

TP1 CouchDB (3h) (Correction)

1. Préalable

Vous travaillerez sur le cluster CouchDB (version 3.1) installé sur les serveurs de la DSIN, et accessible aux adresses des trois nœuds disponibles dans le cluster qui sont :

prodpeda-couchdb3-1.infra.umontpellier.fr:5984,
prodpeda-couchdb3-2.infra.umontpellier.fr:5984
et prodpeda-couchdb3-3.infra.umontpellier.fr:5984.

2. Création et alimentation de la base de données

Vous définirez au préalable une variable d'environnement nommée COUCH3 dans votre .bashrc pour vous simplifier l'accès au cluster en ligne de commande. Les informations concernant la chaîne de connexion sont données dans un fichier textuel à part.

- Vous définirez une base de données nommée **occitanie** en la préfixant avec votre nom de famille (tout en minuscules). Réalisez cette opération en ligne de commandes avec curl ;

```
-- attention bd en minuscules _ ici un exemple avec le prefixe zoe
curl -X PUT $COUCH3/zoe_occitanie
-- reponse du systeme
{"ok":true}

-- verifier avec
-- curl -X GET $COUCH3/_all_dbs
```

Listing 1 – Exemple de création de bd

- Vous exploiterez le mode "ajout par lots" via curl pour insérer les documents des fichiers herault.json, gard.json, aveyron.json, hauteGaronne.json et regions_partiel.json dans la base de données.

```
-- exemple de chargement dans zoe_occitanie
curl -X POST $COUCH3/zoe_occitanie/_bulk_docs -d @herault.json -H "Content-Type:
application/json"
```

Listing 2 – Exemple de chargement par lots

Le contexte applicatif est le même que celui abordé pour Neo4J, à savoir des informations administratives sur des communes françaises. Des exemples de documents json extraits des fichiers de commune et de région sont proposés ici. Vous noterez le champ **type** qui est une tentative de définition d'un schéma.

```
{
```

```

    "_id": "34100",
    "codeInsee": "34100",
    "longitude": "2.8925",
    "latitude": "43.4858",
    "type": "commune",
    "nom": "FERRIERES-POUSSAROU",
    "dep": "34",
    "old_reg": "91",
    "populations": [
      {
        "pop_1975": 38.0961632716736
      },
      {
        "pop_1985": 35.8391614288578
      },
      {
        "pop_1995": 45.2742630644936
      },
      {
        "pop_2005": 56.941818739832
      },
      {
        "pop_2010": 60.6582030010204
      }
    ]
  }
}

```

Listing 3 – Exemple de commune

```

{
  "_id": "91",
  "reg": "91",
  "chef_lieu_reg": "34172",
  "type": "old_region",
  "new_reg": "occitanie",
  "nom_reg": "LANGUEDOC-ROUSSILLON",
  "departements": [
    {
      "dep": "34",
      "nom_dep": "HERAULT",
      "chef_lieu_dep": "34172"
    },
    {
      "dep": "30",
      "nom_dep": "GARD",
      "chef_lieu_dep": "30189"
    },
    {
      "dep": "11",
      "nom_dep": "AUDE",
      "chef_lieu_dep": "11069"
    },
    {
      "dep": "66",
      "nom_dep": "PYRENEES-ORIENTALES",
      "chef_lieu_dep": "66136"
    }
  ]
}

```

Listing 4 – Exemple d'ancienne région

```
{
  "_id": "occitanie",
  "chef_lieu_reg": "31555",
  "nom_reg": "Occitanie",
  "type": "region",
  "president": {
    "nom": "Delga",
    "prenom": "Carole"
  }
}
```

Listing 5 – Exemple de nouvelle région

3. Appropriation de la base de données

Vous répondrez aux questions suivantes en vous aidant de requêtes CURL :

1. lister les informations générales concernant le serveur couchdb, à l'aide du mécanisme GET

```
curl -X GET $COUCH3/
```

Listing 6 – Informations serveur

2. lister les informations générales concernant la base occitanie, à l'aide du mécanisme GET. Pouvez vous connaître le nombre de documents contenus dans la base occitanie ?

```
curl -X GET $COUCH3/zoe_occitanie/
avec "doc_count"
-- remarquer la philosophie tout est document y compris la base
```

Listing 7 – Informations bd

3. lister tous les documents de la BD

```
curl -X GET $COUCH3/zoe_occitanie/_all_docs
```

Listing 8 – Liste des documents

4. faire afficher le contenu d'un document. Quel est son numéro de révision ? Comment savoir si ce document a déjà été modifié ?

```
curl -X GET $COUCH3/zoe_occitanie/73/
regarder l'attribut "_rev" et sa valeur : le premier chiffre est un compteur
1, 2, 3 etc en fonction des modifications
```

Listing 9 – Infos document

4. Définition de vues

Vous définirez des vues (en javascript) dans votre base pour consulter les documents décrivant les régions et les communes. Ce travail peut se faire soit en ligne de commande avec curl, soit avec l'interface Web Fauxton (depuis un navigateur à l'adresse [prodpeda-couchdb3-2.infra.umontpellier.fr :5984/_utils/](http://prodpeda-couchdb3-2.infra.umontpellier.fr:5984/_utils/)).

4.1 MAP seulement

1. donnez toutes les informations sur les régions (de type `old_region`) de la base

```
-- juste la vue javascript a coder
function(doc) { if (doc.type=="old_region") {
  emit(doc._id, doc); }
}

-- exemple en ligne de commande avec le document json correspond a la vue
curl -X PUT $COUCH3/zoe_occitanie/_design/Q1 -d '{"views":{"v1":
  {"map":"function(doc) { if (doc.type=='old_region') {emit(doc._id, doc) ;} }"
  }}}' -H "Content-Type: application/json"
{"ok":true,"id":"_design/Q1","rev":"1-2ae053c303a0125fba76a79a416fe39d"}

-- possibilite de mettre le document json dans un fichier
curl -X PUT $COUCH3/zoe_occitanie/_design/d2 -d @vue.json -H "Content-Type:
  application/json"
{"ok":true,"id":"_design/d2","rev":"1-aa50cfb893896efbe5d9cc0f3ab99214"}
```

Listing 10 – Map1

2. donner les noms (clés) et latitude et longitude de chaque commune

```
-- juste la vue javascript a coder
function(doc) { if (doc.type=="commune") {
  emit(doc.nom, {"latitude":doc.latitude, "longitude":doc.longitude} ); }
}
```

Listing 11 – Map2

3. donner le code insee (clé), le département, la latitude et la longitude de MONTPELLIER (nom de la commune)

```
-- juste la vue javascript a coder
function(doc) { if (doc.nom=="MONTPELLIER") {
  emit(doc.codeInsee, [{"departement":doc.dep}, {"latitude":doc.latitude},
    {"longitude":doc.longitude}]); }
}
```

Listing 12 – Map3

4. donnez le nom et le prénom de la présidente de la région Occitanie

```
-- juste la vue javascript a coder
function (doc) {
  if(doc.type == 'region'){
    if(doc.nom_reg== 'Occitanie'){
      var val = doc.president.prenom + '-' + doc.president.nom
      emit(doc.nom_reg, val);
    }
  }
}
```

```
}
}
```

Listing 13 – Map4

4.2 MAP et REDUCE

1. donner le nombre de communes au total puis par département et enfin par région (old_region)

```
-- le format json de la vue est donne
"REDUCE_Un":{"map":"function(doc) { if (doc.type=='commune')\n emit([doc.old_reg,
    doc.dep, doc._id], 1);\n}",
"reduce":"_count"}
```

Listing 14 – MR1

2. donner le nombre d'habitants par commune en 1985

```
-- la vue javascript a coder identique au partitionnement niveau 3
"REDUCE_Deux":{"map":"function(doc) { if (doc.type=='commune')\n
for (var i = 0; i<doc.populations.length; i++)\n{ \nvar pop = doc.populations[i];\n
if (pop.pop_1985)\n emit([doc.old_reg, doc.dep, doc.nom], pop.pop_1985);\n\n}\n}",
"reduce":"_sum"}
```

Listing 15 – MR2

3. donner le nombre d'habitants par département en 1985

```
-- partitionnement niveau 2
```

Listing 16 – MR3

4. donner le nombre d'habitants par région (anciennes régions) en 1985

```
-- partitionnement niveau 1
```

Listing 17 – MR4

4.3 Autres requêtes

1. donner les communes qui ont vu leurs populations décroître entre 1985 et 1995

```
function(doc)
{ if (doc.type=='commune')
{ for (var t in doc.populations)
{ var pop = doc.populations[t];
if (pop.pop_1975)
{var p75 = doc.populations[t].pop_1975;}
if (pop.pop_1985)
{var p85 = doc.populations[t].pop_1985;}
}
var test = p85 - p75;
if (test < 0)
{emit(doc._id, [doc.nom, p75, p85, test]);}
}
}
```

Listing 18 – Communes en déclin

2. donner les informations sur la nouvelle région Occitanie ainsi que sur les anciennes régions Languedoc-Roussillon et Midi-Pyrénées (forme de jointure)

```
"REDUCE_Quatre":{"map":"function(doc) { if (doc.type=='region')\n
{ \n emit([doc._id, 0], doc.nom_reg);\n}\n
if (doc.type=='old_region')\n{ \n emit([doc.new_reg, 1], doc.nom_reg);\n}\n}"}}
```

Listing 19 – Collation view

5. Distribution de la base de données

Vous répondrez aux questions suivantes en vous aidant de requêtes CURL :

1. quels sont les nœuds mobilisés pour l'organisation du serveur, et la gestion de la base ?

```
curl -X GET $COUCH3/_membership
-- les 3 noeuds prodpeda-couchdb3-1.infra.umontpellier.fr,
   prodpeda-couchdb3-2.infra.umontpellier.fr,
-- prodpeda-couchdb3-3.infra.umontpellier.fr
-- participent a la mise en oeuvre du systeme et des bases de donnees
```

Listing 20 – Les nœuds

2. Combien de partitions sont définies (avant recopie) ? Quel est le nombre de copies ? Combien de partitions répliquées sont définies ?

```
2 partitions qui sont repliquees sur les 3 noeuds
curl -X GET $COUCH3/zoe_occitanie/_shards
{"shards":{"00000000-7fffffff":["couchdb@prodpeda-couchdb3-1.infra.umontpellier.fr",
"couchdb@prodpeda-couchdb3-2.infra.umontpellier.fr",
"couchdb@prodpeda-couchdb3-3.infra.umontpellier.fr"],
"80000000-ffffffff":["couchdb@prodpeda-couchdb3-1.infra.umontpellier.fr",
"couchdb@prodpeda-couchdb3-2.infra.umontpellier.fr","couchdb@prodpeda-
couchdb3-3.infra.umontpellier.fr"]}}
```

Listing 21 – Shards et copies

3. Comment savoir dans quelle(s) partition(s) se trouve un des documents de la base ?

```
curl -X GET $COUCH3/zoe_occitanie/_shards/34172
{"range":"00000000-7fffffff","nodes":["couchdb@prodpeda-couchdb3-1.infra.umontpellier.fr",
"couchdb@prodpeda-couchdb3-2.infra.umontpellier.fr",
"couchdb@prodpeda-couchdb3-3.infra.umontpellier.fr"]}}
dans la premiere partition intervalle "00000000-7fffffff"
```

Listing 22 – Shards et documents

4. Est ce que des copies de toutes les partitions sont présentes sur tous les nœuds ?

```
oui visible avec curl -X GET $COUCH3/zoe_occitanie/_shards
```

Listing 23 – Shards et nœuds

5. Est que toutes les lectures sont consistentes ?

```
dans la grande majorite des cas, oui. On lit 2 copies (quorum en lecture) avant de
retourner la reponse, il y a au moins une copie qui est a jour sur les 3
partitions et deux qui pour des temps tres courts peuvent ne pas etre a jour
mais on a de fortes probabilites de lire une copie qui est la derniere sur les
deux lues
voir "cluster":{"q":2,"n":3,"w":2,"r":2} quand curl -X GET $COUCH3/zoe_occitanie/

-- avec q = nombre de shards, n = nombre de copies, r = quorum en lecture (nombre
de shards parcourus avant lecture),
w = quorum en ecriture (nombre de shards lus avant ecriture)
```

Listing 24 – Nombre de lectures