
IIJbootCamp

MongoDBを触ってみよう
- docker版

IIJ s-fukuda
2019.04.17

MongoDB を触ってみよう

受講者をお願い

- ・ bootcampカリキュラムで利用できるNotePCがある方はCentOS7 VMを一つ使えることが望ましいので事前準備をお願いします
- ・ CentOS[67] へdocker-compose コマンドのインストールをお願いします
- ・ Mongodb-3.6系を利用します
- ・ 以上、必須ではありませんが準備に最善を尽くしてください

MongoDB を触ってみよう

Agenda

- ・ docker-compose による構築
- ・ 起動、ログイン確認
- ・ Database / Collection / Index … 各種作成
- ・ Insert / Upsert / Remove … 各種操作
- ・ Replica Set

MongoDB SetUP -- docker-compose

インストール

BootCamp資料

<https://gh.iiij.jp/gist/s-fukuda/176778f44af9e35ca19a704896d9bd5a>

公式ドキュメント

Package Install : Redhat系

<https://docs.mongodb.com/manual/tutorial/install-mongodb-on-red-hat/>

<https://docs.mongodb.com/manual/tutorial/deploy-replica-set/>

MongoDB 起動

ログイン

- ・ ReplicaSet構成を考慮したオペレーション

```
# docker exec -it mongors01 mongo
```

```
MongoDB shell version v3.6.13
```

```
connecting to: mongodb://127.0.0.1:27017/?gssapiServiceName=mongodb
```

```
Implicit session: session { "id" : UUID("955fcedc-f627-41f1-b121-fb50aa7e4651") }
```

```
MongoDB server version: 3.6.13
```

```
>
```

MongoDB オペレーション

Database 新規作成

- ・ mongors01 にて実行する

```
> rs.initiate()
```

```
rs1:OTHER> use classOf2019
```

MongoDB オペレーション

データ操作

- Collection 新規作成

```
rs1:PRIMARY > db.createCollection("people")
```

```
rs1:PRIMARY > db.people.insert({name: "your name", age: 22})
```

- Query実行

```
rs1:PRIMARY > db.people.find()
```

MongoDB オペレーション

データ操作

- `upsert`

```
rs1:PRIMARY > db.people.update({name: "kenji"},{$set: {sal: 100000}},  
{upsert:true})
```

```
rs1:PRIMARY > db.people.find() ... レコードの変化を確認
```

- `remove`

```
rs1:PRIMARY > db.people.remove( {"sal": {$lt: 10000}} )
```

```
rs1:PRIMARY > db.people.find()
```

MongoDB Replica Set

Replica Setを作成

- ・ mongors01 にて実行する

```
rs1:PRIMARY > rs.status() ... Status確認
```

```
rs1:PRIMARY> rs.add("mongors02:27017")
```

```
rs1:PRIMARY > rs.status()
```

```
rs1:PRIMARY> rs.addArb("mongoa01:27017")
```

MongoDB Replica Set

SECONDARYで実行

```
# docker exec -it mongors02 mongo
```

```
rs1:SECONDARY> use classOf2019
```

```
rs1:SECONDARY> db.people.find()
```

```
...
```

```
rs1:SECONDARY> rs.slaveOk()
```

```
rs1:SECONDARY> db.getMongo().setSlaveOk()
```

MongoDB 課題

どこまで出来るか？

1. replica set を FailOver してみてください
2. SECONDARYで23歳以上のカウントをするコマンドを教えてください
3. バックアップからリストアする方法を検討してみてください
4. 3.の方法について、どうしてそうしたか理由を教えてください
5. MongoDB を稼働させる Linux kernel の”THP” 実装利用で注意する点があれば指摘してください
6. MongoDBの機能を活かし、神保町カレーマップ位置情報を検索するクエリーを実装し社内githubで公開してください