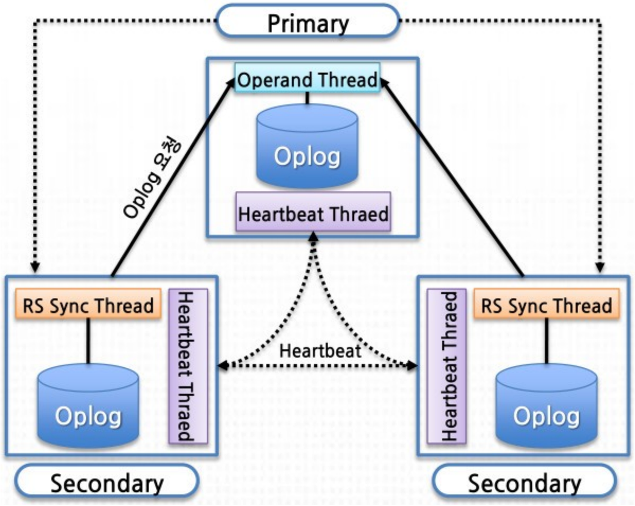
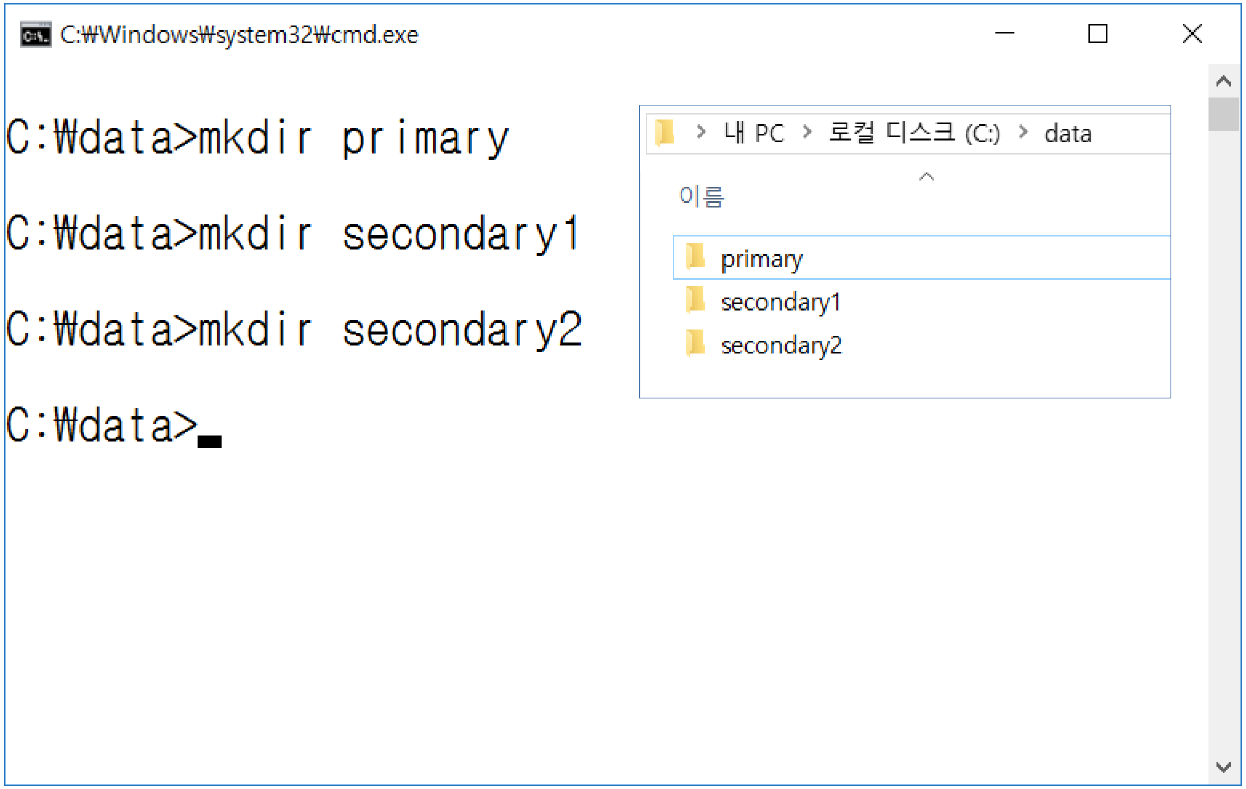
■ ReplicaSet

- 한개의 Primary 와 두개의 Secondary로 구성
- 구성된 각 노드는 자신을 제외한 다른 노드들이 동작하는지 Heartbeat를 이용하여 주기적으로 검사
- Heartbeat를 받은 서버는 자신의 상태 코드를 요청한 서버에게 전송
- Secondary가 사용할 수 없는 상태가 되면 데이터 복제 중단하며 Primary는 데이터 수신/저장을 계속 담당
- Secondary가 복구되면 자동으로 동기화
- Primary 서버에 장애가 발생되면 Secondary 서버를 Primary 서버로 만듬



■ ReplicaSet 시스템 구성

● Primary 및 Secondary 데이터 저장 디렉토리 생성



C:₩Windows₩system32₩cmd.exe

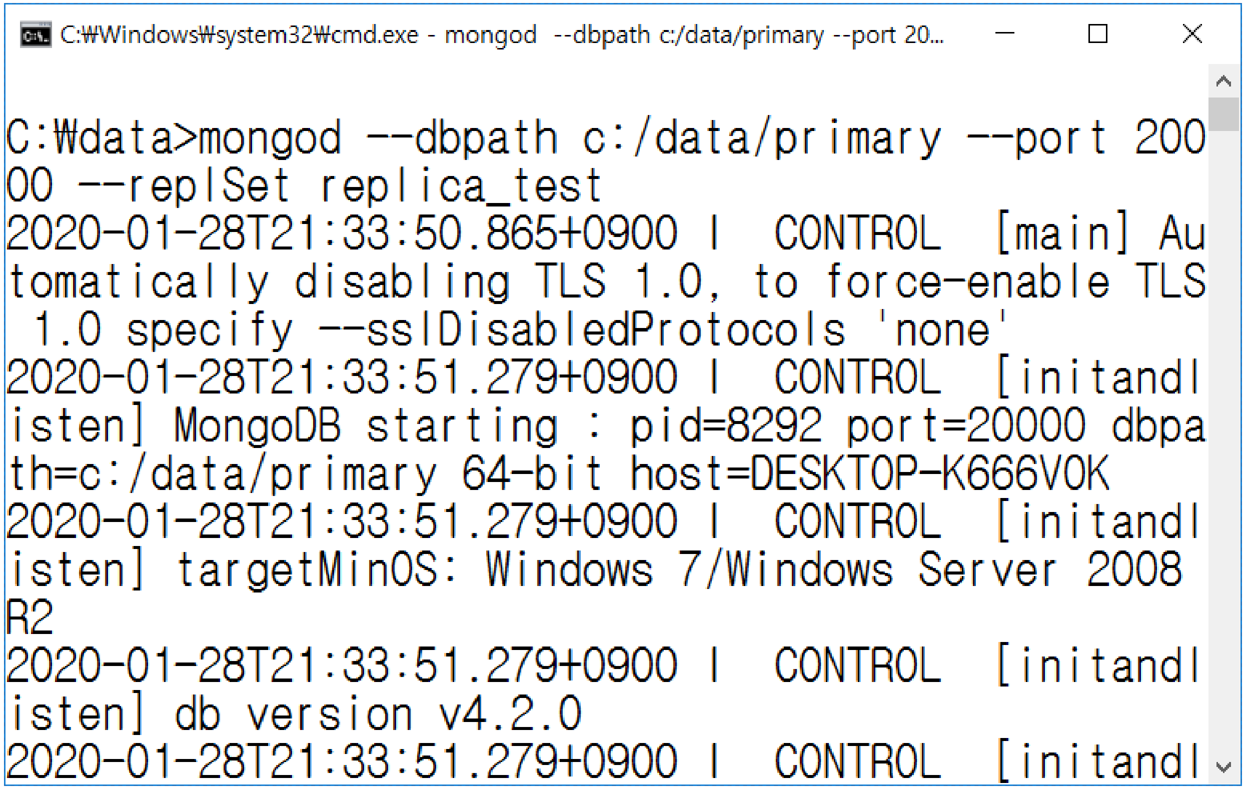
```
C:₩data>mkdir primary

C:₩data>mkdir secondary1

C:₩data>mkdir secondary2

C:₩data>_
```

> 내 PC > 로컬 디스크 (C:) > data
이름

- primary
- secondary1
- secondary2

■ ReplicaSet 시스템 구성

● Primary 서버 실행



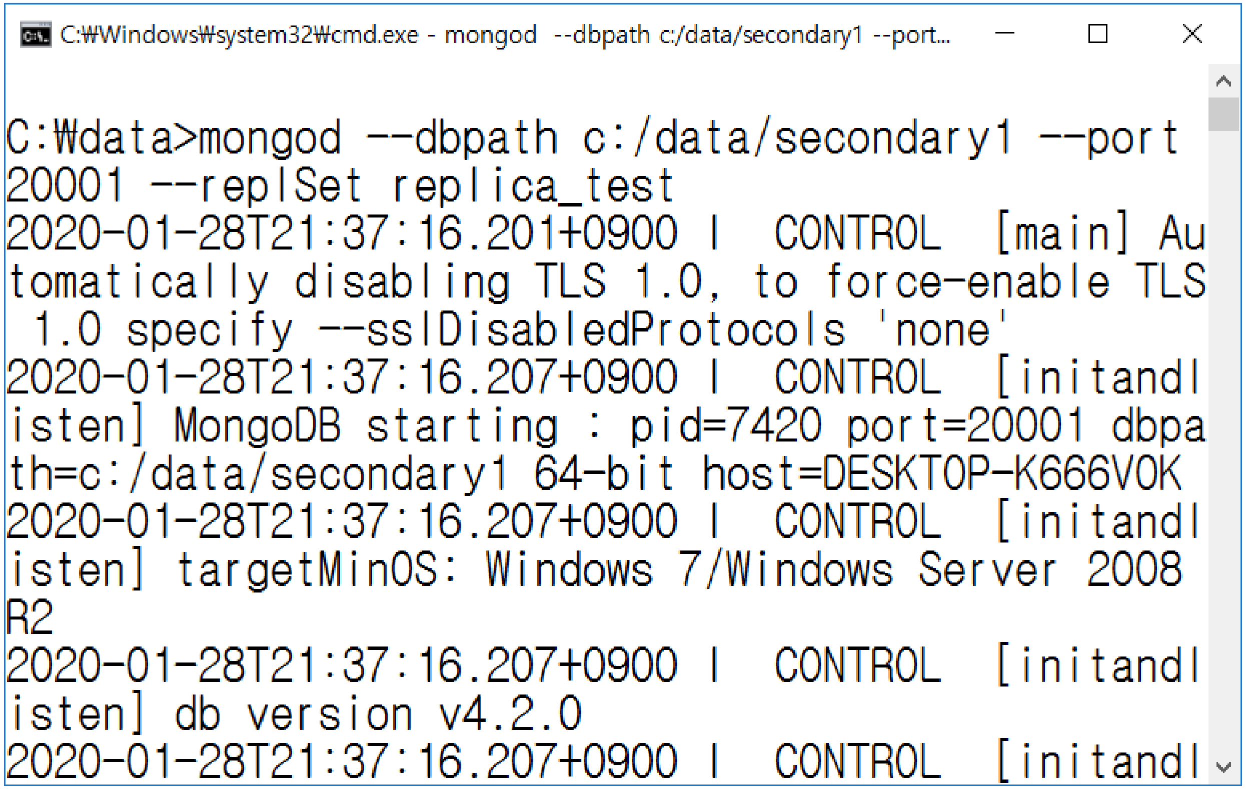
C:₩Windows₩system32₩cmd.exe - mongod --dbpath c:/data/primary --port 20...

```
C:₩data>mongod --dbpath c:/data/primary --port 200
00 --replSet replica_test
2020-01-28T21:33:50.865+0900 I  CONTROL  [main] Au
tomatically disabling TLS 1.0, to force-enable TLS
 1.0 specify --sslDisabledProtocols 'none'
2020-01-28T21:33:51.279+0900 I  CONTROL  [initandl
isten] MongoDB starting : pid=8292 port=20000 dbpa
th=c:/data/primary 64-bit host=DESKTOP-K666V0K
2020-01-28T21:33:51.279+0900 I  CONTROL  [initandl
isten] targetMinOS: Windows 7/Windows Server 2008
R2
2020-01-28T21:33:51.279+0900 I  CONTROL  [initandl
isten] db version v4.2.0
2020-01-28T21:33:51.279+0900 I  CONTROL  [initandl
```

```
mongod --dbpath c:/data/primary --port 20000 --replSet replica_test
```

■ ReplicaSet 시스템 구성

● Secondary1 서버 실행



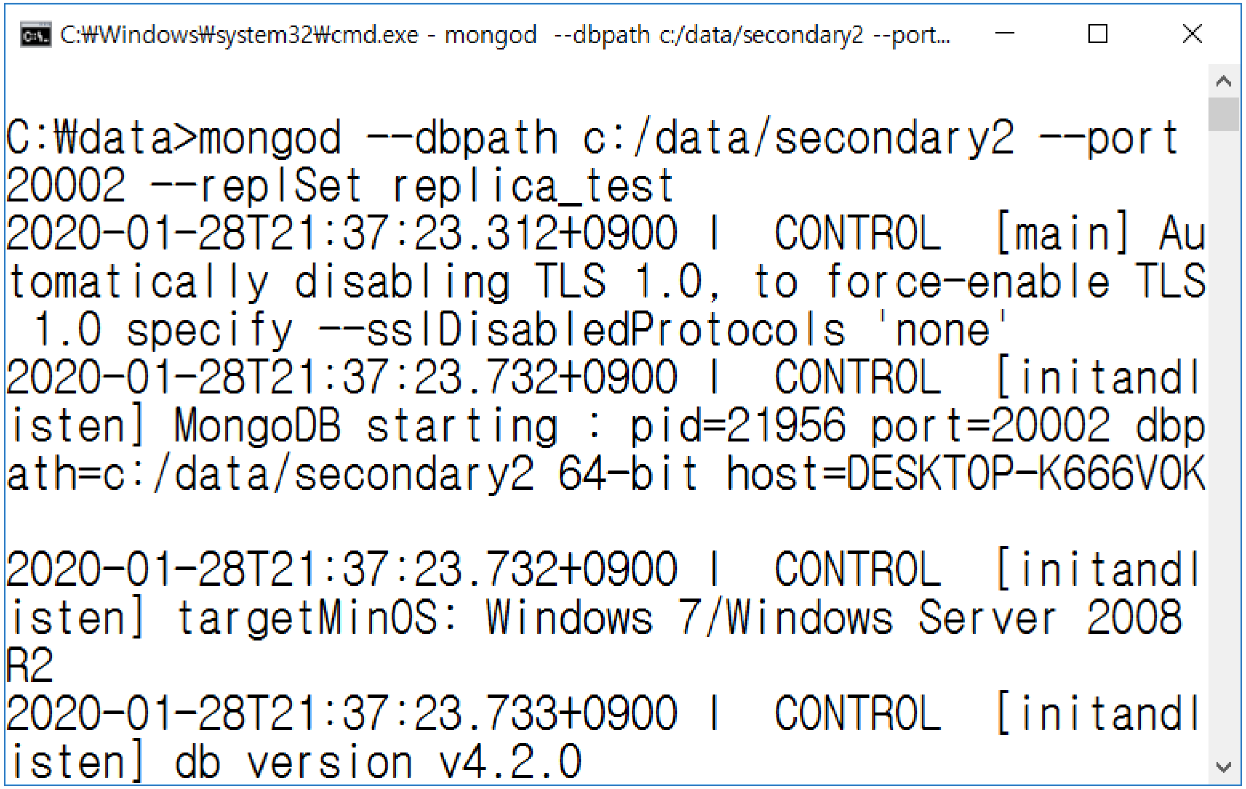
C:₩Windows₩system32₩cmd.exe - mongod --dbpath c:/data/secondary1 --port...

```
C:₩data>mongod --dbpath c:/data/secondary1 --port
20001 --replSet replica_test
2020-01-28T21:37:16.201+0900 I  CONTROL  [main] Au
tomatically disabling TLS 1.0, to force-enable TLS
 1.0 specify --sslDisabledProtocols 'none'
2020-01-28T21:37:16.207+0900 I  CONTROL  [initandl
isten] MongoDB starting : pid=7420 port=20001 dbpa
th=c:/data/secondary1 64-bit host=DESKTOP-K666V0K
2020-01-28T21:37:16.207+0900 I  CONTROL  [initandl
isten] targetMinOS: Windows 7/Windows Server 2008
R2
2020-01-28T21:37:16.207+0900 I  CONTROL  [initandl
isten] db version v4.2.0
2020-01-28T21:37:16.207+0900 I  CONTROL  [initandl
```

```
mongod --dbpath c:/data/secondary1 --port 20001 --replSet replica_test
```

■ ReplicaSet 시스템 구성

● Secondary2 서버 실행

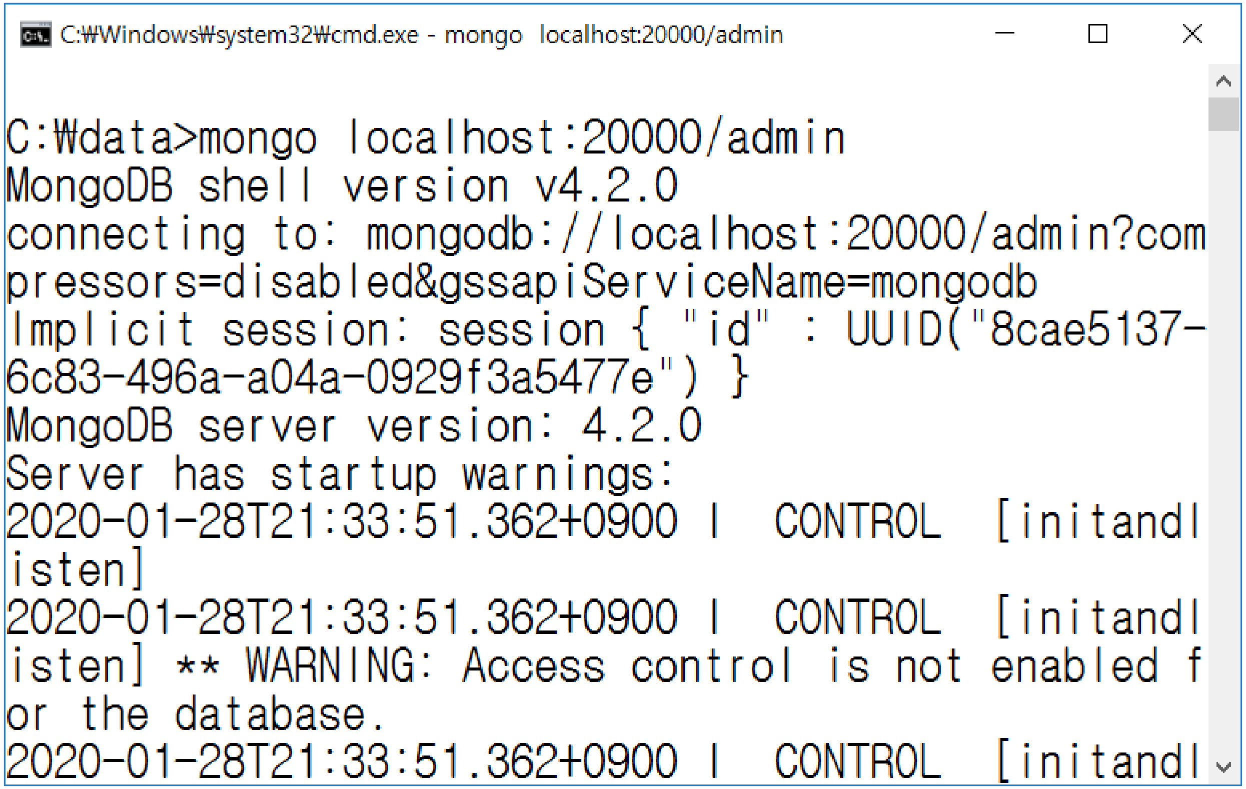

C:₩Windows₩system32₩cmd.exe - mongod --dbpath c:/data/secondary2 --port...

```
C:₩data>mongod --dbpath c:/data/secondary2 --port
20002 --replSet replica_test
2020-01-28T21:37:23.312+0900 I  CONTROL  [main] Au
tomatically disabling TLS 1.0, to force-enable TLS
 1.0 specify --sslDisabledProtocols 'none'
2020-01-28T21:37:23.732+0900 I  CONTROL  [initandl
isten] MongoDB starting : pid=21956 port=20002 dbp
ath=c:/data/secondary2 64-bit host=DESKTOP-K666VOK

2020-01-28T21:37:23.732+0900 I  CONTROL  [initandl
isten] targetMinOS: Windows 7/Windows Server 2008
R2
2020-01-28T21:37:23.733+0900 I  CONTROL  [initandl
isten] db version v4.2.0
```

```
mongod --dbpath c:/data/secondary2 --port 20002 --replSet replica_test
```

■ ReplicaSet 시스템 구성

● Primary 서버 접속

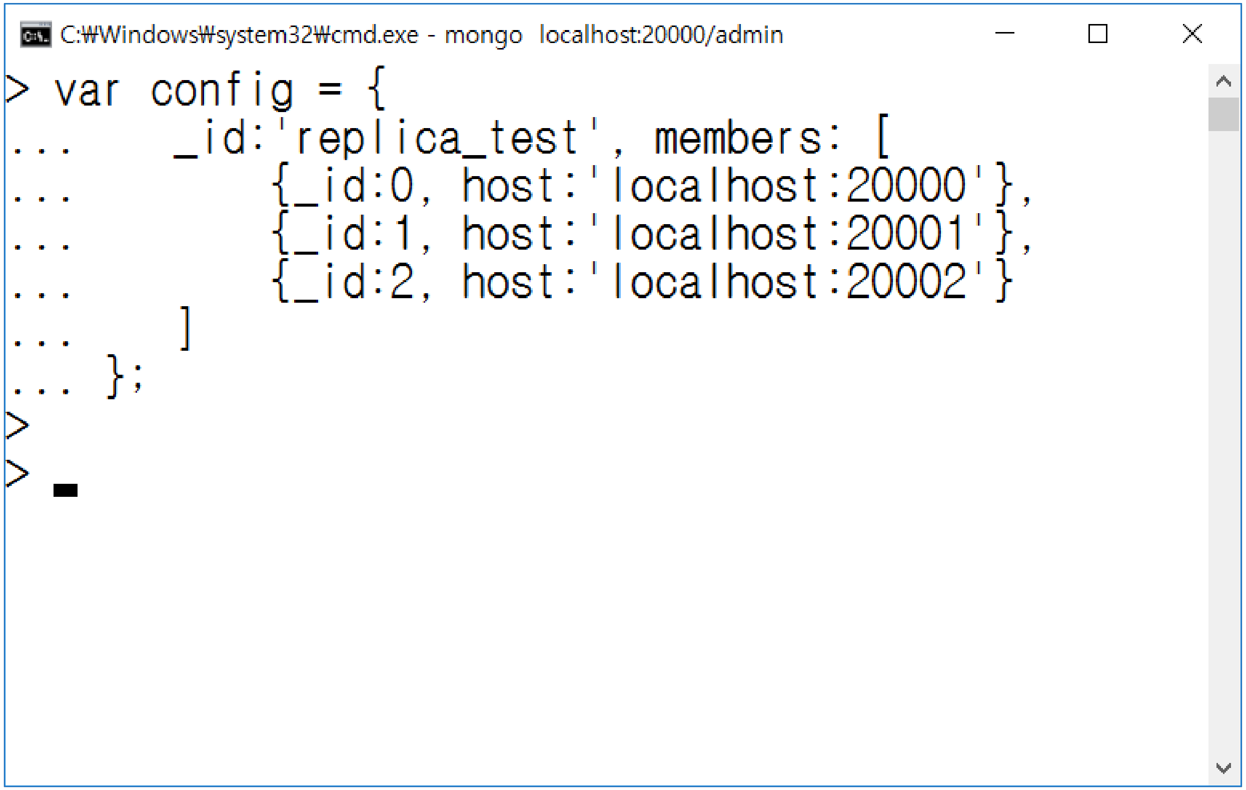


```
C:₩data>mongo localhost:20000/admin
MongoDB shell version v4.2.0
connecting to: mongodb://localhost:20000/admin?com
pressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("8cae5137-
6c83-496a-a04a-0929f3a5477e") }
MongoDB server version: 4.2.0
Server has startup warnings:
2020-01-28T21:33:51.362+0900 I  CONTROL  [initandl
isten]
2020-01-28T21:33:51.362+0900 I  CONTROL  [initandl
isten] ** WARNING: Access control is not enabled f
or the database.
2020-01-28T21:33:51.362+0900 I  CONTROL  [initandl
```

```
mongo localhost:20000/admin
```

■ ReplicaSet 시스템 구성

● Primary 서버 접속 → ReplicaSet 환경설정

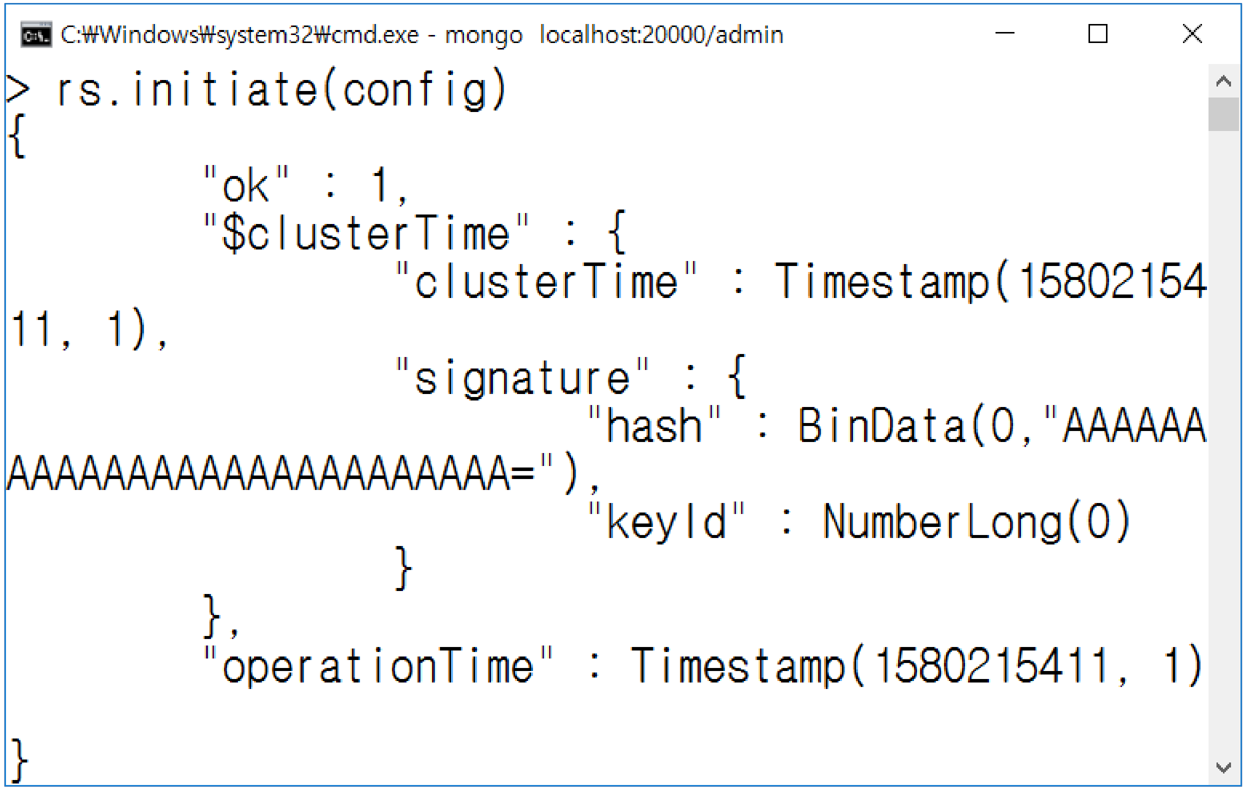


```
> var config = {
...     _id:'replica_test', members: [
...         {_id:0, host:'localhost:20000'},
...         {_id:1, host:'localhost:20001'},
...         {_id:2, host:'localhost:20002'}
...     ]
... };
>
> 
```

```
var config = {
   _id:'replica_test', members: [
       {_id:0, host:'localhost:20000'},
       {_id:1, host:'localhost:20001'},
       {_id:2, host:'localhost:20002'}
   ]
};
```

■ ReplicaSet 시스템 구성

● Primary 서버 접속 → ReplicaSet 환경설정 → 초기화 (환경설정 값으로 변경)

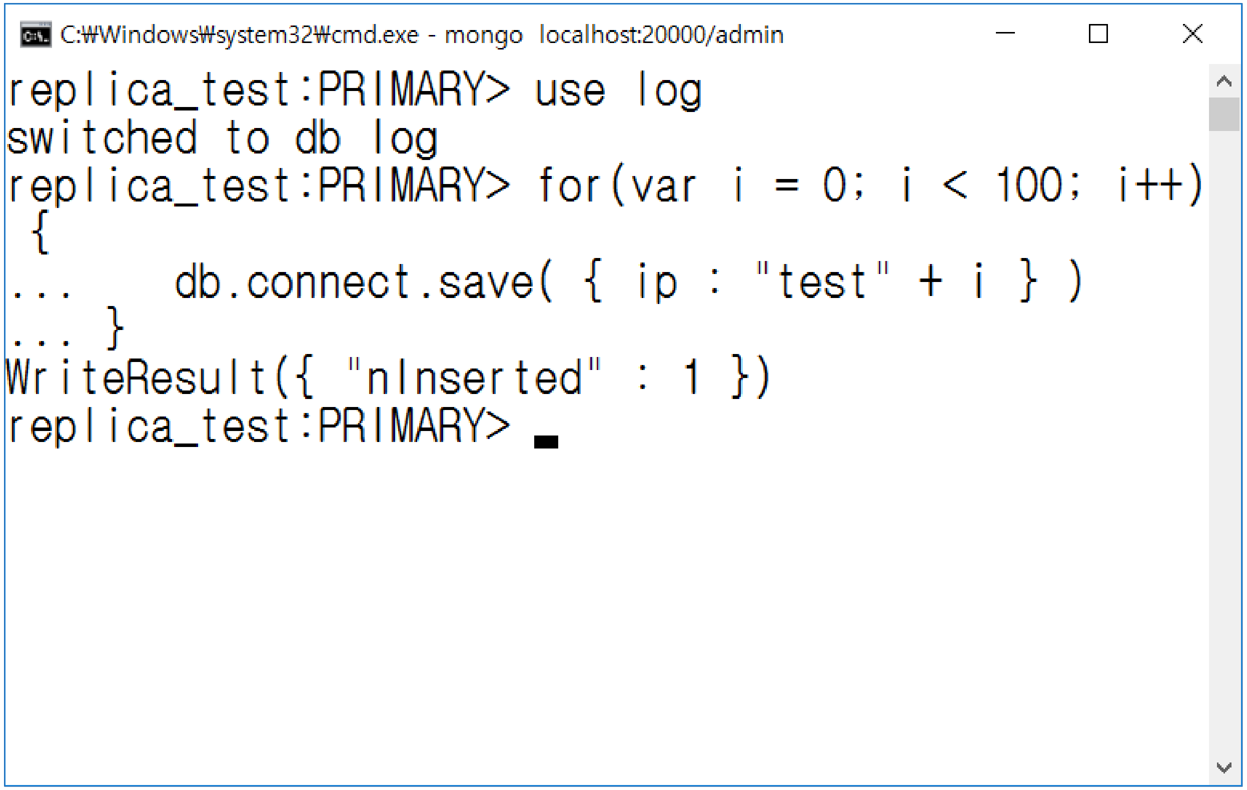


```
> rs.initiate(config)
{
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp(15802154
11, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAA
AAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        },
        "operationTime" : Timestamp(1580215411, 1)

}
```

```
rs.initiate(config)
```

■ ReplicaSet 시스템 구성

● Primary 서버에서 데이터 입력

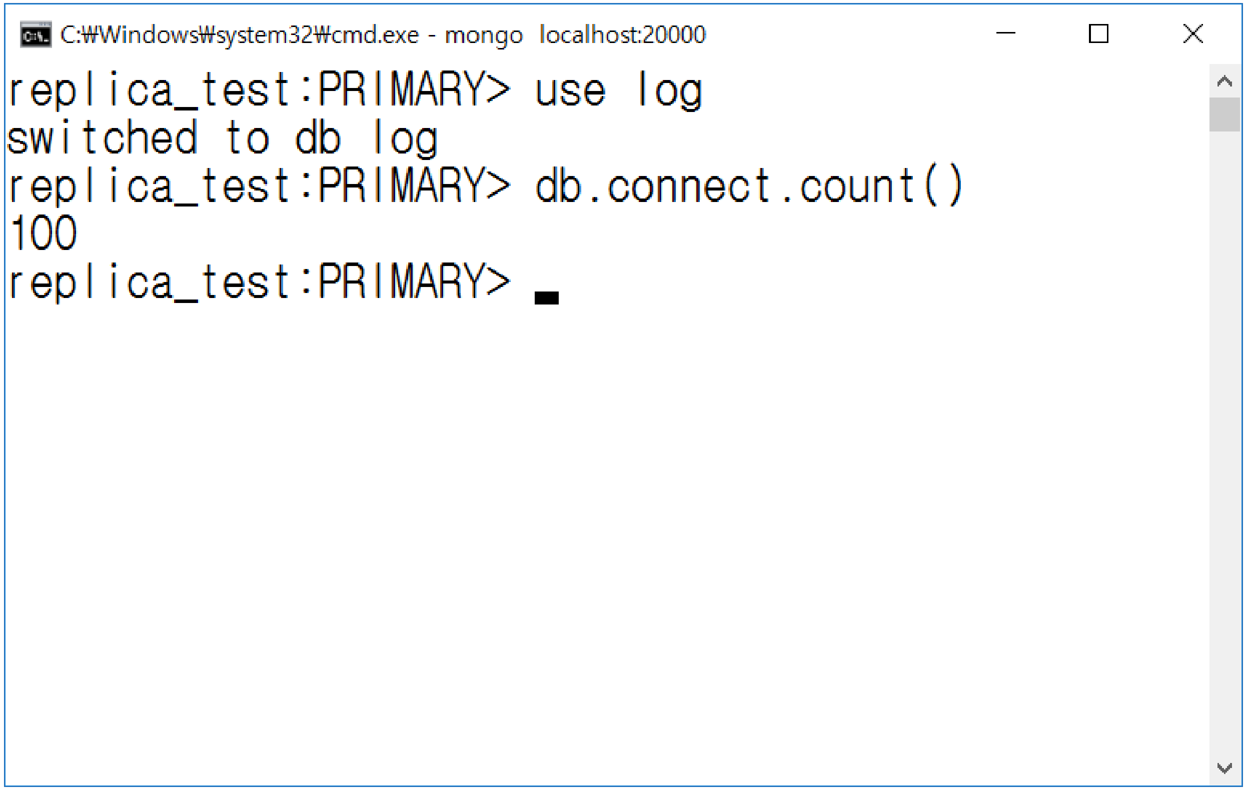


```
replica_test:PRIMARY> use log
switched to db log
replica_test:PRIMARY> for(var i = 0; i < 100; i++)
 {
...     db.connect.save( { ip : "test" + i } )
... }
WriteResult({ "nInserted" : 1 })
replica_test:PRIMARY>
```

```
use log

for(var i = 0; i < 100; i++) {
  db.connect.save( { ip : "test" + i } )
}
```

■ ReplicaSet 시스템 구성

● Primary 서버 데이터 확인

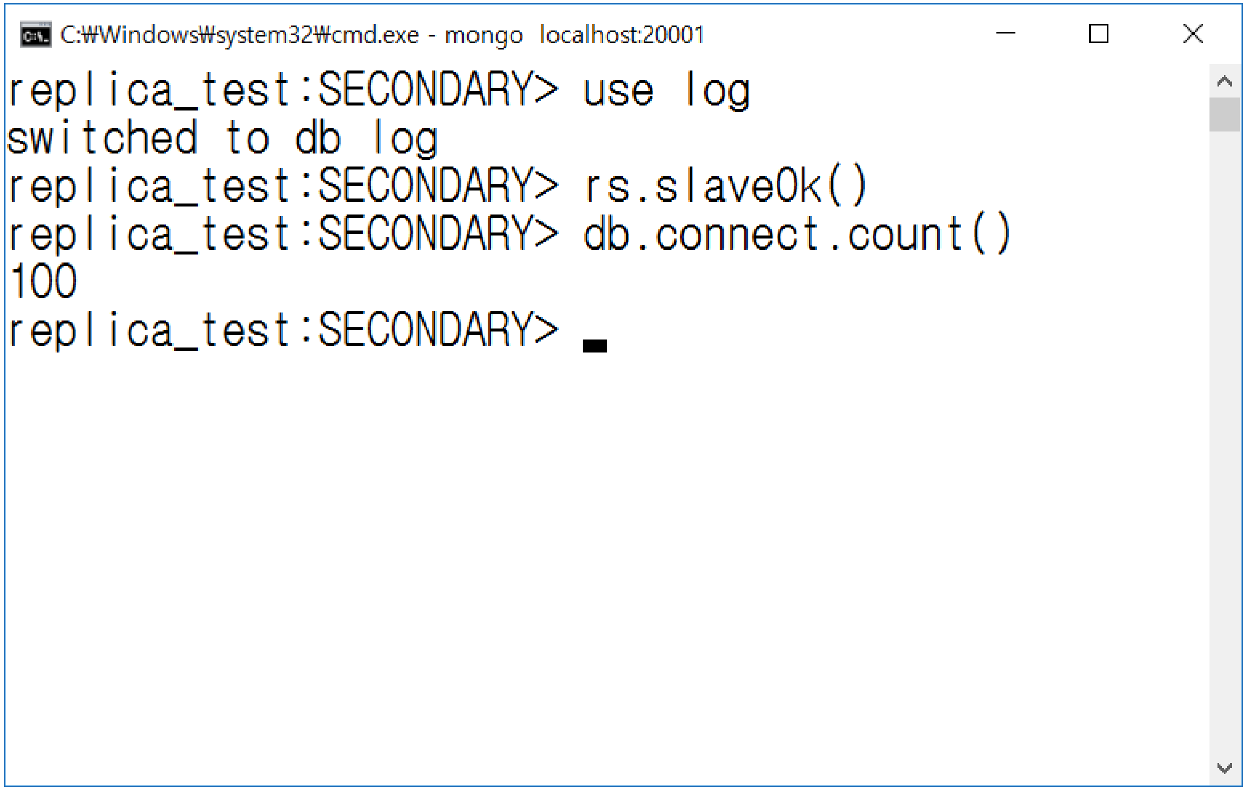


```
replica_test:PRIMARY> use log
switched to db log
replica_test:PRIMARY> db.connect.count()
100
replica_test:PRIMARY> _
```

```
use log

db.connect.count()
```

■ ReplicaSet 시스템 구성

● Secondary1 서버 접속 후 복제된 데이터 확인

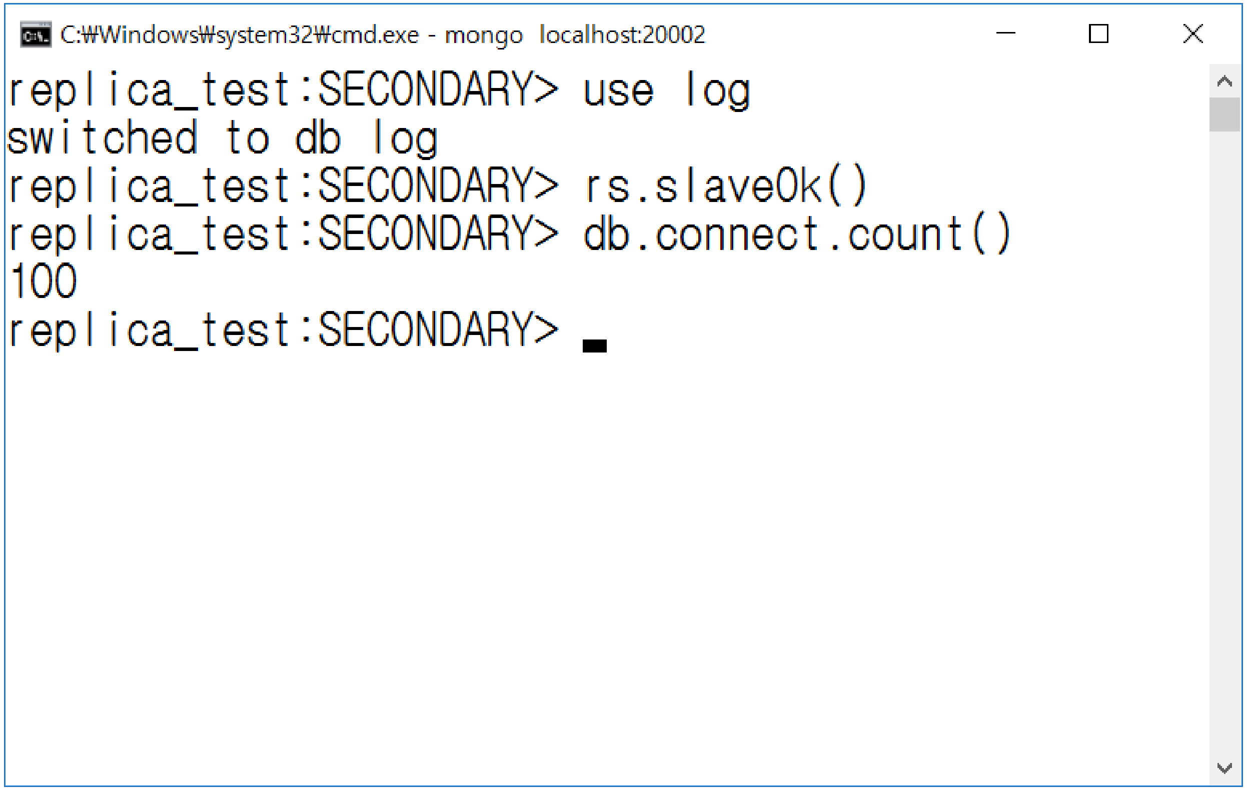


```
replica_test:SECONDARY> use log
switched to db log
replica_test:SECONDARY> rs.slaveOk()
replica_test:SECONDARY> db.connect.count()
100
replica_test:SECONDARY> _
```

```
use log

rs.slaveOk()

db.connect.count()
```

■ ReplicaSet 시스템 구성

● Secondary2 서버 접속 후 복제된 데이터 확인



```
replica_test:SECONDARY> use log
switched to db log
replica_test:SECONDARY> rs.slaveOk()
replica_test:SECONDARY> db.connect.count()
100
replica_test:SECONDARY> _
```

```
use log

rs.slaveOk()

db.connect.count()
```

■ ReplicaSet 시스템 구성

- ● Secondary 서버 추가 / 삭제 / 확인

1. 새로운 Secondary 서버 실행 (ex. mongod –port 20003 …)
2. Primary 서버 접속 후

   rs.add('localhost:20003')
3. ReplicaSet 상태 확인

   rs.conf()
4. rs.remove('localhost:20003')

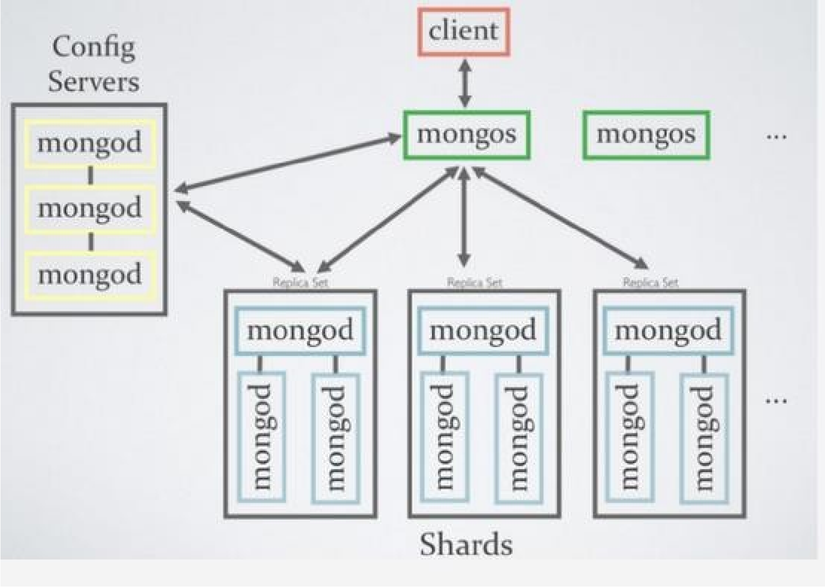
■ Sharding

● 대용량 데이터 저장을 위해 데이터를 분산 저장하는 기능

● 응용 ↔ 중계 ↔ 데이터 3계층 구조

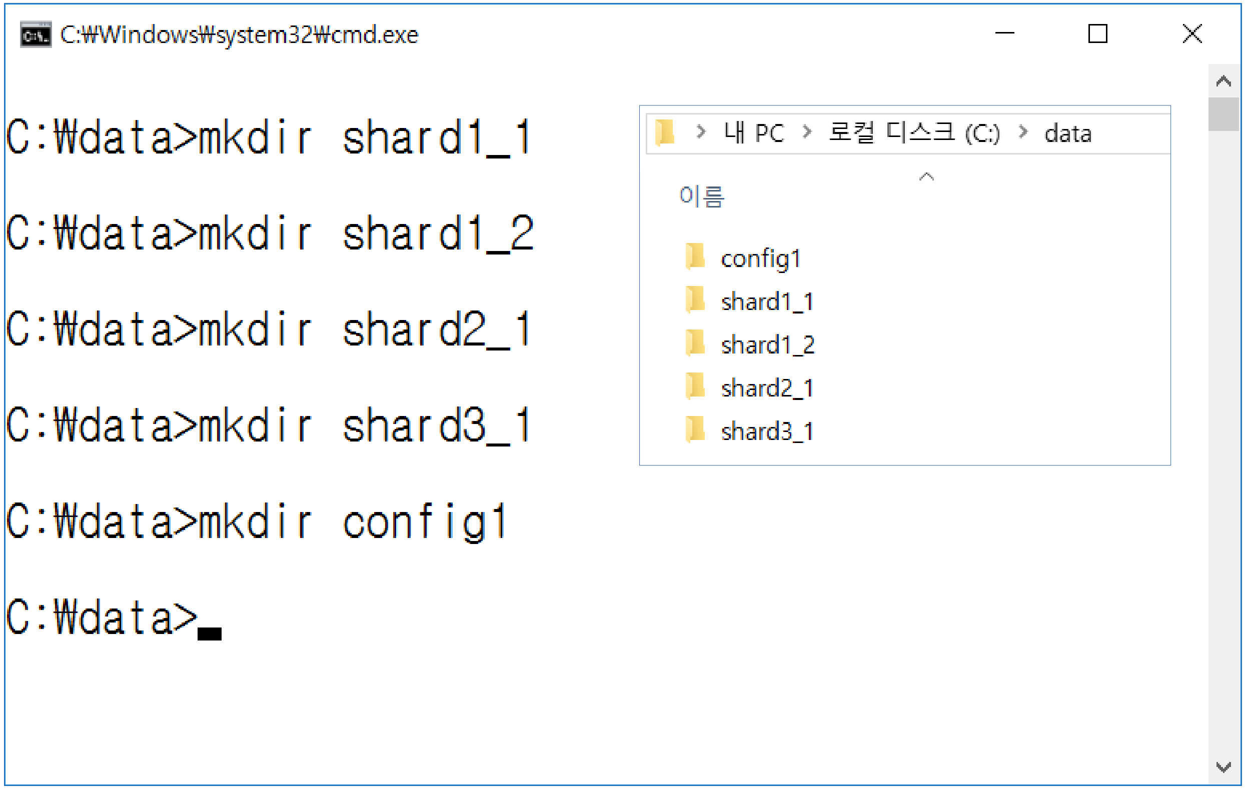
● 중계 계층에서 config 서버를 사용해야 하며 20~30% 정도 추가 메모리 필요

- config : Sharding을 위한 메타 데이터를 저장 (데이터의 위치 정보 저장)
- mongos : Client의 요청 처리, config 서버의 메타 데이터를 이용하여
  각 MongoDB의 데이터에 접근
- mongod : MongoDB의 데이터 서버 (상황에 따라 레플리카셋으로 구성)



■ Sharding 시스템 구성

● mongo 데이터 저장 서버 및 config 디렉토리 생성



C:₩Windows₩system32₩cmd.exe

```
C:₩data>mkdir shard1_1

C:₩data>mkdir shard1_2

C:₩data>mkdir shard2_1

C:₩data>mkdir shard3_1

C:₩data>mkdir config1

C:₩data>
```

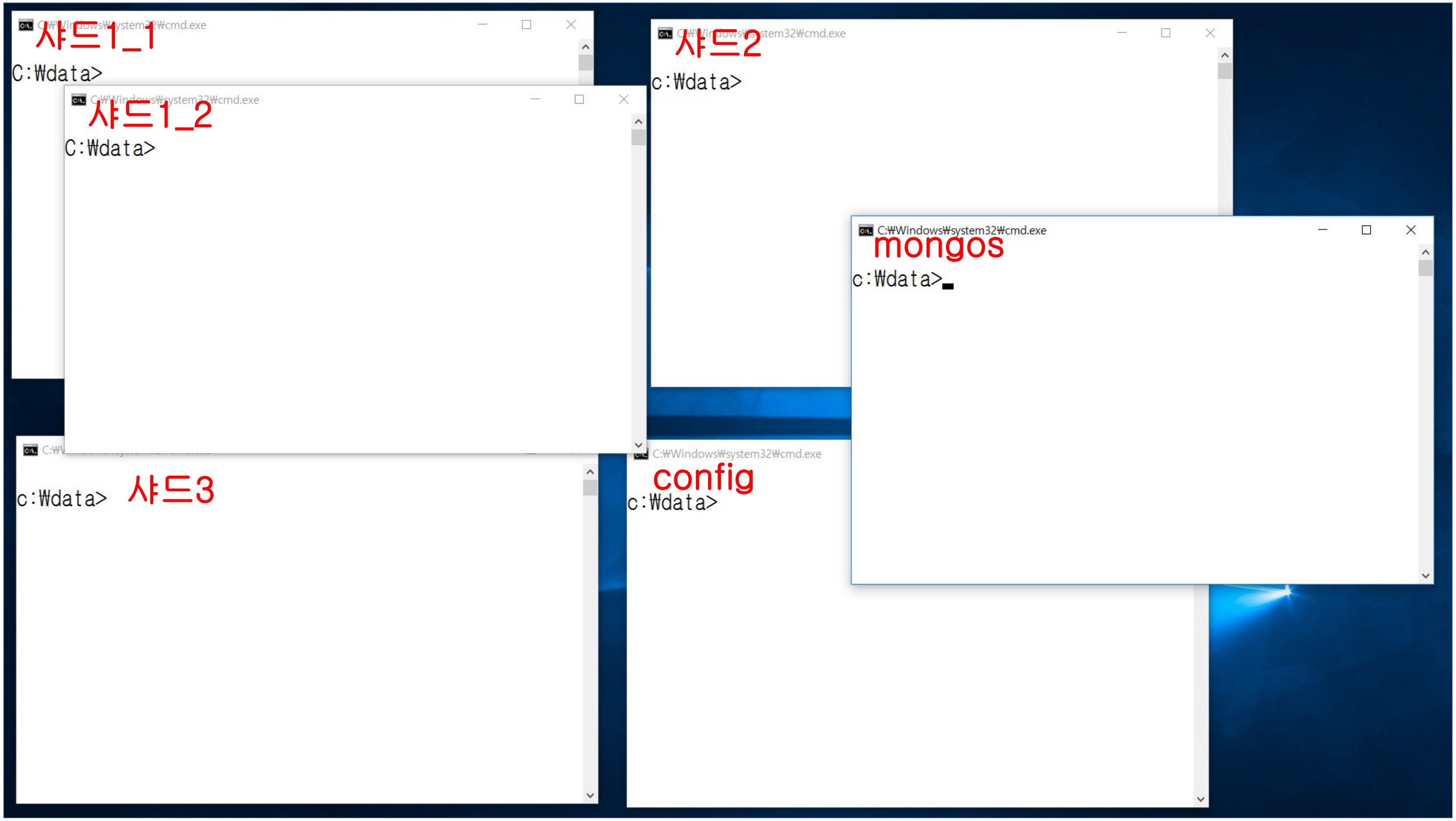
› 내 PC › 로컬 디스크 (C:) › data
이름

```
config1
shard1_1
shard1_2
shard2_1
shard3_1
```

- 데이터 서버 1 (레플리카셋 구성)
- 데이터 서버 2
- 데이터 서버 3
- 중계 서버 1

■ Sharding 시스템 구성

● 명령 프롬프트 6개 (샤드1_1/샤드1_2/샤드2/샤드3/config/mongos)



```
C:₩data>
```

```
C:₩data>
```

```
c:₩data>
```

```
c:₩data>_
```

```
c:₩data>
```

```
c:₩data>
```

■ Sharding 시스템 구성

● 데이터 서버 1 구동

```
mongod --shardsvr --dbpath c:₩data₩shard1_1 --port 40001 --replSet firstset

mongod --shardsvr --dbpath c:₩data₩shard1_2 --port 40002 --replSet firstset
```

● 레플리카셋 환경설정 (mongo localhost:40001/admin)

```
var config = {
  'replSetInitiate' : {
    _id : 'firstset',
    'members' : [
      { _id : 1, host : 'localhost:40001' },
      { _id : 2, host : 'localhost:40002' }
    ]
  }
}

db.runCommand( config )
```

■ Sharding 시스템 구성

● 데이터 서버 2 구동

```
mongod --shardsvr --dbpath c:₩data₩shard2_1 --port 40004 --replSet secondset
```

● 레플리카셋 환경설정 (mongo localhost:40004/admin)

```
var config = {
  'replSetInitiate' : {
    _id : 'secondset',
    'members' : [
      { _id : 1, host : 'localhost:40004' }
    ]
  }
}

db.runCommand( config )
```

- 구성된 서버가 1대만 있더라도 레플리카셋 환경설정 필요

■ Sharding 시스템 구성

● 데이터 서버 3 구동

```
mongod --shardsvr --dbpath c:₩data₩shard3_1 --port 40007 --replSet thirdset
```

● 레플리카셋 환경설정 (mongo localhost:40007/admin)

```
var config = {
  'replSetInitiate' : {
    _id : 'thirdset',
    'members' : [
      { _id : 1, host : 'localhost:40007' }
    ]
  }
}

db.runCommand( config )
```

- 구성된 서버가 1대만 있더라도 레플리카셋 환경설정 필요

■ Sharding 시스템 구성

● config 서버 구동

```
mongod --configsvr --replSet replica --dbpath c:₩data₩config1 --port 50001
```

● 레플리카셋 환경설정 (mongo localhost:50001/admin)

```
var config = {
   _id:'replica', members: [
      {_id:0, host:'localhost:50001'}
   ]
};

rs.initiate(config)
```

■ Sharding 시스템 구성

- ● mongos 서버 구동 (config 서버 등록)

```
mongos --configdb replica/localhost:50001 --port 50000
```

- ● Shard 서버 (데이터 서버) 등록 (mongo localhost:50000/admin)

```
db.runCommand( { addshard : 'firstset/localhost:40001,localhost:40002' } )
db.runCommand( { addshard : 'secondset/localhost:40004' } )
db.runCommand( { addshard : 'thirdset/localhost:40007' } )
```

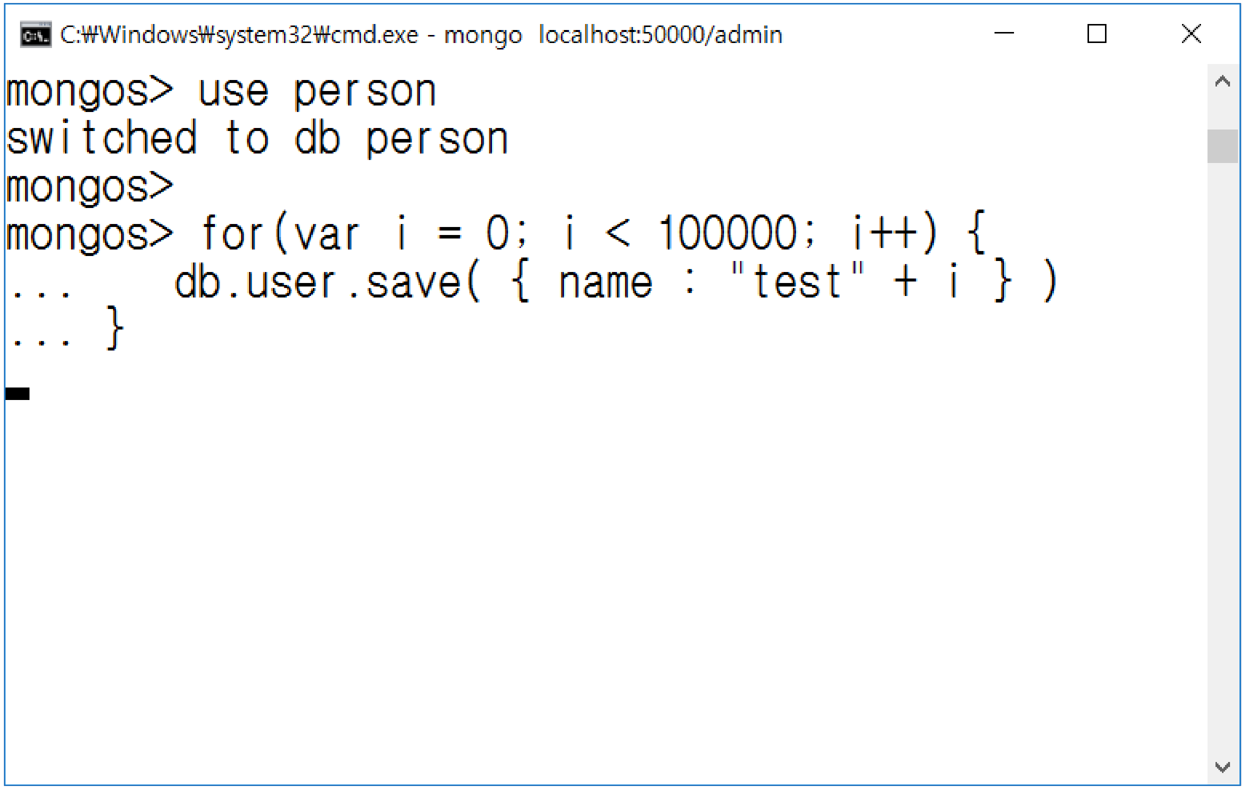
- ● Shard 활성화 및 Key 등록

```
db.runCommand({enablesharding: 'person'})
sh.shardCollection('person.user', {_id:'hashed'})
```

■ Sharding 시스템 구성

● mongos 서버에서 데이터 입력

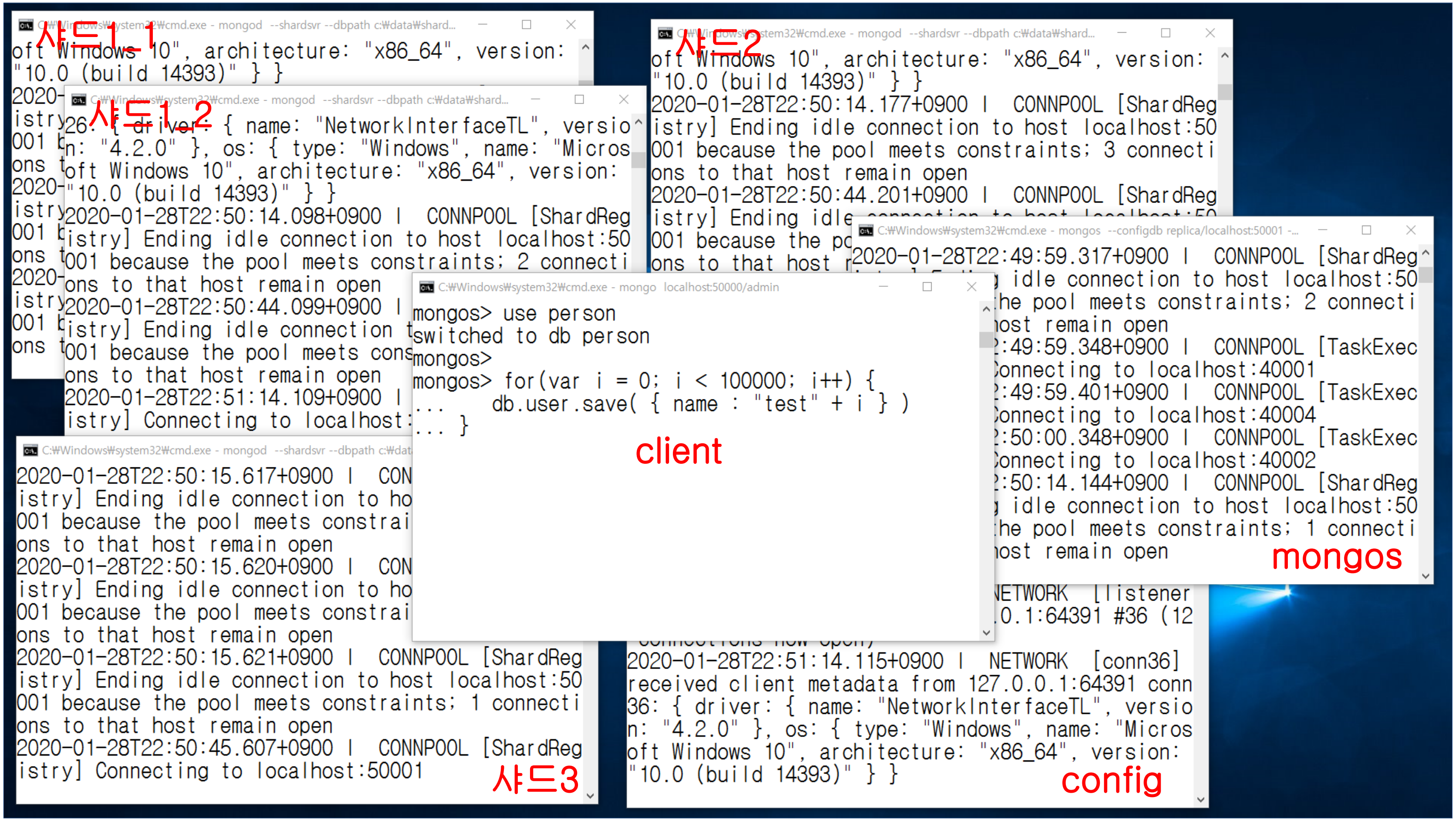
```
use person

for(var i = 0; i < 100000; i++) {
   db.user.save( { name : "test" + i } )
}
```



```
mongos> use person
switched to db person
mongos>
mongos> for(var i = 0; i < 100000; i++) {
...     db.user.save( { name : "test" + i } )
... }
```

■ Sharding 시스템 구성

● mongos 서버에서 데이터 입력


샤드1-1
C:₩Windows₩system32₩cmd.exe - mongod --shardsvr --dbpath c:₩data₩shard...
샤드1-2
C:₩Windows₩system32₩cmd.exe - mongod --shardsvr --dbpath c:₩data₩shard...
샤드2
C:₩Windows₩system32₩cmd.exe - mongod --shardsvr --dbpath c:₩data₩shard...
C:₩Windows₩system32₩cmd.exe - mongos --configdb replica/localhost:50001 -...
C:₩Windows₩system32₩cmd.exe - mongo localhost:50000/admin
client
mongos
C:₩Windows₩system32₩cmd.exe - mongod --shardsvr --dbpath c:₩dat
샤드3
config

```
oft Windows 10", architecture: "x86_64", version:
"10.0 (build 14393)" } }
2020-
istry
001 b
ons t
2020-
istry
001 b
ons t
2020-
istry
001 b
ons t
```

```
26: { driver: { name: "NetworkInterfaceTL", versio
n: "4.2.0" }, os: { type: "Windows", name: "Micros
oft Windows 10", architecture: "x86_64", version:
"10.0 (build 14393)" } }
2020-01-28T22:50:14.098+0900 I  CONNPOOL [ShardReg
istry] Ending idle connection to host localhost:50
001 because the pool meets constraints; 2 connecti
ons to that host remain open
2020-01-28T22:50:44.099+0900 I
istry] Ending idle connection t
001 because the pool meets cons
ons to that host remain open
2020-01-28T22:51:14.109+0900 I
istry] Connecting to localhost:
```

```
oft Windows 10", architecture: "x86_64", version:
"10.0 (build 14393)" } }
2020-01-28T22:50:14.177+0900 I  CONNPOOL [ShardReg
istry] Ending idle connection to host localhost:50
001 because the pool meets constraints; 3 connecti
ons to that host remain open
2020-01-28T22:50:44.201+0900 I  CONNPOOL [ShardReg
istry] Ending idle
001 because the po
ons to that host r
```

```
2020-01-28T22:49:59.317+0900 I  CONNPOOL [ShardReg
 idle connection to host localhost:50
he pool meets constraints; 2 connecti
ost remain open
:49:59.348+0900 I  CONNPOOL [TaskExec
onnecting to localhost:40001
:49:59.401+0900 I  CONNPOOL [TaskExec
onnecting to localhost:40004
:50:00.348+0900 I  CONNPOOL [TaskExec
onnecting to localhost:40002
:50:14.144+0900 I  CONNPOOL [ShardReg
 idle connection to host localhost:50
he pool meets constraints; 1 connecti
ost remain open
```

```
mongos> use person
switched to db person
mongos>
mongos> for(var i = 0; i < 100000; i++) {
...     db.user.save( { name : "test" + i } )
... }
```

```
2020-01-28T22:50:15.617+0900 I  CON
istry] Ending idle connection to ho
001 because the pool meets constrai
ons to that host remain open
2020-01-28T22:50:15.620+0900 I  CON
istry] Ending idle connection to ho
001 because the pool meets constrai
ons to that host remain open
2020-01-28T22:50:15.621+0900 I  CONNPOOL [ShardReg
istry] Ending idle connection to host localhost:50
001 because the pool meets constraints; 1 connecti
ons to that host remain open
2020-01-28T22:50:45.607+0900 I  CONNPOOL [ShardReg
istry] Connecting to localhost:50001
```

```
NETWORK  [listener
.0.1:64391 #36 (12
connections now open)
2020-01-28T22:51:14.115+0900 I  NETWORK  [conn36]
received client metadata from 127.0.0.1:64391 conn
36: { driver: { name: "NetworkInterfaceTL", versio
n: "4.2.0" }, os: { type: "Windows", name: "Micros
oft Windows 10", architecture: "x86_64", version:
"10.0 (build 14393)" } }
```

■ Sharding 시스템 구성

● 샤드 서버(데이터 서버 1) 접속 후 데이터 확인 (mongo localhost:40001)

```
use person

db.user.count()
```

● 샤드 서버(데이터 서버 2) 접속 후 데이터 확인 (mongo localhost:40004)

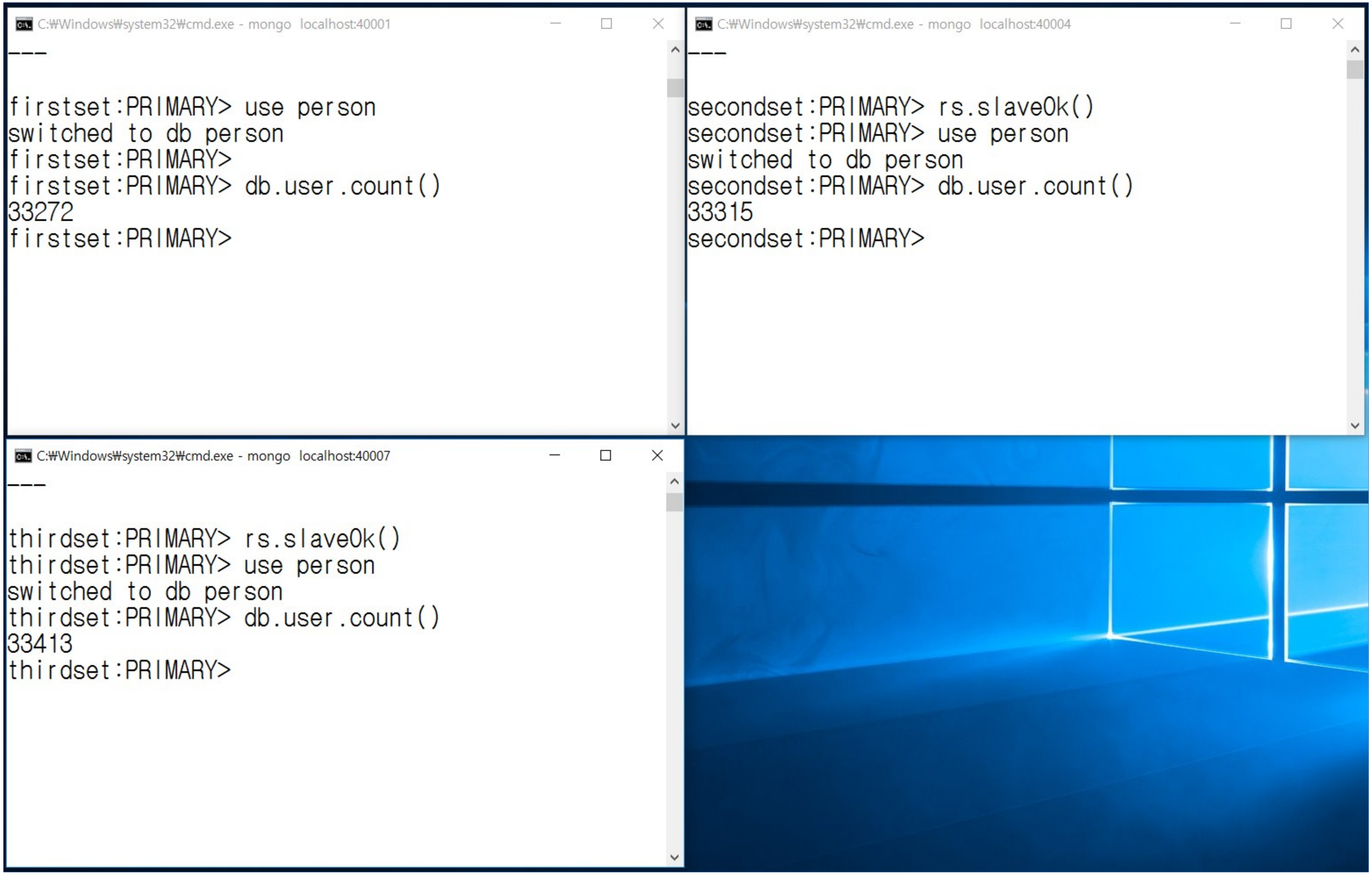
```
rs.slaveOk()
use person
db.user.count()
```

● 샤드 서버(데이터 서버 3) 접속 후 데이터 확인 (mongo localhost:40007)

```
rs.slaveOk()
use person
db.user.count()
```

■ Sharding 시스템 구성

● 각 서버 저장 데이터 확인



C:₩Windows₩system32₩cmd.exe - mongo localhost:40001

```
---

firstset:PRIMARY> use person
switched to db person
firstset:PRIMARY>
firstset:PRIMARY> db.user.count()
33272
firstset:PRIMARY>
```

C:₩Windows₩system32₩cmd.exe - mongo localhost:40004

```
---

secondset:PRIMARY> rs.slaveOk()
secondset:PRIMARY> use person
switched to db person
secondset:PRIMARY> db.user.count()
33315
secondset:PRIMARY>
```

C:₩Windows₩system32₩cmd.exe - mongo localhost:40007

```
---

thirdset:PRIMARY> rs.slaveOk()
thirdset:PRIMARY> use person
switched to db person
thirdset:PRIMARY> db.user.count()
33413
thirdset:PRIMARY>
```