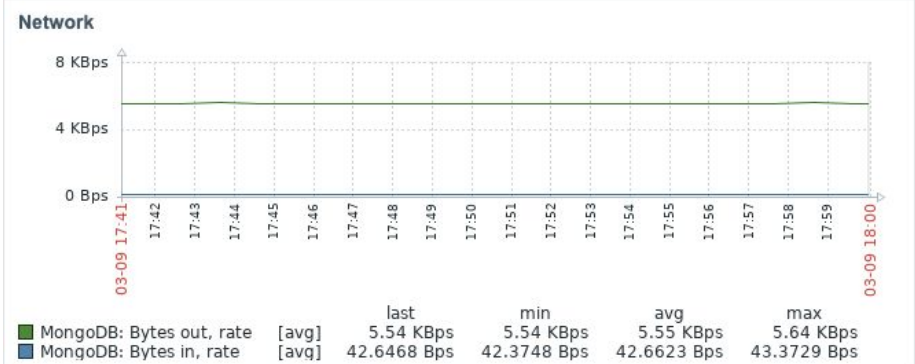
Эффект наблюдателя

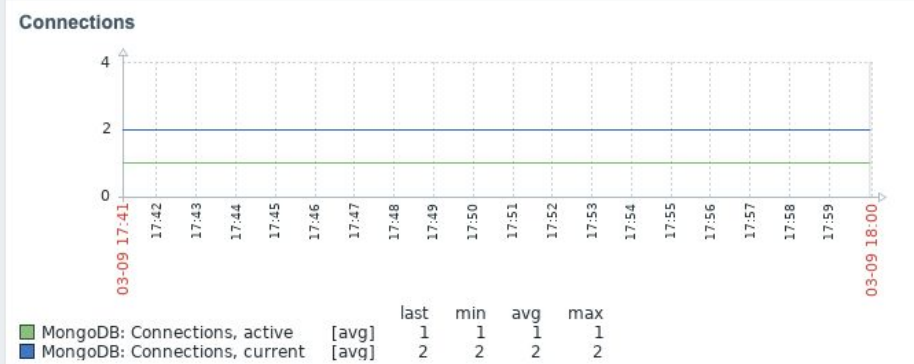




Эффект наблюдателя







Немного про наш workflow

1

2

3

PROFIT!

ZABBIX MEETUP ONLINE '21

Template mongoDB by Zabbix agent 2

Два шаблона

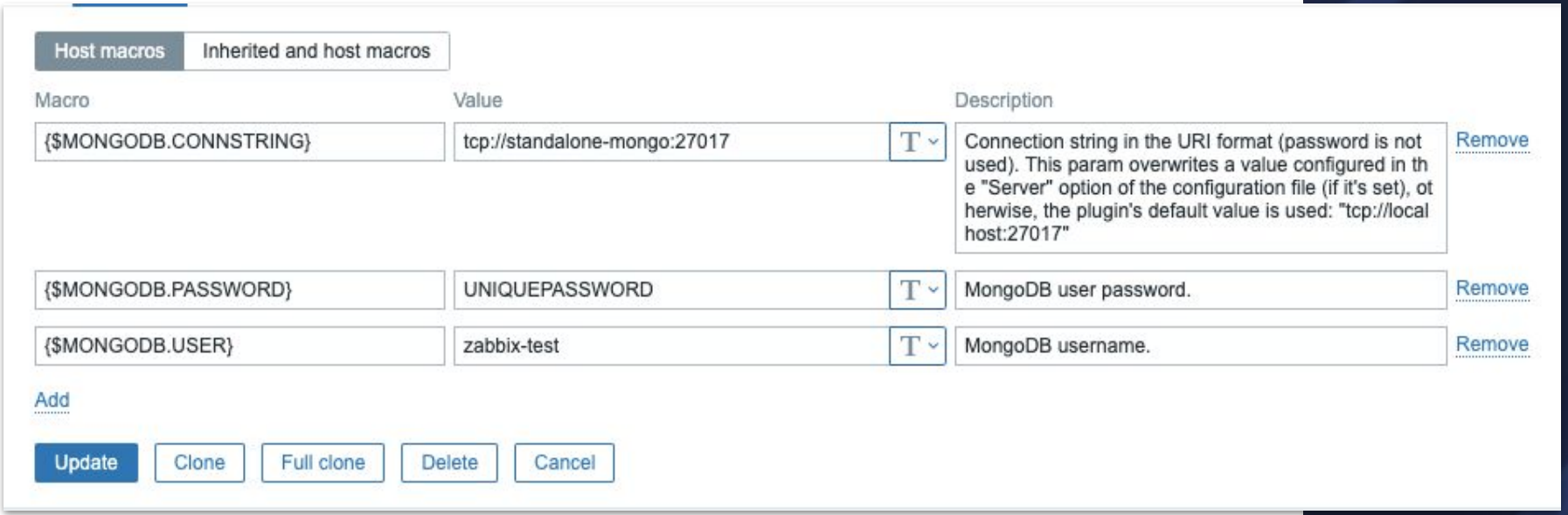
- MongoDB node by Zabbix Agent 2
- MongoDB cluster by Zabbix Agent 2

Два шаблона

- MongoDB node by Zabbix Agent 2
- MongoDB cluster by Zabbix Agent 2

mongoDB command	Zabbix agent key	mongod	mongos
db.runCommand({ serverStatus: 1 })	mongodb.server.status		
db.runCommand("ping").ok	mongodb.ping		
db.adminCommand("top")	mongodb.collections.usage		
db.runCommand({ replSetGetStatus: 1 })	mongodb.rs.status		
db.runCommand({ dbStats: 1 })	mongodb.db.stats		
db.runCommand({collStats:"<coll_name>"})	mongodb.collection.stats		
db.runCommand({ connPoolStats: 1 })	mongodb.connpool.stats		
db.getCollection('chunks').find({'jumbo': true}).count()	mongodb.jumbo_chunks.count		
db.getCollection('oplog.rs').find({"ts": {"$exists": 1}}).sort({"$natural":-1}).limit(1)	mongodb.oplog.stats		
db.adminCommand({ replSetGetConfig: 1, commitmentStatus: true });	mongodb.rs.config		
db.getCollection('shards').find({ })	mongodb.sh.discovery		
db.adminCommand("getShardMap")	mongodb.cfg.discovery		
db.adminCommand({ listDatabases: 1 })	mongodb.db.discovery		
db.runCommand({ listCollections: 1.0 })	mongodb.collections.discovery		

Настройки шаблонов



ZABBIX MEETUP ONLINE '21

MongoDB node

он же шаблон для mongod

Шаблон MongoDB node

- > 100 метрик для mongod

- Обнаружение:
  - Баз данных
  - Коллекций

- Обнаружение метрик по:
  - Базам
  - Коллекциям
  - Репликации
  - WiredTiger

Алерты

- **Недоступность ноды**

Алерты

- Недоступность ноды
- **Задержка репликации**

Алерты

- Недоступность ноды
- Задержка репликации
- **Количество нод в ReplicaSet в статусе unhealthy**

Алерты

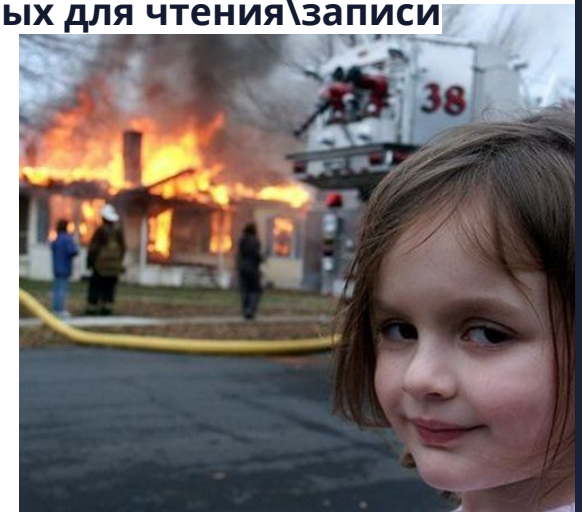
- Недоступность ноды
- Задержка репликации
- Количество нод в ReplicaSet в статусе unhealthy
- **Изменение состояния ноды в ReplicaSet**

Алерты

- Недоступность ноды
- Задержка репликации
- Количество нод в ReplicaSet в статусе unhealthy
- Изменение состояния ноды в ReplicaSet
- **Количество доступных соединений**

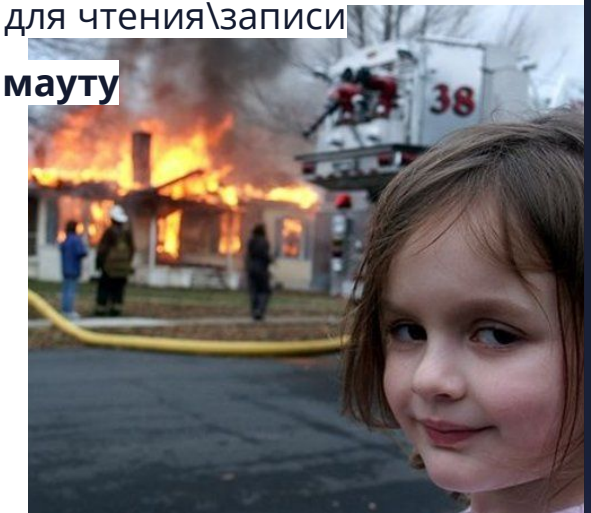
Алерты

- Недоступность ноды
- Задержка репликации
- Количество нод в ReplicaSet в статусе unhealthy
- Изменение состояния ноды в ReplicaSet
- Количество доступных соединений
- **Количество тикетов WiredTiger доступных для чтения\записи**



Алерты

- Недоступность ноды
- Задержка репликации
- Количество нод в ReplicaSet в статусе unhealthy
- Изменение состояния ноды в ReplicaSet
- Количество доступных соединений
- Количество тикетов WiredTiger доступных для чтения\записи
- **Количество курсоров закрытых по таймауту**



Обнаружение баз данных

```
[
    {
        "{#DBNAME}":"MyDatabase"
    },
    {
        "{#DBNAME}":"admin"
    },
    {
        "{#DBNAME}":"config"
    },
    {
        "{#DBNAME}":"local"
    }
]
```

Обнаружение баз данных

```
[
    {
        "{#DBNAME}":"MyDatabase"
    },
    {
        "{#DBNAME}":"admin"
    },
    {
        "{#DBNAME}":"config"
    },
    {
        "{#DBNAME}":"local"
    }
]
```

{$MONGODB.LLD.FILTER.DB.MATCHES}: “.*”

{$MONGODB.LLD.FILTER.DB.NOT_MATCHES}: “(admin|config|local)"

Обнаружение коллекций

```
[
    {
        "{#COLLECTION}":"system.users",
        "{#DBNAME}":"admin"
    },
    {
        "{#COLLECTION}":"ITEM",
        "{#DBNAME}":"test_db"
    },
    {
        "{#COLLECTION}":"NEW_ORDER",
        "{#DBNAME}":"test_db"
    },
 <...>

]
```

Обнаружение коллекций

```
[
    {
        "{#COLLECTION}":"system.users",
        "{#DBNAME}":"admin"
    },
    {
        "{#COLLECTION}":"ITEM",
        "{#DBNAME}":"test_db"
    },
    {
        "{#COLLECTION}":"NEW_ORDER",
        "{#DBNAME}":"test_db"
    },
 <...>
]
```

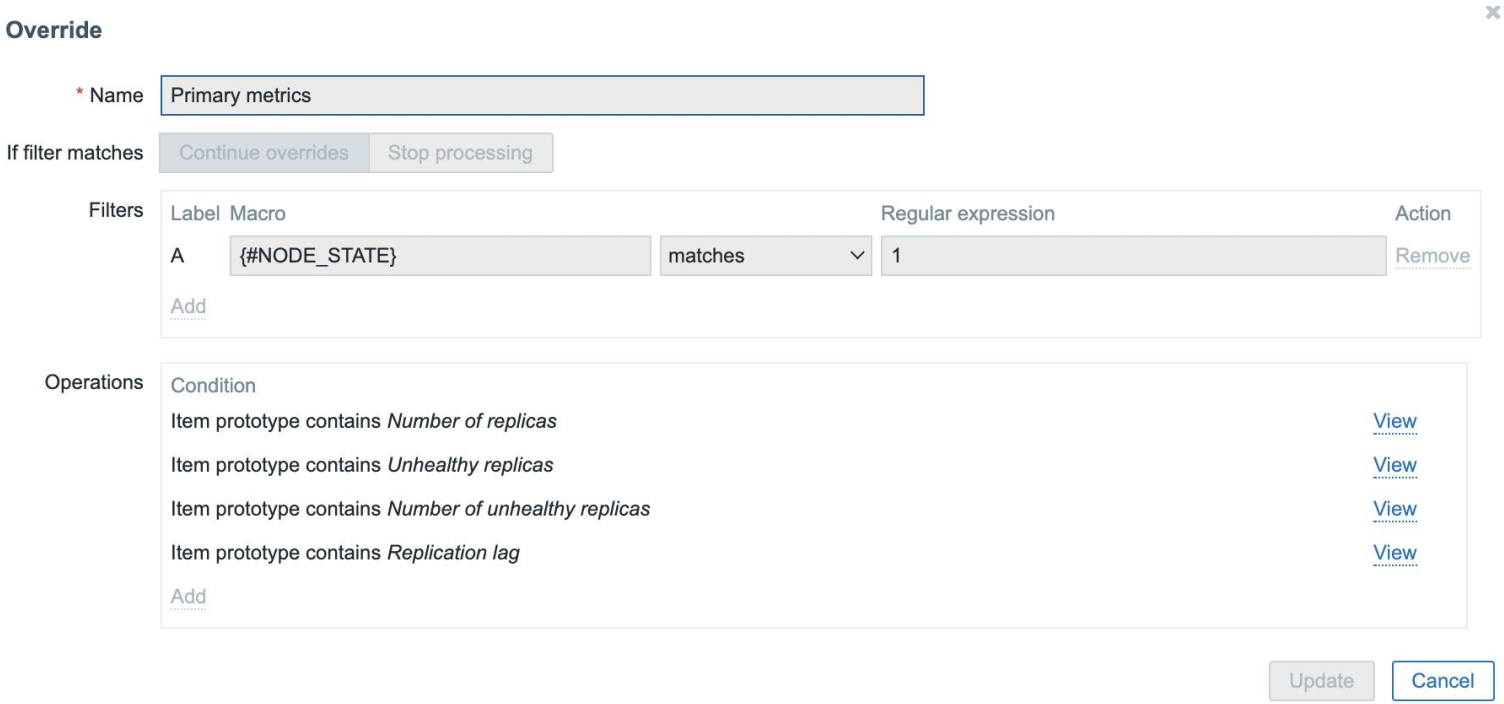
{$MONGODB.LLD.FILTER.COLLECTION.MATCHES}: “.*”

{$MONGODB.LLD.FILTER.COLLECTION.NOT_MATCHES}: “CHANGE_IF_NEEDED”

ReplicaSet мониторинг

```
[
    {
        "{#RS_NAME}":"rs-shard-01",
        "{#NODE_STATE}":2
    }
]
```

ReplicaSet мониторинг



ZABBIX MEETUP ONLINE '21

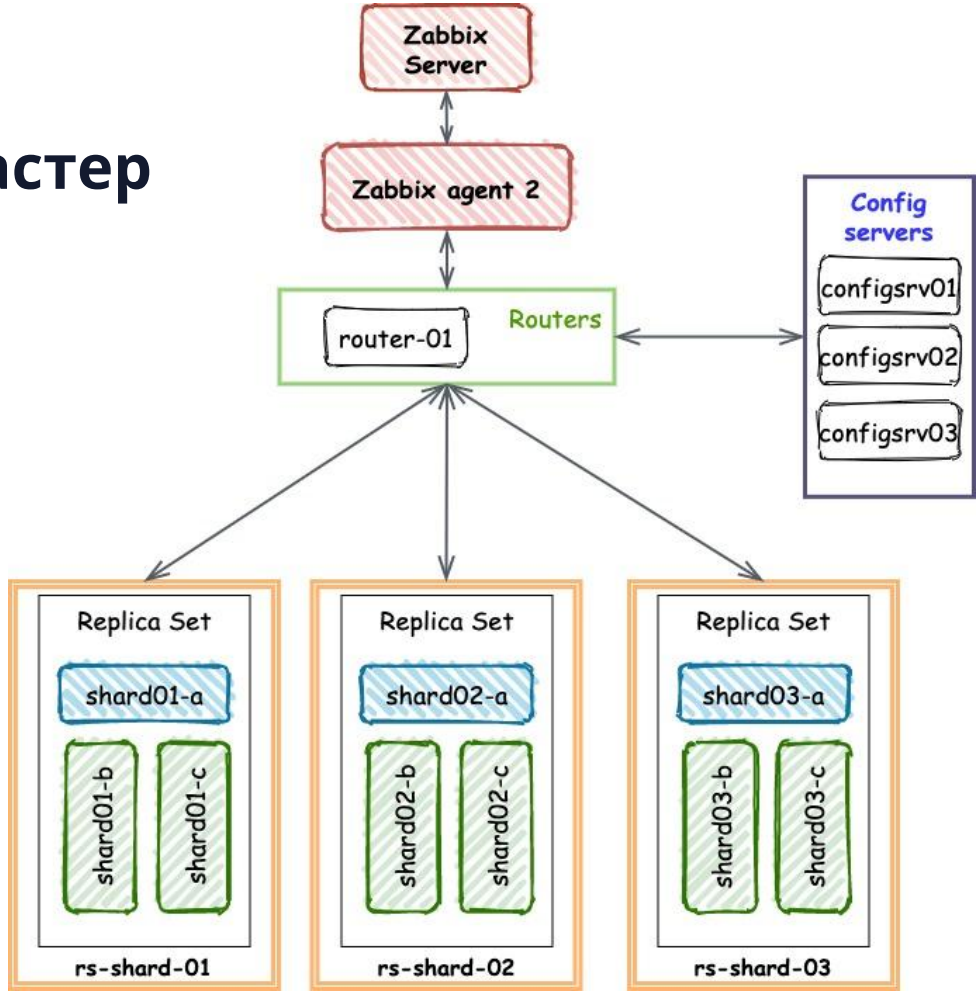
MongoDB cluster

он же шаблон для mongos

Шаблон MongoDB cluster

- > 50 метрик для mongos
- Обнаружение:
  - Баз данных
  - Коллекций
  - Нод в шардированном кластере
  - Конфиг серверов
- Обнаружение метрик по:
  - Базам данных
  - Коллекциям
- Алерты аналогичны шаблону MongoDB node

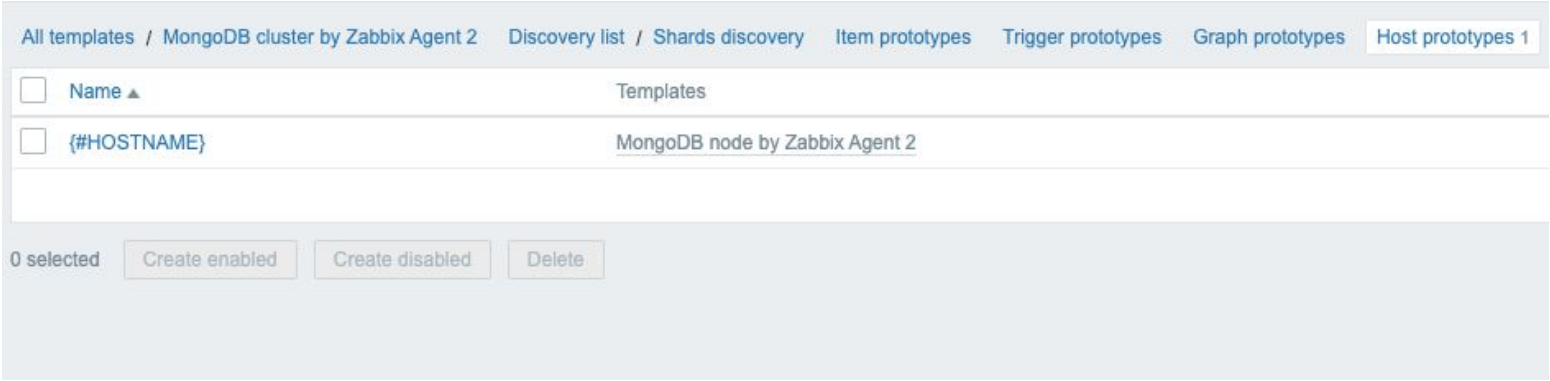
Тестовый кластер mongoDB



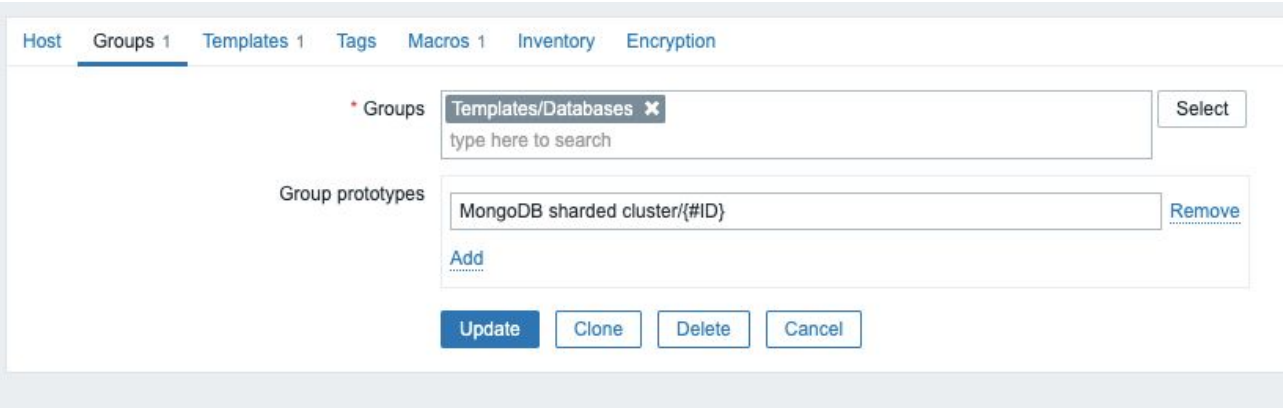
Обнаружение шардов в кластере

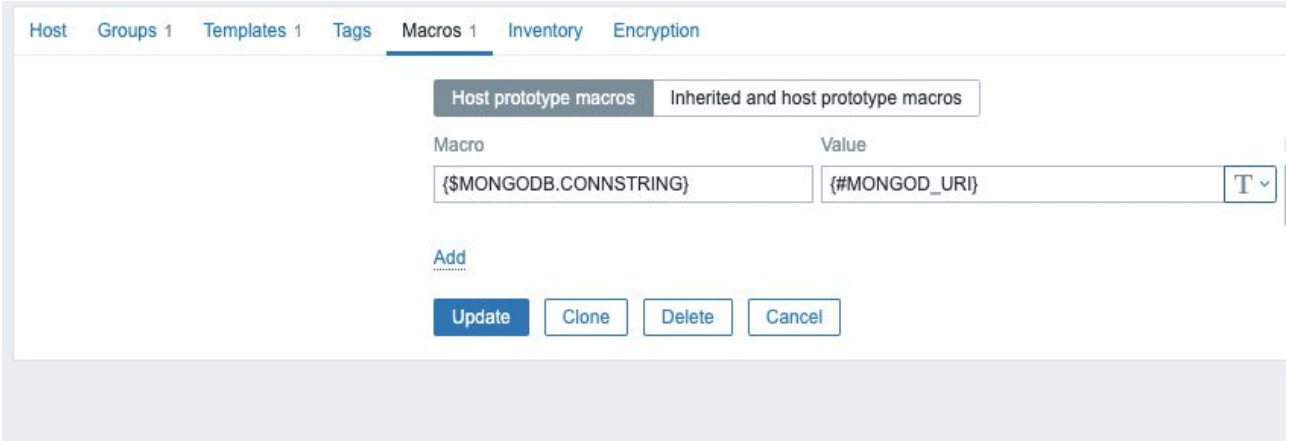
```
[
    {
        "{#ID}": "rs-shard-01",
        "{#HOSTNAME}": "shard01-a",
        "{#MONGOD_URI}": "tcp://shard01-a:27017",
        "{#STATE}": "1"
    },
    {
        "{#ID}": "rs-shard-01",
        "{#HOSTNAME}": "shard01-b",
        "{#MONGOD_URI}": "tcp://shard01-b:27017",
        "{#STATE}": "1"
    }
<...>
]
```

Обнаружение шардов в кластере



Обнаружение шардов в кластере

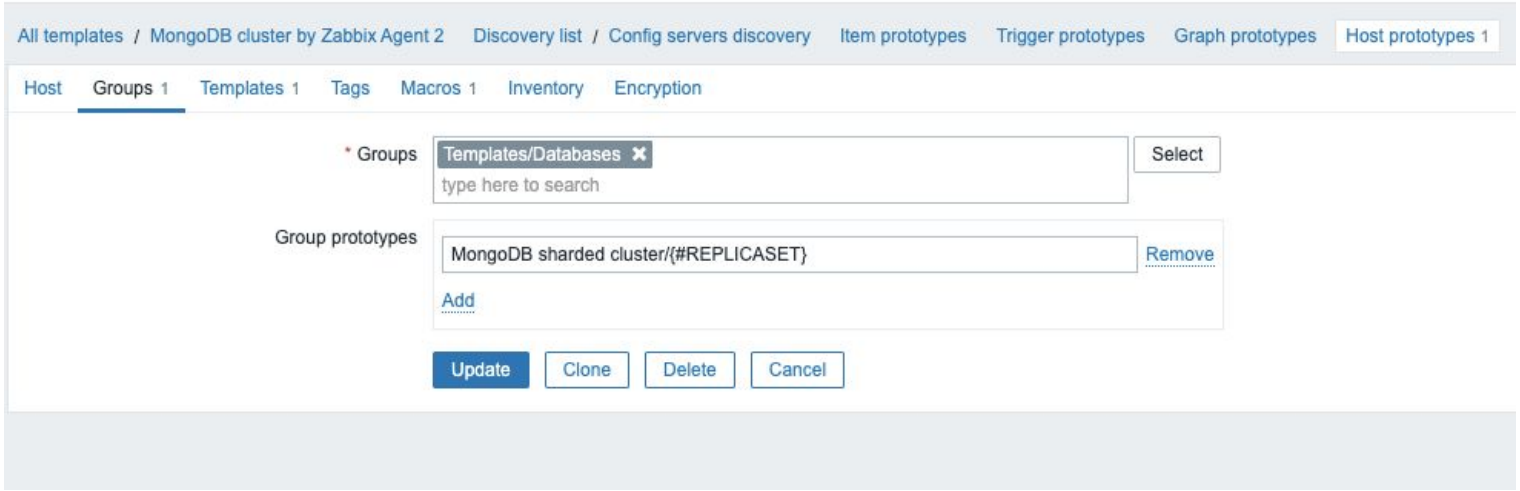




Обнаружение конфиг серверов в кластере

```
[
    {
        "{#REPLICASET}":"rs-config-server",
        "{#HOSTNAME}":"configsvr01",
        "{#MONGOD_URI}":"tcp://configsvr01:27017"
    },
    {
        "{#REPLICASET}":"rs-config-server",
        "{#HOSTNAME}":"configsvr02",
        "{#MONGOD_URI}":"tcp://configsvr02:27017"
    },
<...>
]
```

Обнаружение конфиг серверов в кластере




A FEW
MOMENTS LATER

Результат работы дискавери

Name ▲	Applications	Items	Triggers	Graphs	Discovery	Web	Interface	Proxy	Templates
Config servers discovery: configsvr01	Applications 3	Items 79	Triggers 11	Graphs 10	Discovery 4	Web	agent2_mongo: 10050		Template DB MongoDB node by Zabbix Agent 2
Config servers discovery: configsvr02	Applications 3	Items 79	Triggers 11	Graphs 10	Discovery 4	Web	agent2_mongo: 10050		Template DB MongoDB node by Zabbix Agent 2
Config servers discovery: configsvr03	Applications 3	Items 81	Triggers 11	Graphs 9	Discovery 4	Web	agent2_mongo: 10050		Template DB MongoDB node by Zabbix Agent 2
router-01	Applications 4	Items 51	Triggers 7	Graphs 6	Discovery 4	Web	agent2_mongo: 10050		Template DB MongoDB cluster by Zabbix Agent 2
Shards discovery: shard01-a	Applications 5	Items 115	Triggers 11	Graphs 16	Discovery 4	Web	agent2_mongo: 10050		Template DB MongoDB node by Zabbix Agent 2
Shards discovery: shard01-b	Applications 5	Items 117	Triggers 11	Graphs 15	Discovery 4	Web	agent2_mongo: 10050		Template DB MongoDB node by Zabbix Agent 2
Shards discovery: shard01-c	Applications 5	Items 115	Triggers 11	Graphs 16	Discovery 4	Web	agent2_mongo: 10050		Template DB MongoDB node by Zabbix Agent 2
Shards discovery: shard02-a	Applications 5	Items 117	Triggers 11	Graphs 15	Discovery 4	Web	agent2_mongo: 10050		Template DB MongoDB node by Zabbix Agent 2
Shards discovery: shard02-b	Applications 5	Items 115	Triggers 11	Graphs 16	Discovery 4	Web	agent2_mongo: 10050		Template DB MongoDB node by Zabbix Agent 2
Shards discovery: shard02-c	Applications 5	Items 115	Triggers 11	Graphs 16	Discovery 4	Web	agent2_mongo: 10050		Template DB MongoDB node by Zabbix Agent 2
Shards discovery: shard03-a	Applications 5	Items 115	Triggers 11	Graphs 16	Discovery 4	Web	agent2_mongo: 10050		Template DB MongoDB node by Zabbix Agent 2
Shards discovery: shard03-b	Applications 5	Items 117	Triggers 11	Graphs 15	Discovery 4	Web	agent2_mongo: 10050		Template DB MongoDB node by Zabbix Agent 2
Shards discovery: shard03-c	Applications 5	Items 115	Triggers 11	Graphs 16	Discovery 4	Web	agent2_mongo: 10050		Template DB MongoDB node by Zabbix Agent 2

Результат работы дискавери

☐	mongoDB	Hosts 1	Templates	router-01
☐	Config servers discovery: MongoDB sharded cluster/rs-config-server	Hosts 3	Templates	configsvr01, configsvr02, configsvr03
☐	Shards discovery: MongoDB sharded cluster/rs-shard-01	Hosts 3	Templates	shard01-a, shard01-b, shard01-c
☐	Shards discovery: MongoDB sharded cluster/rs-shard-02	Hosts 3	Templates	shard02-a, shard02-b, shard02-c
☐	Shards discovery: MongoDB sharded cluster/rs-shard-03	Hosts 3	Templates	shard03-a, shard03-b, shard03-c

ZABBIX MEETUP ONLINE '21

ZABBIX MEETUP ONLINE '21

Q&A