Сконфігуруйте 3 інстанси MongoDB як шарди у двох варіантах:

```
mongos> use store;
switched to db store
mongos> sh.enableSharding("store")
{
        "ok" : 1,
        "operationTime" : Timestamp(1622641715, 5),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1622641715, 5),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
mongos> sh.status();
--- Sharding Status ---
  sharding version: {
        "_id" : 1,
        "minCompatibleVersion" : 5,
        "currentVersion" : 6,
        "clusterId" : ObjectId("60b78b8b14411e9bed23e657")
  }
  shards:
        {  "_id" : "rs1",  "host" : "rs1/m1:8001",  "state" : 1 }
        {  "_id" : "rs2",  "host" : "rs2/m2:8002",  "state" : 1 }
        {  "_id" : "rs3",  "host" : "rs3/m3:8003",  "state" : 1 }
  active mongoses:
        "4.2.7" : 1
  autosplit:
        Currently enabled: yes
  balancer:
        Currently enabled:  yes
        Currently running:  no
        Failed balancer rounds in last 5 attempts:  0
        Migration Results for the last 24 hours:
                No recent migrations
  databases:
        {  "_id" : "config",  "primary" : "config",  "partitioned" : true }
        {  "_id" : "store",  "primary" : "rs1",  "partitioned" : true,  "version" : {  "uuid" : UUID("31058be7-1437-49ee-a345-8b11f5066659"),  "lastMod" : 1 } }

mongos>
```

Для колекції замовлень використовуючи, наприклад, поштовий індекс (або суму замовлення чи інше безперервне поле) у якості shard key налаштуйте Ranged Sharding

```
mongos> sh.addShardTag("rs1", "lt_50");
{
        "ok" : 1,
        "operationTime" : Timestamp(1622641906, 1),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1622641906, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
mongos> sh.addShardTag("rs2", "50_100");
{
        "ok" : 1,
        "operationTime" : Timestamp(1622641906, 2),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1622641906, 2),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
mongos> sh.addShardTag("rs3", "gt_100");
{
        "ok" : 1,
        "operationTime" : Timestamp(1622641908, 17),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1622641908, 17),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
mongos>
```

```
mongos> sh.addTagRange("store.orders", {total: MinKey}, {total: 50}, "lt_50");
{
        "ok" : 1,
        "operationTime" : Timestamp(1622642121, 1),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1622642121, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
mongos> sh.addTagRange("store.orders", {total: 51}, {total: 100}, "50_100");
{
        "ok" : 1,
        "operationTime" : Timestamp(1622642121, 2),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1622642121, 2),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
mongos> sh.addTagRange("store.orders", {total: 101}, {total: MaxKey}, "gt_50");
{
        "ok" : 0,
        "errmsg" : "zone gt_50 does not exist",
        "code" : 177,
        "codeName" : "ZoneNotFound",
        "operationTime" : Timestamp(1622642122, 26),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1622642122, 26),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
mongos>
```

```
mongos> sh.addTagRange("store.orders", {total: 101}, {total: MaxKey}, "gt_100");
{
        "ok" : 1,
        "operationTime" : Timestamp(1622642167, 1),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1622642167, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
```

```
mongos> sh.shardCollection("store.orders", {total: 1});
{
        "collectionsharded" : "store.orders",
        "collectionUUID" : UUID("863f0e0b-697a-4b23-a958-13478037bf35"),
        "ok" : 1,
        "operationTime" : Timestamp(1622642715, 41),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1622642715, 41),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
mongos>
```

```
mongos> sh.status()
--- Sharding Status ---
  sharding version: {
        "_id" : 1,
        "minCompatibleVersion" : 5,
        "currentVersion" : 6,
        "clusterId" : ObjectId("60b78b8b14411e9bed23e657")
  }
  shards:
        {  "_id" : "rs1",  "host" : "rs1/m1:8001",  "state" : 1,  "tags" : [ "lt_50" ] }
        {  "_id" : "rs2",  "host" : "rs2/m2:8002",  "state" : 1,  "tags" : [ "50_100" ] }
        {  "_id" : "rs3",  "host" : "rs3/m3:8003",  "state" : 1,  "tags" : [ "gt_100" ] }
  active mongoses:
        "4.2.7" : 1
  autosplit:
        Currently enabled: yes
  balancer:
        Currently enabled:  yes
        Currently running:  no
        Failed balancer rounds in last 5 attempts:  0
        Migration Results for the last 24 hours:
                682 : Success
  databases:
        {  "_id" : "config",  "primary" : "config",  "partitioned" : true }
                config.system.sessions
                        shard key: { "_id" : 1 }
                        unique: false
                        balancing: true
                        chunks:
                                rs1     342
                                rs2     341
                                rs3     341
                        too many chunks to print, use verbose if you want to force print
        {  "_id" : "store",  "primary" : "rs1",  "partitioned" : true,  "version" : {  "uuid" : UUID("31058be7-1437-49ee-a345-8b11f5066659"),  "lastMod" : 1 } }
                store.orders
                        shard key: { "total" : 1 }
                        unique: false
                        balancing: true
                        chunks:
                                rs1     2
                                rs2     2
                                rs3     1
                        { "total" : { "$minKey" : 1 } } -->> { "total" : 50 } on : rs1 Timestamp(1, 0)
                        { "total" : 50 } -->> { "total" : 51 } on : rs1 Timestamp(1, 1)
                        { "total" : 51 } -->> { "total" : 100 } on : rs2 Timestamp(1, 2)
                        { "total" : 100 } -->> { "total" : 101 } on : rs2 Timestamp(1, 3)
                        { "total" : 101 } -->> { "total" : { "$maxKey" : 1 } } on : rs3 Timestamp(1, 4)
                         tag: lt_50  { "total" : { "$minKey" : 1 } } -->> { "total" : 50 }
```

```
mongos> db.orders.getShardDistribution()

Shard rs1 at rs1/m1:8001
 data : 0B docs : 0 chunks : 2
 estimated data per chunk : 0B
 estimated docs per chunk : 0

Shard rs2 at rs2/m2:8002
 data : 0B docs : 0 chunks : 2
 estimated data per chunk : 0B
 estimated docs per chunk : 0

Shard rs3 at rs3/m3:8003
 data : 0B docs : 0 chunks : 1
 estimated data per chunk : 0B
 estimated docs per chunk : 0

Totals
 data : 0B docs : 0 chunks : 5
 Shard rs1 contains 0% data, 0% docs in cluster, avg obj size on shard : 0B
 Shard rs2 contains 0% data, 0% docs in cluster, avg obj size on shard : 0B
 Shard rs3 contains 0% data, 0% docs in cluster, avg obj size on shard : 0B
```

```
mongos> db.orders.insertMany([
...     {total: 20},
...     {total: 70},
...     {total: 120},
... ]);
{
        "acknowledged" : true,
        "insertedIds" : [
                ObjectId("60b7928a8d8f88cde85c2b7b"),
                ObjectId("60b7928a8d8f88cde85c2b7c"),
                ObjectId("60b7928a8d8f88cde85c2b7d")
        ]
}
mongos>
```

```
mongos> use store;
switched to db store
mongos> var totals = [20, 70, 120];
mongos> for (var i of totals) {
...     var shards = db.orders.find({total:i}).explain()['queryPlanner']['winningPlan']['shards'];
...     print("total shards of " + i.toString() + " " + shards.length.toString());
...     print(i.toString() + " shard name is " + shards[0]['shardName']);
... }
total shards of 20 1
20 shard name is rs1
total shards of 70 1
70 shard name is rs2
total shards of 120 1
120 shard name is rs3
```

Для колекції товарів використовуючи модель (або тип, або виробник чи інше дискретне поле) у якості shard key налаштуйте Zones

```
mongos> sh.addShardTag("rs1", "edibles");
{
        "ok" : 1,
        "operationTime" : Timestamp(1622645660, 1),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1622645660, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
mongos> sh.addShardTag("rs2", "edibles");
{
        "ok" : 1,
        "operationTime" : Timestamp(1622645661, 1),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1622645661, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
mongos> sh.addShardTag("rs3", "ediblesButOnce");
{
        "ok" : 1,
        "operationTime" : Timestamp(1622645661, 2),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1622645661, 2),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
```

```
mongos> sh.addTagRange(
...     "store.items",
...     { "category" : "meat", "_id" : MinKey },
...     { "category" : "meat", "_id" : MaxKey },
...     "edibles"
... );
{
        "ok" : 1,
        "operationTime" : Timestamp(1622645661, 3),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1622645661, 3),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
mongos> sh.addTagRange(
...     "store.items",
...     { "category" : "alcohol", "_id" : MinKey },
...     { "category" : "alcohol", "_id" : MaxKey },
...     "edibles"
... );
{
        "ok" : 1,
        "operationTime" : Timestamp(1622645661, 4),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1622645661, 4),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
```

```
mongos> sh.addTagRange(
...     "store.items",
...     { "category" : "washing powder", "_id" : MinKey },
...     { "category" : "washing powder", "_id" : MaxKey },
...     "ediblesButOnce"
... );
{
        "ok" : 1,
        "operationTime" : Timestamp(1622645662, 1),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1622645662, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
mongos>
```

```
mongos> use store;
switched to db store
mongos> db.items.insertMany([
...     {"category": "meat", "name": "pork"},
...     {"category": "meat", "name": "chicken"},
...     {"category": "alcohol", "name": "vodka"},
...     {"category": "washing powder", "name": "tide"},
...     {"category": "washing powder", "name": "persil"},
...     {"category": "washing powder", "name": "pervol"},
...     {"category": "washing powder", "name": "zvuchyainuy poroshok"},
... ]);
{
        "acknowledged" : true,
        "insertedIds" : [
                ObjectId("60b79d36568d0f1e50441755"),
                ObjectId("60b79d36568d0f1e50441756"),
                ObjectId("60b79d36568d0f1e50441757"),
                ObjectId("60b79d36568d0f1e50441758"),
                ObjectId("60b79d36568d0f1e50441759"),
                ObjectId("60b79d36568d0f1e5044175a"),
                ObjectId("60b79d36568d0f1e5044175b")
        ]
}
mongos>
```

```
mongos> db.items.ensureIndex({category: 1, _id: 1});
{
        "raw" : {
                "rs1/m1:8001" : {
                        "createdCollectionAutomatically" : false,
                        "numIndexesBefore" : 2,
                        "numIndexesAfter" : 3,
                        "ok" : 1
                }
        },
        "ok" : 1,
        "operationTime" : Timestamp(1622646596, 2),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1622646596, 2),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
mongos> sh.shardCollection("store.items", {category: 1, _id: 1});
{
        "collectionsharded" : "store.items",
        "collectionUUID" : UUID("e51efc05-af04-4173-b53e-62dcb8e8ed4e"),
        "ok" : 1,
        "operationTime" : Timestamp(1622646600, 9),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1622646600, 9),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
mongos>
```

Перевірте появи шард і зон командою sh.status()

```
mongos> sh.status()
--- Sharding Status ---
  sharding version: {
        "_id" : 1,
        "minCompatibleVersion" : 5,
        "currentVersion" : 6,
        "clusterId" : ObjectId("60b78b8b14411e9bed23e657")
  }
  shards:
        {  "_id" : "rs1",  "host" : "rs1/m1:8001",  "state" : 1,  "tags" : [ "lt_50", "edibles" ] }
        {  "_id" : "rs2",  "host" : "rs2/m2:8002",  "state" : 1,  "tags" : [ "50_100", "edibles" ] }
        {  "_id" : "rs3",  "host" : "rs3/m3:8003",  "state" : 1,  "tags" : [ "gt_100", "ediblesButOnce" ] }
  active mongoses:
        "4.2.7" : 1
  autosplit:
        Currently enabled: yes
  balancer:
        Currently enabled:  yes
        Currently running:  no
        Failed balancer rounds in last 5 attempts:  5
        Last reported error:  Could not find host matching read preference { mode: "primary" } for set rs3
        Time of Reported error:  Wed Jun 02 2021 18:30:41 GMT+0300 (EEST)
        Migration Results for the last 24 hours:
                686 : Success
  databases:
```

```
mongos> db.items.getShardDistribution()

Shard rs1 at rs1/m1:8001
 data : 459B docs : 7 chunks : 3
 estimated data per chunk : 153B
 estimated docs per chunk : 2

Shard rs2 at rs2/m2:8002
 data : 60B docs : 1 chunks : 2
 estimated data per chunk : 30B
 estimated docs per chunk : 0

Shard rs3 at rs3/m3:8003
 data : 284B docs : 4 chunks : 2
 estimated data per chunk : 142B
 estimated docs per chunk : 2

Totals
 data : 803B docs : 12 chunks : 7
 Shard rs1 contains 57.16% data, 58.33% docs in cluster, avg obj size on shard : 65B
 Shard rs2 contains 7.47% data, 8.33% docs in cluster, avg obj size on shard : 60B
 Shard rs3 contains 35.36% data, 33.33% docs in cluster, avg obj size on shard : 71B
```

Продемонструйте роботу шардінгу (тобто що записи зберігаються на різних нодах):

- відключити одну з ноди

```
olekthunder@mellon dist_systems_labs/labs/lab6 [master] → docker-compose stop m3
Stopping m3 ... done
olekthunder@mellon dist_systems_labs/labs/lab6 [master] →
```

- спробувати додати записи зі значеннями shard key (Ranged та Zones), що потрапляють на відключену ноду

```
mongos> db.items.insert({"category": "washing powder", "name": "gala"});
WriteResult({
        "nInserted" : 0,
        "writeError" : {
                "code" : 133,
                "errmsg" : "Could not find host matching read preference { mode: \"pri
mary\" } for set rs3"
        }
})
mongos> db.orders.insert({"total": 100500});
we are dooomed, this node is not working, why am I writing this to buffer?WriteResult(
{
        "nInserted" : 0,
        "writeError" : {
                "code" : 133,
                "errmsg" : "Could not find host matching read preference { mode: \"pri
mary\" } for set rs3"
        }
})
```

- спробувати додати записи зі значеннями shard key (Ranged та Zones), що потрапляють на працюючу ноду

```
mongos> db.items.insert({"category": "meat", "name": "goose"});
WriteResult({ "nInserted" : 1 })
mongos> db.orders.insert({"total": 10});
WriteResult({ "nInserted" : 1 })
mongos>
```

- спробувати знайти всі записи з shard key для Zone, яка відповідає ноді яка працює/яка не працює

```
mongos> db.items.find({"category": {$in: ["meat", "alcohol"]}});
{ "_id" : ObjectId("60b79d36568d0f1e50441755"), "category" : "meat", "name" : "pork" }
{ "_id" : ObjectId("60b79d36568d0f1e50441756"), "category" : "meat", "name" : "chicken
" }
{ "_id" : ObjectId("60b7a25b22e28949c7a1343b"), "category" : "meat", "name" : "goose"
}
{ "_id" : ObjectId("60b79d36568d0f1e50441757"), "category" : "alcohol", "name" : "vodk
a" }
mongos> db.items.find({"category": "washing powder"});
request timeout, lazy to wait
```

- спробувати знайти записи для shard key з певного проміжку, який входить до проміжку ноди яка не працює для Ranged Sharding

```
mongos> db.orders.find({total: 100500});
```

Включити відключену ноду та перевірити працездатність запитів з попереднього пункту

```
mongos> db.orders.find({total: {$gt: 100}});
{ "_id" : ObjectId("60b7a4b122e28949c7a1343f"), "total" : 120 }
mongos> db.items.find({"category": "washing powder"});
{ "_id" : ObjectId("60b79d36568d0f1e50441758"), "category" : "washing powder", "name"
: "tide" }
{ "_id" : ObjectId("60b79d36568d0f1e50441759"), "category" : "washing powder", "name"
: "persil" }
{ "_id" : ObjectId("60b79d36568d0f1e5044175a"), "category" : "washing powder", "name"
: "pervol" }
{ "_id" : ObjectId("60b79d36568d0f1e5044175b"), "category" : "washing powder", "name"
: "zvuchyainuy poroshok" }
mongos>
```