第5章 MongoDB分片集群

学习目标

- 熟悉分片集群的基础知识，能够描述分片集群的核心概念；
- 了解分片策略，能够描述范围分片和哈希分片的实现方式；
- 掌握分片集群的部署，能够基于 Linux 操作系统部署分片集群；
- 掌握分片集群的操作，能够使用不同分片策略对集合执行分片操作；
- 熟悉访问控制，能够采用 Keyfiles 身份验证方法启用分片集群的访问控制。

副本集虽然提供了高可用性和数据冗余，并能通过水平扩展提升读性能，但由于写操作仅限于主节点，且各节点数据完全一致，因此无法通过水平扩展提升写性能和存储容量。对于大规模数据和高并发写操作的场景，副本集显得力不从心。为此，MongoDB 提供了另一种分布式架构——分片集群(sharded cluster)。本章将对分片集群的概念和部署方式进行详细讲解。

5.1 分片集群概述

分片集群可以利用分片操作将集合拆分为多个数据块(chunk)，每个数据块包含了集合的部分文档。为了实现负载均衡，分片集群会尝试在各个分片(shard)之间均匀分配这些数据块。分片集群中的每个分片都可以独立处理客户端发送的读写请求，因此，当数据规模或吞吐量增长时，可以通过添加分片进行水平扩展，以提升分片集群的存储容量和读写性能。

分片集群主要由分片、路由(router)和配置服务器(config servers)组成。一个分片集群可以包含一个配置服务器，以及多个分片和路由，其中至少包含两个分片，以实现分布式数据存储。

下面是包含两个分片、一个路由和一个配置服务器的分片集群架构，如图 5-1 所示。

针对图 5-1 中分片集群架构的核心概念进行如下讲解。

(1) 分片负责存储数据块和未被拆分的集合。在分片集群中，未被拆分的集合将完整地存储在一个分片上。

(2) 路由可以被看作客户端与分片集群之间的桥梁，当客户端向分片集群发送请求时，路由会通过配置服务器获取相关数据块或集合的元数据，从而将请求精准地分发到正确的分片。分片处理完请求后，将结果返回给路由，路由再将汇总后的结果返回给客户端。

(3) 配置服务器负责存储分片集群的元数据和配置信息，包括分片信息、数据块信息等。

MongoDB 要求分片集群中的分片和配置服务器必须以副本集的形式进行部署。同时，为了确保数据冗余和高可用性，建议将分片和配置服务器都部署为包含三个节点的副本集。例如，图 5-1 的分片集群架构中，分片和配置服务器都是包含一个主节点和两个从节点

的副本集，如图 5-2 所示。

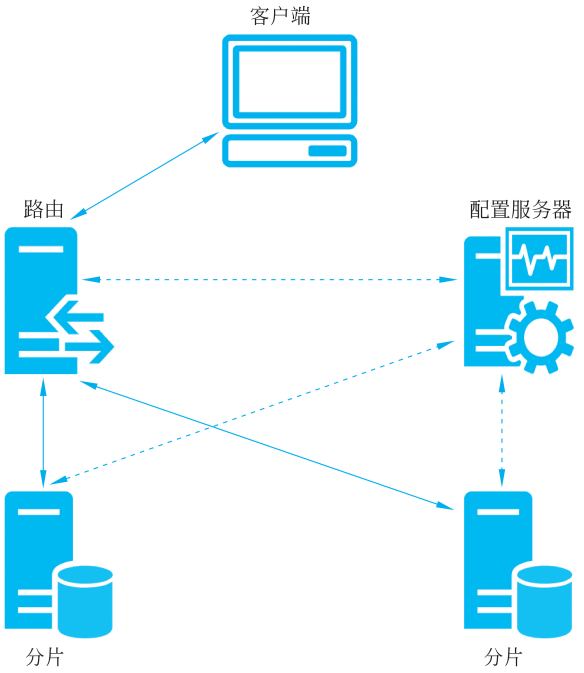

图 5-1　分片集群架构(1)

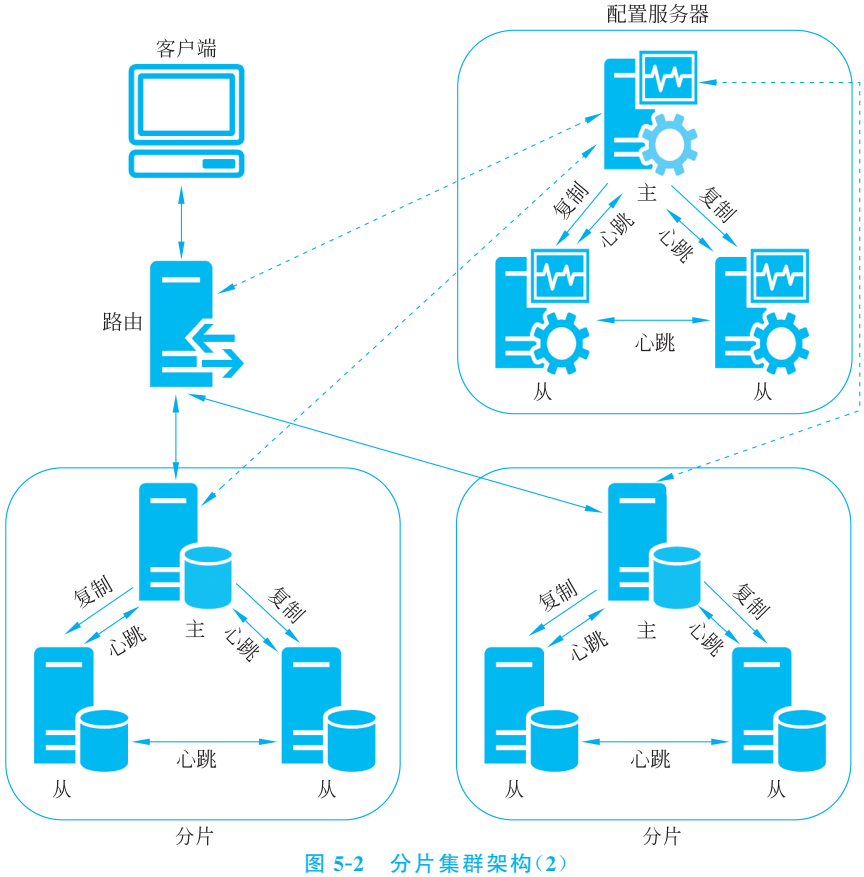

图 5-2　分片集群架构(2)

在图 5-2 中,每个副本集都具有不同的名称并且相互独立。分片和配置服务器中的主节点将处理读写请求。当主节点不可用时,副本集会通过选举机制将某个符合条件的从节点选举为新的主节点。

需要注意的是,配置服务器的副本集中不能包含仲裁节点和延迟节点,并且每个节点必须构建索引。此外,在分片的副本集中,不建议包含仲裁节点。

小提示:通常情况下,不建议在客户端通过直接连接的方式连接分片上的主节点或从节点执行数据操作,因为这些操作可能会导致数据损坏或数据丢失。客户端只能通过直接连接的方式连接分片上的主节点或从节点执行管理或维护操作。

5.2 分片策略

集合被拆分为多少个数据块,取决于集合和数据块的大小。在 MongoDB 7.0 中,数据块的默认大小为 128MB。因此,当集合的大小不超过 128MB 时,集合通常只包含一个数据块,所有插入的文档都会存储在该数据块中。当数据块的大小超过 128MB 时,分片集群会自动将其拆分为两个新的数据块,并为每个数据块分配相应的分片键范围。每个数据块的分片键范围与分片键(shard key)的值密切相关,且分片键范围包括下边界,但不包括上边界。文档会根据分片键的值被插入对应分片键范围的数据块中。

在分片集群中内置了两种用于拆分集合的分片策略,分别是范围分片和哈希分片,具体介绍如下。

1. 范围分片

范围分片(range sharding)根据分片键的值拆分集合。在使用范围分片时,分片键可以是集合中创建单字段索引的键,也可以是集合中复合索引的第一个键,还可以是集合中已创建复合索引的多个键的组合。

例如,根据分片键 x 的值进行范围分片的效果如图 5-3 所示。

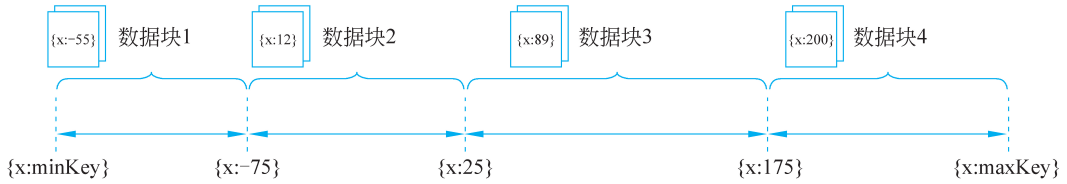

图 5-3 范围分片

从图 5-3 中可以看出,集合包含 4 个数据块,每个数据块对应一个特定的分片键范围。例如,数据块 2 的分片键范围为[-75, 25),集合中所有键 x 的值在该分片键范围内的文档都会被插入该数据块中。

在使用范围分片时,分片键的值相近的文档会被存储在同一个数据块中,这样可以提高范围查询的效率。然而,在批量插入文档时,如果分片键的值集中在某一范围内,可能会导致数据分布不均匀,从而使某个分片承受过重的负载。

需要注意的是,使用范围分片时,分片键在索引中的排序方式必须是升序。

2. 哈希分片

哈希分片(hashed sharding)通过哈希函数对分片键的值进行哈希运算,根据哈希值来拆分集合。在使用哈希分片时,分片键必须是集合中创建哈希索引的键。

例如，根据分片键 x 的值进行哈希分片的效果如图 5-4 所示。

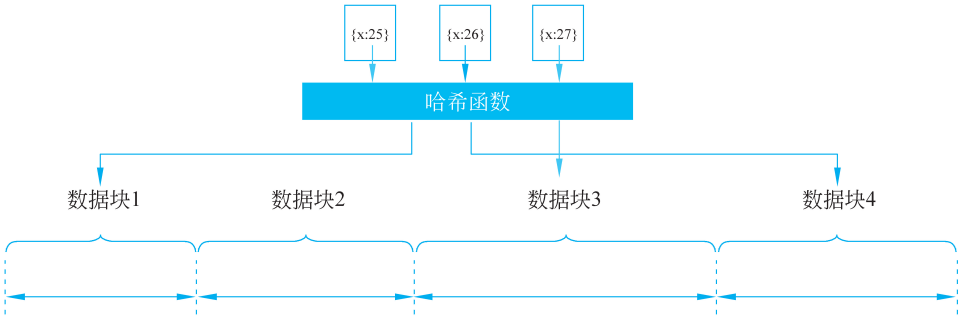

图 5-4　哈希分片

从图 5-4 中可以看出，集合包含 4 个数据块，每个数据块对应一个特定的分片键范围。由于在使用哈希分片时，分片键范围的上边界和下边界是通过哈希运算生成的哈希值，因此图 5-4 中并未标注分片键范围的具体边界。例如，集合中键 x 的值为 25 的文档，通过哈希函数计算后的哈希值会决定其插入数据块 1 中。

使用哈希分片可以实现数据的均匀分布，但在进行范围查询时，通常无法将查询限制在单个分片内。

小提示：分片键的值为 null 的文档和缺失分片键的文档属于同一个数据块。

多学一招：数据块迁移

在分片集群的运行过程中，为了确保数据在所有分片上均匀分布，会通过负载均衡器在各分片之间动态迁移数据块。具体来说，负载均衡器会监控集合在每个分片上的数据量。默认情况下，当某个分片上集合的数据量达到另一个分片的三倍时，会将数据量较多的分片中的部分数据块迁移到数据量较少的分片中，以实现集合数据的均匀分布。

例如，假设每个数据块的默认大小为 128MB，分片 A 存储了集合 A 的 128MB 数据，而分片 B 存储了集合 A 的 384MB 数据，这意味着分片 B 中集合 A 的数据量是分片 A 的三倍。此时，负载均衡器会认为集合 A 的数据分布不均衡，并启动数据块迁移，将分片 B 中集合 A 的一个数据块迁移到分片 A 中。

5.3　部署分片集群

5.3.1　环境准备

本书部署的分片集群由一个配置服务器、一个路由服务器和两个分片组成。为了确保数据冗余和高可用性，分片和配置服务器都被配置为包含三个节点的副本集。副本集由一个主节点和两个从节点组成。

在实际应用中，部署上述架构的分片集群通常需要使用 10 台服务器。其中，1 台服务器用于运行路由，3 台服务器用于运行配置服务器的副本集，其余 6 台服务器用于运行两个分片的副本集。然而，为了便于学习，本书将通过虚拟机 NoSQL01、NoSQL02 和 NoSQL03 来简化分片集群的部署，旨在帮助读者更轻松地理解和实践分片集群的搭建过程，而无须投入大量硬件资源。

在启动分片集群时，每台虚拟机都需要运行多个 MongoDB 服务，为了让读者更清楚地了

解每台虚拟机上端口号的使用情况,避免端口冲突,下面通过一张表格来介绍分片集群中各组件在虚拟机 NoSQL01、NoSQL02 和 NoSQL03 上的端口号分配情况,具体如表 5-1 所示。

表 5-1 分片集群中各组件的端口号分配情况

虚拟机	配置服务器	路由	分片 1	分片 2
NoSQL01	27021(从节点)		27018(主节点)	27019(从节点)
NoSQL02	27021(主节点)	21020	27018(从节点)	27019(从节点)
NoSQL03	27021(从节点)		27018(从节点)	27019(主节点)

在本书中,部署分片集群的环境准备工作主要包括创建所需目录和安装 MongoDB。由于在第 4 章的操作过程中,已经在虚拟机 NoSQL01、NoSQL02 和 NoSQL03 上安装了 MongoDB,所以这里将重点介绍创建所需目录的相关操作。

为了避免在 NoSQL01、NoSQL02 和 NoSQL03 上部署分片集群时各组件的配置文件、日志文件和数据文件存储目录互相混淆,我们需要在 NoSQL01、NoSQL02 和 NoSQL03 上分别创建相关目录,具体内容如下。

(1) 在虚拟机 NoSQL01 和 NoSQL03 上执行下列命令,创建相关目录。

```
#存放分片集群相关文件
mkdir /export/servers/shardCluster
#存放配置服务器的配置文件
mkdir /export/servers/shardCluster/configServer
#存放配置服务器的数据文件
mkdir /export/servers/shardCluster/configServer/db
#存放配置服务器的日志文件
mkdir /export/servers/shardCluster/configServer/logs
#存放分片 1 的配置文件
mkdir /export/servers/shardCluster/shard1
#存放分片 1 的数据文件
mkdir /export/servers/shardCluster/shard1/db
#存放分片 1 的日志文件
mkdir /export/servers/shardCluster/shard1/logs
#存放分片 2 的配置文件
mkdir /export/servers/shardCluster/shard2
#存放分片 2 的数据文件
mkdir /export/servers/shardCluster/shard2/db
#存放分片 2 的日志文件
mkdir /export/servers/shardCluster/shard2/logs
```

(2) 在虚拟机 NoSQL02 上执行下列命令,创建相关目录。

```
mkdir /export/servers/shardCluster
mkdir /export/servers/shardCluster/configServer
mkdir /export/servers/shardCluster/configServer/db
mkdir /export/servers/shardCluster/configServer/logs
#存放路由的配置文件
mkdir /export/servers/shardCluster/mongos
#存放路由的日志文件
mkdir /export/servers/shardCluster/mongos/logs
mkdir /export/servers/shardCluster/shard1
mkdir /export/servers/shardCluster/shard1/db
mkdir /export/servers/shardCluster/shard1/logs
mkdir /export/servers/shardCluster/shard2
mkdir /export/servers/shardCluster/shard2/db
mkdir /export/servers/shardCluster/shard2/logs
```

脚下留心:查看端口号占用情况

在为分片集群中的各组件分配端口号之前,需要确保这些端口号不会与系统中其他服

务使用的端口号发生冲突。可以通过在虚拟机上安装 netstat 工具来查看端口号占用情况，具体命令如下。

```
#安装 netstat 工具
yum install net-tools -y
#查看端口号占用情况
netstat -ant
```

5.3.2　部署配置服务器

在分片集群中，配置服务器必须以副本集的形式进行部署。因此，配置服务器实际上是一组协同工作的 MongoDB 服务，它们共同维护分片集群中同一份元数据的多个副本。当用户通过配置文件或命令行选项启动 MongoDB 服务时，不仅需要添加基础配置和副本集的相关配置，还需要添加分片集群的相关配置。有关基础配置和副本集配置的详细信息，可以参考 3.2 节和 4.3 节，下面主要对分片集群的核心配置进行介绍，具体内容如下。

（1）如果用户通过配置文件启动 MongoDB 服务，那么必须在配置文件中添加配置文件选项 sharding，该选项包含参数 clusterRole，用于指定 MongoDB 服务在分片集群中的角色。参数 clusterRole 的可选值包括 configsvr 和 shardsvr，前者表示配置服务器，后者表示分片。

（2）如果用户通过命令行选项启动 MongoDB 服务，则必须添加命令行选项--configsvr，表示当前 MongoDB 服务为在分片集群中配置服务器。

接下来演示如何在虚拟机 NoSQL01、NoSQL02 和 NoSQL03 上部署配置服务器，具体操作步骤如下。

1. 配置 MongoDB 服务运行参数

本书重点介绍如何通过配置文件启动 MongoDB 服务。在三台虚拟机上创建配置文件的操作步骤如下。

（1）在三台虚拟机的/export/servers/shardCluster/configServer 目录中创建并编辑配置文件 configServer.conf。分别在这三台虚拟机上执行如下命令。

```
vi /export/servers/shardCluster/configServer/configServer.conf
```

（2）在虚拟机 NoSQL01 的配置文件 configServer.conf 中添加如下内容。

```
systemLog:
  destination: file
  path: /export/servers/shardCluster/configServer/logs/configServer.log
  logAppend: true
net:
  bindIp: nosql01
  port: 27021
storage:
  dbPath: /export/servers/shardCluster/configServer/db
processManagement:
  fork: true
replication:
  replSetName: config
sharding:
  clusterRole: configsvr
```

上述内容添加完成后，保存并退出配置文件 configServer.conf。

（3）在虚拟机 NoSQL02 的配置文件 configServer.conf 中添加如下内容。

```
systemLog:
  destination: file
  path: /export/servers/shardCluster/configServer/logs/configServer.log
```

```
  logAppend: true
net:
  bindIp: nosql02
  port: 27021
storage:
  dbPath: /export/servers/shardCluster/configServer/db
processManagement:
  fork: true
replication:
  replSetName: config
sharding:
  clusterRole: configsvr
```

上述内容添加完成后,保存并退出配置文件 configServer.conf。

(4) 在虚拟机 NoSQL03 的配置文件 configServer.conf 中添加如下内容。

```
systemLog:
  destination: file
  path: /export/servers/shardCluster/configServer/logs/configServer.log
  logAppend: true
net:
  bindIp: nosql03
  port: 27021
storage:
  dbPath: /export/servers/shardCluster/configServer/db
processManagement:
  fork: true
replication:
  replSetName: config
sharding:
  clusterRole: configsvr
```

上述内容添加完成后,保存并退出配置文件 configServer.conf。

2. 启动配置服务器

在三台虚拟机上通过配置文件 configServer.conf 启动 MongoDB 服务,分别在这三台虚拟机上执行如下命令。

```
mongod -f /export/servers/shardCluster/configServer/configServer.conf
```

上述命令执行完成后,若三台虚拟机均输出了 child process started successfully 信息,则说明 MongoDB 服务已成功启动。

3. 初始化配置服务器的副本集

通过 mongosh 连接任意虚拟机中配置服务器的 MongoDB 服务,并执行初始化副本集的操作。在初始化配置服务器的副本集时,除了指定副本集名称和各节点的配置信息外,还需添加一个参数 configsvr,并将其值设置为 true,以标明该副本集是分片集群中的配置服务器。初始化配置服务器的副本集的操作步骤如下。

(1) 连接虚拟机 NoSQL01 中配置服务器的 MongoDB 服务,在虚拟机 NoSQL01 上执行如下命令。

```
mongosh --host nosql01 --port 27021
```

(2) 初始化配置服务器的副本集,在 mongosh 执行如下命令。

```
rs.initiate( {
  _id : "config",
  configsvr: true,
  members: [
      {_id: 0, host: "nosql01:27021", priority: 2},
```

```
        {_id: 1, host: "nosql02:27021", priority: 3},
        {_id: 2, host: "nosql03:27021"}
    ]
})
```

上述命令中，指定地址为 nosql02：27021 的节点优先级最高，其目的是初始化完成后通过选举机制将其选举为主节点。读者可以在 mongosh 中执行 rs.status()命令查看副本集中各节点的状态，这里不再赘述。

至此，完成部署配置服务器的相关操作。

5.3.3　部署分片

在分片集群中，分片必须以副本集的形式进行部署。因此，分片实际上是一组协同工作的 MongoDB 服务，它们共同维护分片集群中同一份数据的多个副本。当用户通过配置文件启动 MongoDB 服务时，必须在配置文件中添加配置文件选项 sharding，并将参数 clusterRole 的值指定为 shardsvr。若用户通过命令行选项启动 MongoDB 服务，则必须添加命令行选项--shardsvr。

接下来演示如何在虚拟机 NoSQL01、NoSQL02 和 NoSQL03 上部署分片，具体操作步骤如下。

1. 配置分片 1 的 MongoDB 服务运行参数

本书重点介绍如何通过配置文件启动 MongoDB 服务。在三台虚拟机上创建配置文件的操作步骤如下。

（1）在三台虚拟机的/export/servers/shardCluster/shard1 目录中创建并编辑配置文件 shard1.conf，分别在这三台虚拟机上执行如下命令。

```
vi /export/servers/shardCluster/shard1/shard1.conf
```

（2）在虚拟机 NoSQL01 的配置文件 shard1.conf 中添加如下内容。

```
systemLog:
  destination: file
  path: /export/servers/shardCluster/shard1/logs/shard1.log
  logAppend: true
net:
  bindIp: nosql01
  port: 27018
storage:
  dbPath: /export/servers/shardCluster/shard1/db
processManagement:
  fork: true
replication:
  replSetName: shard1
sharding:
  clusterRole: shardsvr
```

上述内容添加完成后，保存并退出配置文件 shard1.conf。

（3）在虚拟机 NoSQL02 的配置文件 shard1.conf 中添加如下内容。

```
systemLog:
  destination: file
  path: /export/servers/shardCluster/shard1/logs/shard1.log
  logAppend: true
net:
  bindIp: nosql02
  port: 27018
```

```
storage:
  dbPath: /export/servers/shardCluster/shard1/db
processManagement:
  fork: true
replication:
  replSetName: shard1
sharding:
  clusterRole: shardsvr
```

上述内容添加完成后,保存并退出配置文件 shard1.conf。

(4) 在虚拟机 NoSQL03 的配置文件 shard1.conf 中添加如下内容。

```
systemLog:
  destination: file
  path: /export/servers/shardCluster/shard1/logs/shard1.log
  logAppend: true
net:
  bindIp: nosql03
  port: 27018
storage:
  dbPath: /export/servers/shardCluster/shard1/db
processManagement:
  fork: true
replication:
  replSetName: shard1
sharding:
  clusterRole: shardsvr
```

上述内容添加完成后,保存并退出配置文件 shard1.conf。

2. 启动分片 1

在三台虚拟机中通过配置文件 shard1.conf 启动 MongoDB 服务,分别在这三台虚拟机上执行如下命令。

```
mongod -f /export/servers/shardCluster/shard1/shard1.conf
```

上述命令执行完成后,若三台虚拟机均输出了 child process started successfully 信息,则说明 MongoDB 服务已成功启动。

3. 初始化分片 1 的副本集

通过 mongosh 连接任意虚拟机中分片 1 的 MongoDB 服务,并执行初始化副本集的操作,具体操作步骤如下。

(1) 连接虚拟机 NoSQL01 中分片 1 的 MongoDB 服务,在虚拟机 NoSQL01 上执行如下命令。

```
mongosh --host nosql01 --port 27018
```

(2) 初始化分片 1 的副本集,在 mongosh 执行如下命令。

```
rs.initiate( {
  _id : "shard1",
  members: [
      {_id: 0, host: "nosql01:27018", priority: 3},
      {_id: 1, host: "nosql02:27018", priority: 2},
      {_id: 2, host: "nosql03:27018"}
  ]
})
```

上述命令中,指定地址为 nosql01:27018 的节点优先级最高,其目的是初始化完成后通过选举机制将其选举为主节点。读者可以在 mongosh 中执行 rs.status()命令查看副本集中各节点的状态,这里不再赘述。

4. 配置分片 2 的 MongoDB 服务运行参数

在三台虚拟机上创建配置文件的操作步骤如下。

(1) 在三台虚拟机的/export/servers/shardCluster/shard2 目录中创建并编辑配置文件 shard2.conf。分别在这三台虚拟机上执行如下命令。

```
vi /export/servers/shardCluster/shard2/shard2.conf
```

(2) 在虚拟机 NoSQL01 的配置文件 shard2.conf 中添加如下内容。

```
systemLog:
  destination: file
  path: /export/servers/shardCluster/shard2/logs/shard2.log
  logAppend: true
net:
  bindIp: nosql01
  port: 27019
storage:
  dbPath: /export/servers/shardCluster/shard2/db
processManagement:
  fork: true
replication:
  replSetName: shard2
sharding:
  clusterRole: shardsvr
```

上述内容添加完成后,保存并退出配置文件 shard2.conf。

(3) 在虚拟机 NoSQL02 的配置文件 shard2.conf 中添加如下内容。

```
systemLog:
  destination: file
  path: /export/servers/shardCluster/shard2/logs/shard2.log
  logAppend: true
net:
  bindIp: nosql02
  port: 27019
storage:
  dbPath: /export/servers/shardCluster/shard2/db
processManagement:
  fork: true
replication:
  replSetName: shard2
sharding:
  clusterRole: shardsvr
```

上述内容添加完成后,保存并退出配置文件 shard2.conf。

(4) 在虚拟机 NoSQL03 的配置文件 shard2.conf 中添加如下内容。

```
systemLog:
  destination: file
  path: /export/servers/shardCluster/shard2/logs/shard2.log
  logAppend: true
net:
  bindIp: nosql03
  port: 27019
storage:
  dbPath: /export/servers/shardCluster/shard2/db
processManagement:
  fork: true
replication:
  replSetName: shard2
sharding:
  clusterRole: shardsvr
```