#-----# MongoDB **のインストール** #-----

1.ダウンロード

以下のホームページから MongoDB をダウンロード http://www.mongodb.org/downloads

(Mac,Linux) mongodb-linux-x86_64-2.6.2.tgz をダウンロード (64bitOS を前提)

wget http://fastdl.mongodb.org/linux/mongodb-linux-x86_64-2.6.2.tgz でも OK

(Windows)

mongodb-win32-x86 64-2008plus-2.6.2.zip をダウンロード(64bit OS を前提)

2.解凍

→回答した中に GNU-AGPL-3.0 README THIRD-PARTY-NOTICES bin のディレクトリが ある事を確認

#-----# MongoDB **の起動** #------

3.MongoDB 解凍ディレクトリに移動

(Mac,Linux) cd mongodb-linux-x86 64-2.6.2

(Windows) cd mongodb-win32-x86 64-2008plus-2.6.2

4.データディレクトリの作成

mkdir data

5.起動

(Mac,Linux)
./bin/mongod --dbpath data --nojournal

(Windows) bin\mongod.exe --dbpath data --nojournal

→ 「waiting for connections on port 27017」と出力されることを確認

6.別のターミナルを立ち上げる

7.MongoDB 解凍ディレクトリに移動

8.MongoDB **クライアントで接続**

(Mac,Linux) ./bin/mongo

(Windows) bin\mongo.exe

> →プロンプト受付になる事を確認

#------# **基本的な** CRUD #------

簡単なデータで試す

9.DB 作成

use mydb

10.挿入

```
db.mycol.insert({"key1":"value1"})
db.mycol.insert({"key2":"value2"})
```

11.検索

```
db.mycol.find()
db.mycol.find({"key1":"value1"})
```

12.更新

db.mycol.update({"key1":"value1"}, {"key1":"value-hoge"})

13.削除

```
db.mycol.remove({"key1":"value1"})
db.mycol.remove({})
```

少し実践的なデータで試してみる

```
# 14.変数を定義
doc = {
    "name" : "watanabe",
    "skill" : ["MongoDB","KVM","ruby"],
    "editor": "emacs"
}
```

```
# 15.変数を確認
doc
```

```
# 16.変数を挿入
db.profile.insert( doc )
```

17.もう二つほどドキュメントを入れる

```
doc2 = \{
  "name" : "ogasawara",
  "skill" : ["MongoDB", "LibreOffice", "Printing"],
  "editor": "vim"
}
db.profile.insert( doc2 )
doc3 = \{
 "name"
         : "kubota",
  "skill" : ["MongoDB", "MySQL", "PostgreSQL", "ruby", "c++", "java", "Web"],
  "editor" : "emacs",
  "keybord": "kinesis"
}
db.profile.insert( doc3)
# 18.skill が MongoDB である人を検索する
db.profile.find({"skill":"MongoDB"})
# 19.skill が MongoDB で検索し、name 属性だけを射影する
db.profile.find({"skill":"MongoDB"}, {"name":1})
# 20.skill が KVM で検索し、name 属性だけを射影する
db.profile.find({"skill":"KVM"}, {"name":1})
# 21.name が watanabe なドキュメントを一つ検索して、変数 watanabe に格納
watanabe = db.profile.findOne({"name":"watanabe"})
# 22.watanabeのjobにアクセス
watanabe["skill"]
watanabe["skill"][1]
# 23.MongoDB コンソールを抜ける
exit
#-----
# 集計してみる
#_____
# 24.サンプルデータを MongoDB 解凍ディレクトリにダウンロード
https://github.com/fetaro/myrepo/raw/master/json-sample/scores.js
# 25.サンプルデータのインポート
(Mac,Linux)
./bin/mongoimport -d test -c scores < scores.js
(Windows)
bin\mongoimport -d test -c scores < scores.js</pre>
→「2014-05-08T07:10:18.257+0000 imported 800 objects」を確認
```

```
# 26.MongoDB クライアントで接続
```

(Mac,Linux) ./bin/mongo

(Windows) bin\mongo.exe

27.中身を見る

)

db.scores.find()

28. 続きを見たい時は it it

```
# 29.type が homework なドキュメントに対して、class_id と student_id で、平均を集計
してみる
db.scores.aggregate(
 {$match: {type: "homework"}},
 {$group: {
 __id: { class_id : "$class_id", student_id : "$student_id"},
 avg: {$avg: "$score"}
 }
 }
```

30.出力結果をコレクションに保存する

```
db.scores.aggregate(
 {$match: {type: "homework"}},
 {$group: {
    _id: { class_id : "$class_id", student_id : "$student id"},
    avg: {$avg: "$score"}
    }
 },
 {$out: "avgs"}
)
# 31.MongoDB コンソールを抜ける
exit
#-----
# MongoDB の停止
#-----
# 32.MongoDB を起動しているターミナルで停止
Ctrl + C
#-----
# レプリケーション
#-----
```

cd (MongoDB 解凍ディレクトリ)

33.データディレクトリの作成

mkdir nodel mkdir node2 mkdir node3

34.インスタンスの起動

```
(Mac,Linux)
./bin/mongod --dbpath=node1 --replSet=myrep --port=30000 --fork --logpath
mongodb30000.log --nojournal &
./bin/mongod --dbpath=node2 --replSet=myrep --port=30001 --fork --logpath
mongodb30001.log --nojournal &
./bin/mongod --dbpath=node3 --replSet=myrep --port=30002 --fork --logpath
mongodb30002.log --nojournal &
(Windows)
start bin\mongod --dbpath=node1 --replSet=myrep --port=30000 --nojournal
start bin\mongod --dbpath=node2 --replSet=myrep --port=30001 --nojournal
start bin\mongod --dbpath=node3 --replSet=myrep --port=30002 --nojournal
# 35.接続
(Mac,Linux)
./bin/mongo --port=30000
(Windows)
bin\mongo --port=30000
# 36.設定ドキュメントの定義
config = \{
  _id : "myrep",
 members : [
    { id : 0, host : "localhost:30000" },
    { _id : 1, host : "localhost:30001" },
    { id : 2, host : "localhost:30002" } ] }
# 37. レプリケーション開始
rs.initiate(config)
→ "ok" : 1 なら成功
# 38.状態確認
rs.status()
### 実際にレプリケーションが行われるか試してみる
# 39.データの挿入
```

use mydb

for(var i=0; i<10000; i++) db.logs.insert({ "uid":i,
"value":Math.floor(Math.random()*10000+1) })
db.logs.count()</pre>

他のノードにレプリケーションされているか見る

41.Mongo コンソールを終了 exit

42.30001 番ポートのノードに接続 (Mac,Linux) ./bin/mongo --port=30001

(Windows) bin\mongo --port=30001

use mydb
db.logs.count()
db.getMongo().setSlaveOk()
db.logs.count()

43.フェイルオーバテスト プロセスを殺す kill