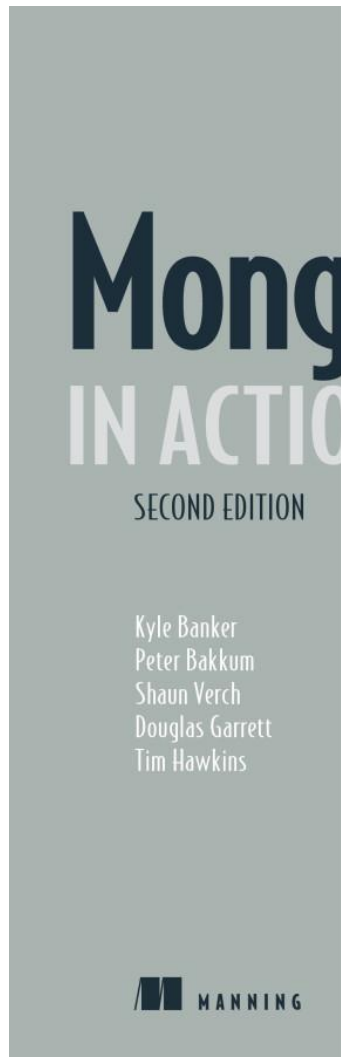
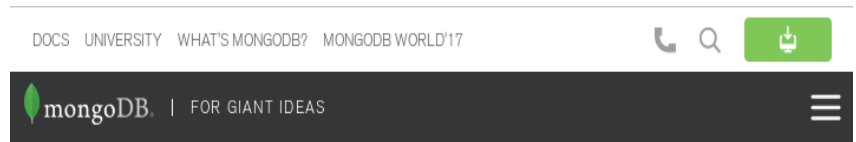


Mongo DB

Eine Dokumenten-orientierte Datenbank



Covers MongoDB version 3.0



Datenverwaltung neu erfunden

Werden Sie agiler und skalierbarer - mit
MongoDB, der führenden NoSQL-Datenbank

Jetzt herunterladen

[View English Version](#)

MongoDB ist eine der marktführenden NoSQL-Datenbanken. MongoDB (von "huMONGOus", sprich "extrem groß") wurde für die heutige moderne IT-Landschaft

- Die benutzten Datenbanken sind entweder Open Source oder werden in der Community-Edition benutzt
- Weitere in diesem Seminar verwendete Werkzeuge sind Open Source
 - LPGL Lizenzmodell
- Benutzt wird eine Vielzahl von Produkten und Technologien
 - Konsolen
 - Abfragesprachen
 - Programmiersprachen
- Programmierung
 - Damit werden die Inhalte durch Übungen vertieft und verinnerlicht
 - Eine Musterlösung wird in elektronischer Form angeboten
 - Diese muss jedoch weder die eleganteste noch beste Lösung sein!

© Integrata Cegos GmbH

Integrata Cegos GmbH
Zettachring 4
70567 Stuttgart

Alle Rechte, einschließlich derjenigen des auszugsweisen Abdrucks, der fotomechanischen und elektronischen Wiedergabe vorbehalten.

Die MongoDB

6

1

DIE MONGODB

1.1

BEGRIFFE

■ ...

1.2

ADMINISTRATION

- mongodump
 - alles
 - -d Datenbank
 - -c Collection
- mongorestore
- Slave Server
- Sperren

```
$mongo
>use admin
>db.runCommand({fsync:1,lock:1})
{
  "info" : "now locked against writes",
  "ok" : 1
}
```

- Format
 - csv
 - tsv
 - json
- mongoimport
- mongoexport

```
■ db.createUser(  
■ {  
■   user: "reportsUser",  
■   pwd: "12345678",  
■   roles: [  
■     { role: "read", db: "reporting" },  
■     { role: "read", db: "products" },  
■     { role: "read", db: "sales" },  
■     { role: "readWrite", db: "accounts" }  
■   ]  
■ }  
■ )
```

- # mongodb.conf
- dbpath=/var/lib/mongodb
- logpath=/var/log/mongodb/mongodb.log
- logappend=true
- auth = false
- #enable the rest interface
- rest =true

- in admin
 - db.version()
 - db.serverStatus()
 - db.top()
 - db.currentOp()

- `mongod -f /etc/mongodb.conf --repair`
- `db.repairDataBase()`
- `db.my_db.reIndex()`

- `mongod -f /etc/mongodb.conf --repair`
- `db.spreadsheets.reIndex()`

- mongostat
- http-interface
- mongosniff
- bsondump

- `mongod --slowms <ms>`
- Slow Query Warning
- `db.setProfilingLevel(0|1|2), db.setProfilingLevel(0|1|2, slowms)`
- `query.explain()`
 - `explain(true)` mit Query Optimizer
 - `hint({my_index = 1})`: mit Index, den der Optimizer benutzen soll
- Coverage index

- Journaling
 - Versuch, Datenkonsistenz bei Crash zu erreichen
 - Append-only-log
- Replikationsstrategien
 - Master-Slave
 - Replica Sets
 - bieten mehr Möglichkeiten

- Redundancy
- Failover
- Intensive Operationen laufen auf dem Secondary
 - Reindexing
 - Kompaktierung
- Read-Loadbalancing

Aufsetzen des Replica Sets

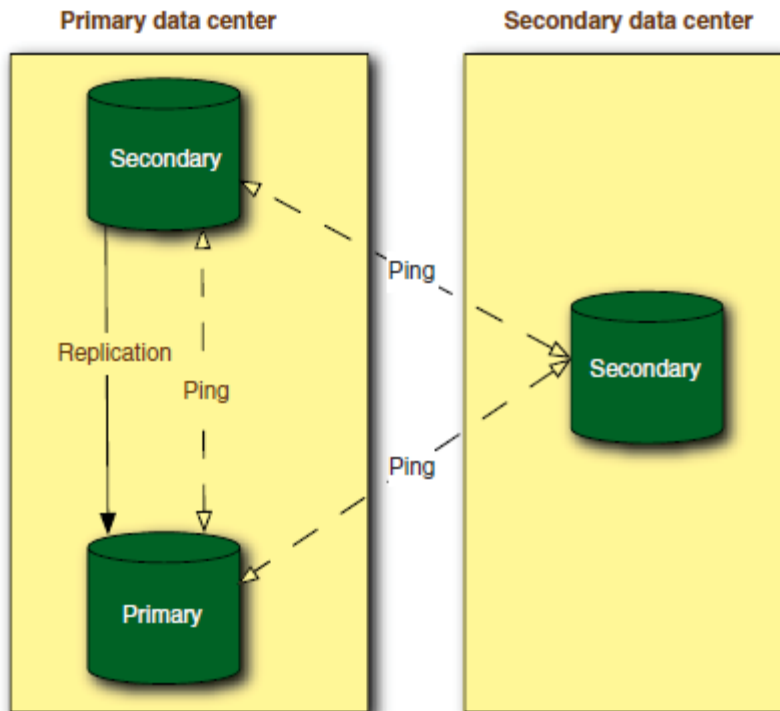


Figure 8.1 A basic replica set consisting of a primary, a secondary, and an arbiter

Befehle zum Aufsetzen des Replica Sets

```
mkdir /data/node1
mkdir /data/node2
mkdir /data/arbiter
mongod --replSet myapp --dbpath /data/node1 --port
40000
mongod --replSet myapp --dbpath /data/node2 --port
40001
mongod --replSet myapp --dbpath /data/arbiter --port
40002
```

- Vorher Fehler, deshalb:
- `rs.initiate()`
- `rs.add("localhost:40001")`
- `{ "ok" : 1 }`
- `> rs.add("localhost.local:40002", {arbiterOnly: true})`
- `{ "ok" : 1 }`
- `db.isMaster()`
- `rs.status()`

- Olog
 - Bestandteil der Nicht-replizierten local-Datenbank
 - Olog-Einträge enthalten alle notwendigen Informationen, um eine Datenänderung durchzuführen
 - Olog-Einträge werden repliziert
 - Olog ist eine Capped Collection
 - Damit kann ein Secondary aus der Replikation herausfallen, falls sein Einstiegspunkt nicht mehr vorhanden ist
 - `mongod --replSet myapp --oplogSize 1024`
- Majority: Ein Primary muss die Majorität der anderen Beteiligten des Replica Sets sehen
 - Sonst setzt er sich selbst als secondary

Sharding

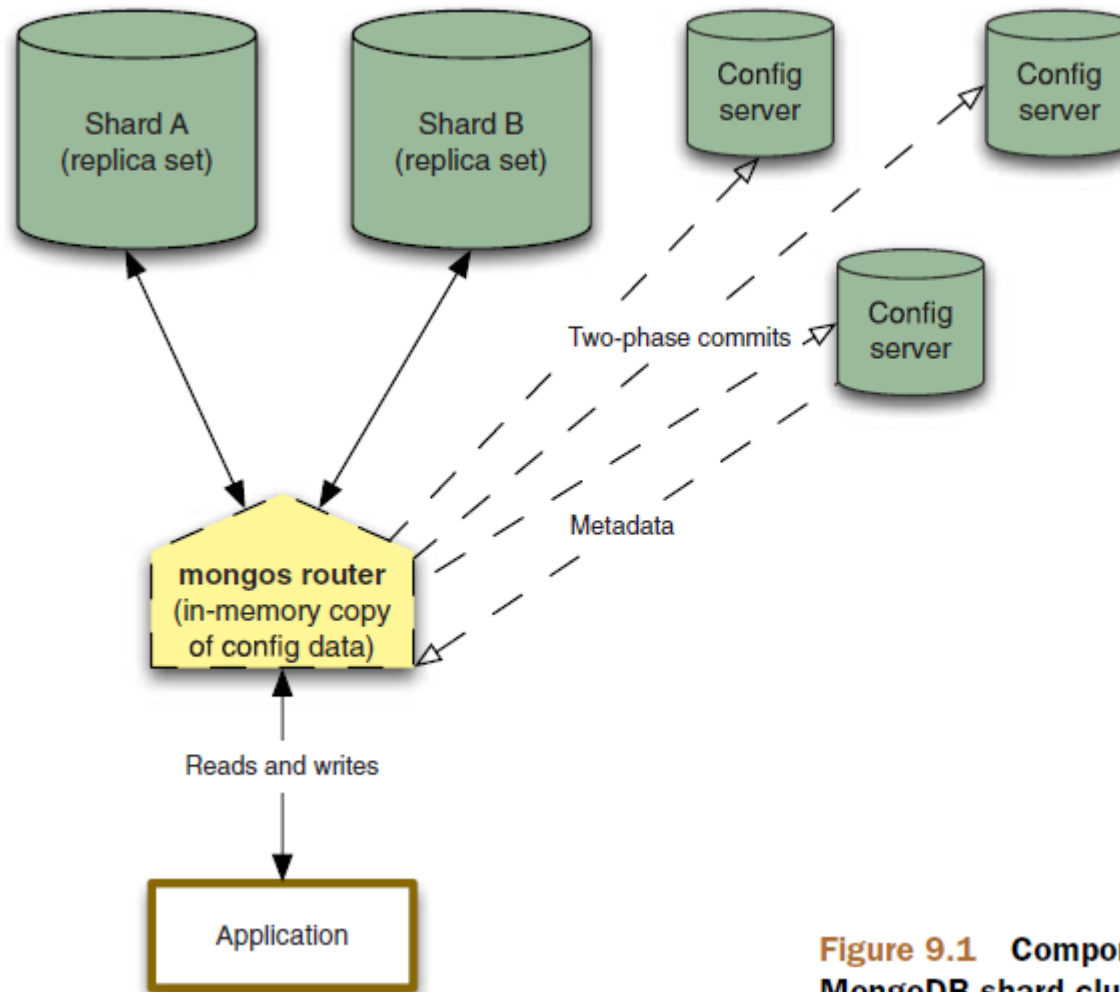


Figure 9.1 Components in a MongoDB shard cluster

Was wird gesharded?

- Jede Datenbank
- Collections, für die ein Sharding Key definiert ist

- Zwei Shards m, it jeweils 3 Replica Sets
- `$ mkdir /data/rs-a-1 (-3)`
- `$ mkdir /data/rs-b-1 (-3)`
- Starten
- `mongod --shardsvr --replSet shard-a --dbpath /data/rs-a-1 \`
- `--port 30000 --logpath /data/rs-a-1.log --fork --nojournal`
- ...
- `mongod --shardsvr --replSet shard-b --dbpath /data/rs-b-1 \`
- `--port 30100 --logpath /data/rs-b-1.log --fork --nojournal`
- ...
- `$ mongo localhost:30000`
- `> rs.initiate()`
- `> rs.add("localhost:30000")`
- `> rs.add("localhost:30001", {arbiterOnly: true})`

- Configserver
- `$ mkdir /data/config-1`
- `$ mongod --configsvr --dbpath /data/config-1 --port 27019 \`
- `--logpath /data/config-1.log --fork --nojournal`
- `$ mkdir /data/config-2`
- `$ mongod --configsvr --dbpath /data/config-2 --port 27020 \`
- `--logpath /data/config-2.log --fork --nojournal`
- `$ mkdir /data/config-3`
- `$ mongod --configsvr --dbpath /data/config-3 --port 27021 \`
- `--logpath /data/config-3.log --fork --nojournal`

- Mongos
- `$ mongos --configdb localhost:27019,localhost:27020,localhost:27021 \`
- `--logpath /data/mongos.log --fork --port 40000`

- Cluster
- `$ mongo localhost:40000`
- `> sh.addShard("shard-a/localhost:30000,localhost:30001")`
- `{ "shardAdded" : "shard-a", "ok" : 1 }`
- `> sh.addShard("shard-b/localhost:30100,localhost:30101")`
- `{ "shardAdded" : "shard-b", "ok" : 1 }`

- Kontrolle
- `db.getSiblingDB("config").shards.find()`
- `{ "_id" : "shard-a", "host" : "shard-a/localhost:30000,localhost:30001" }`
- `{ "_id" : "shard-b", "host" : "shard-b/localhost:30100,localhost:30101" }`
- oder
- `> use admin`
- `> db.runCommand({listshards: 1})`

- Datenbank
- `> sh.enableSharding("cloud-docs")`

- Check
- `> db.getSiblingDB("config").databases.find()`
- `{ "_id" : "admin", "partitioned" : false, "primary" : "config" }`
- `{ "_id" : "cloud-docs", "partitioned" : true, "primary" : "shard-a" }`

- Collection
- `> sh.shardCollection("cloud-docs.spreadsheets", {username: 1, _id: 1})`

- Check
- `> db.getSiblingDB("config").collections.findOne()`
- `{`
- `"_id" : "cloud-docs.spreadsheets",`
- `"lastmod" : ISODate("1970-01-16T00:50:07.268Z"),`
- `"dropped" : false,`
- `"key" : {`
- `"username" : 1,`
- `"_id" : 1`
- `},`
- `"unique" : false`
- `}`