任务一、

1. 集群规划：

根据题目中的MongoDB集群部署规划图，列出各节点功能及端口号如表1.1所示

表1.1MongoDB集群规划表

|  |  |  |
| --- | --- | --- |
| 节点名称 | 节点功能 | 节点端口 |
| Mongos1 | 提供web服务、集群路由 | 27017 |
| Mongos2 | 提供web服务、集群路由 | 20117 |
| Primary Config | Config Server主节点 | 27119 |
| Secondary Config | Config Server从节点 | 27219 |
| Secondary Config | Config Server从节点 | 27219 |
| Primary Mongod1 | Shard1主节点 | 27018 |
| Secondary Mongod1 | Shard1从节点 | 27118 |
| Arbiter Mongod1 | Shard1仲裁节点 | 27218 |
| Primary Mongod2 | Shard2主节点 | 27318 |
| Secondary Mongod2 | Shard2从节点 | 27418 |
| Arbiter Mongod2 | Shard2仲裁节点 | 27518 |

1. Docker Compose管理集群

完成集群规划表后，采用Docker Compose管理集群容器，集群网络使用Docker Network创建mongo网络，集群中的所有容器均桥接到该网络中。

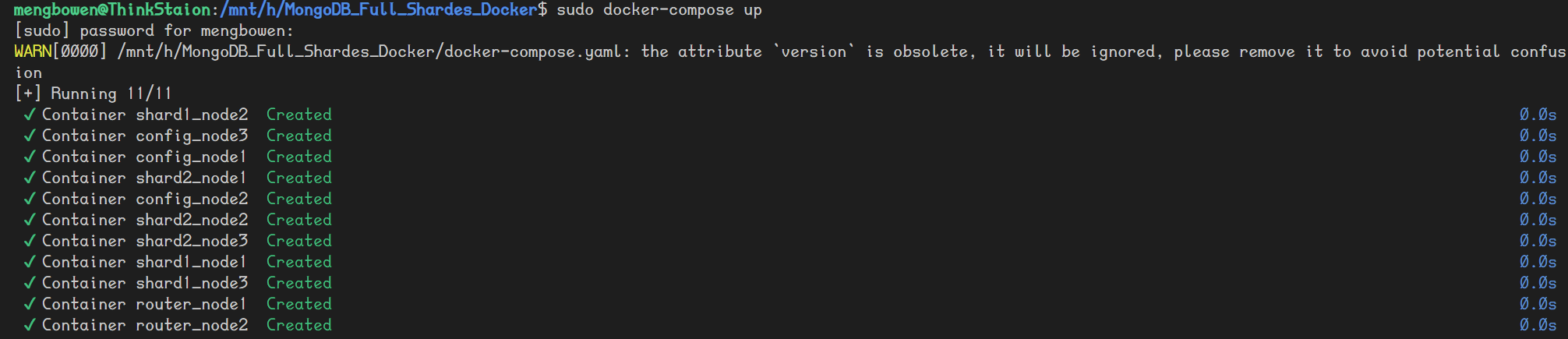
集群的Docker Compose文件如下附件所示







在WSL中使用sudo docker-compose up命令启动容器



1. 集群配置文件编写

Config Server配置文件编写

编写Config Server配置文件，在该配置文件中对Config Server的名称及成员进行配置，在配置文件中指定端口为27019的config节点优先级为2，使其在选举为主配置节点时具有更高优先级



在mongosh中运行配置代码，完成Config Server配置



Shard 1初始化配置文件编写

编写Shard1配置文件，在该配置文件中对Shard1中的节点名称及成员进行配置，在配置文件中分别指定主节点、从节点及仲裁节点的配置，同时设定端口27018的节点优先级为2，使其更易被选举为主节点，配置文件如下所示



在mongosh中运行配置文件，完成Shard1初始化

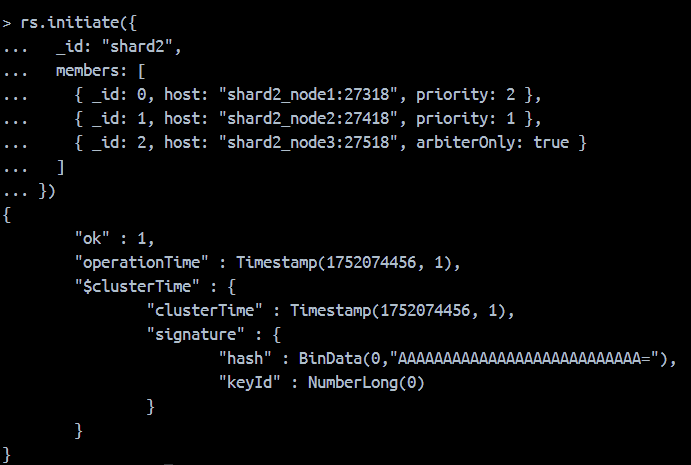


Shard2初始化配置文件编写

由于Shard2的架构与Shard1没有区别，仅节点端口有区别，因此不再对配置文件进行详细描述，仅给出具体的配置文件内容



在mongosh中运行配置文件，完成Shard2初始化

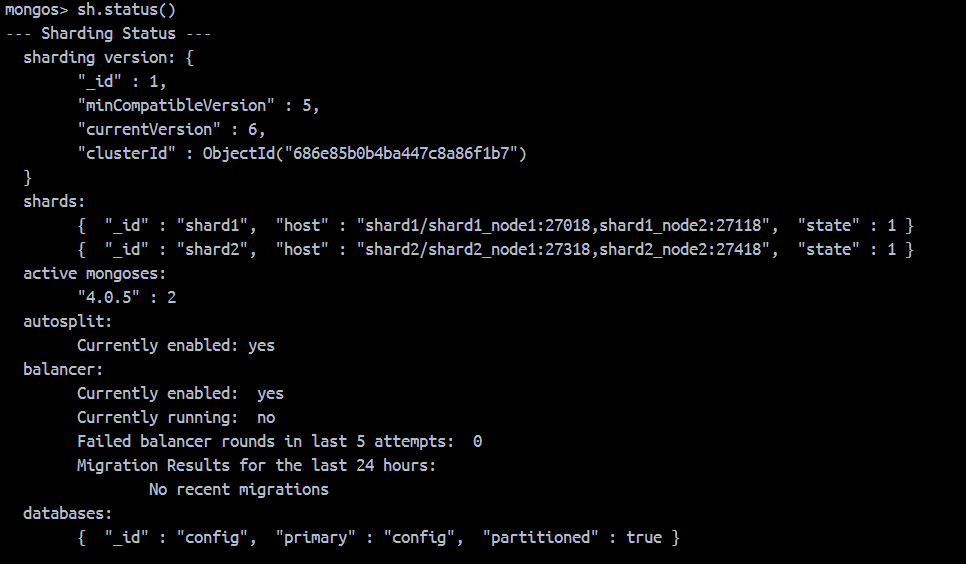


Router节点初始化配置文件编写

Router节点的主要任务时为集群提供路由和负载均衡服务，在Router节点配置集群分片服务的配置文件如下所示

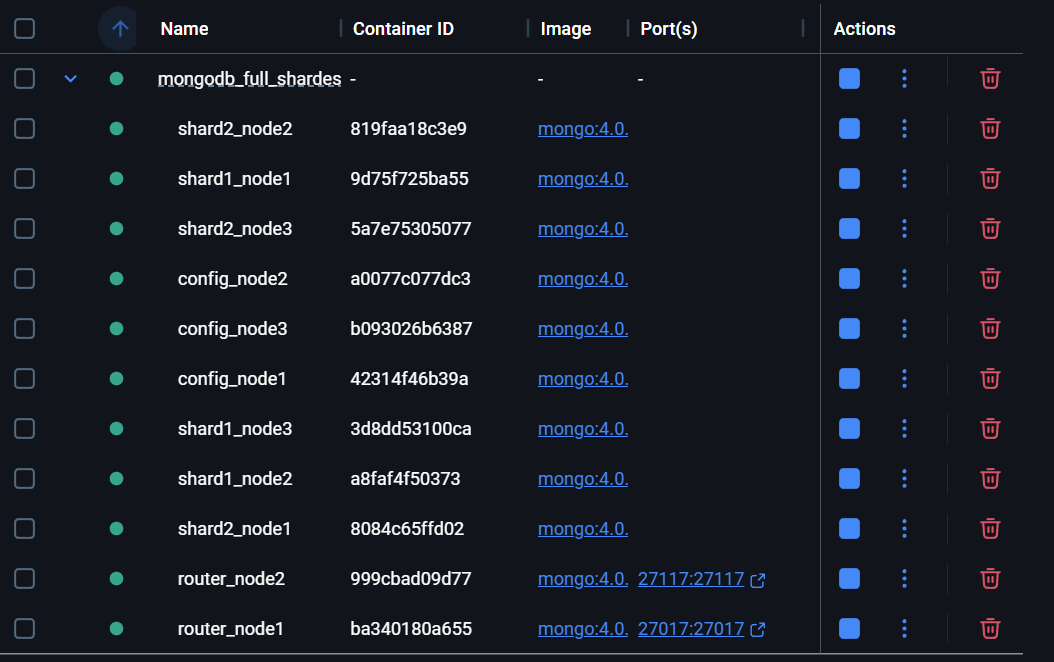


在Router节点的mongosh中运行配置文件，完成Router节点的配置

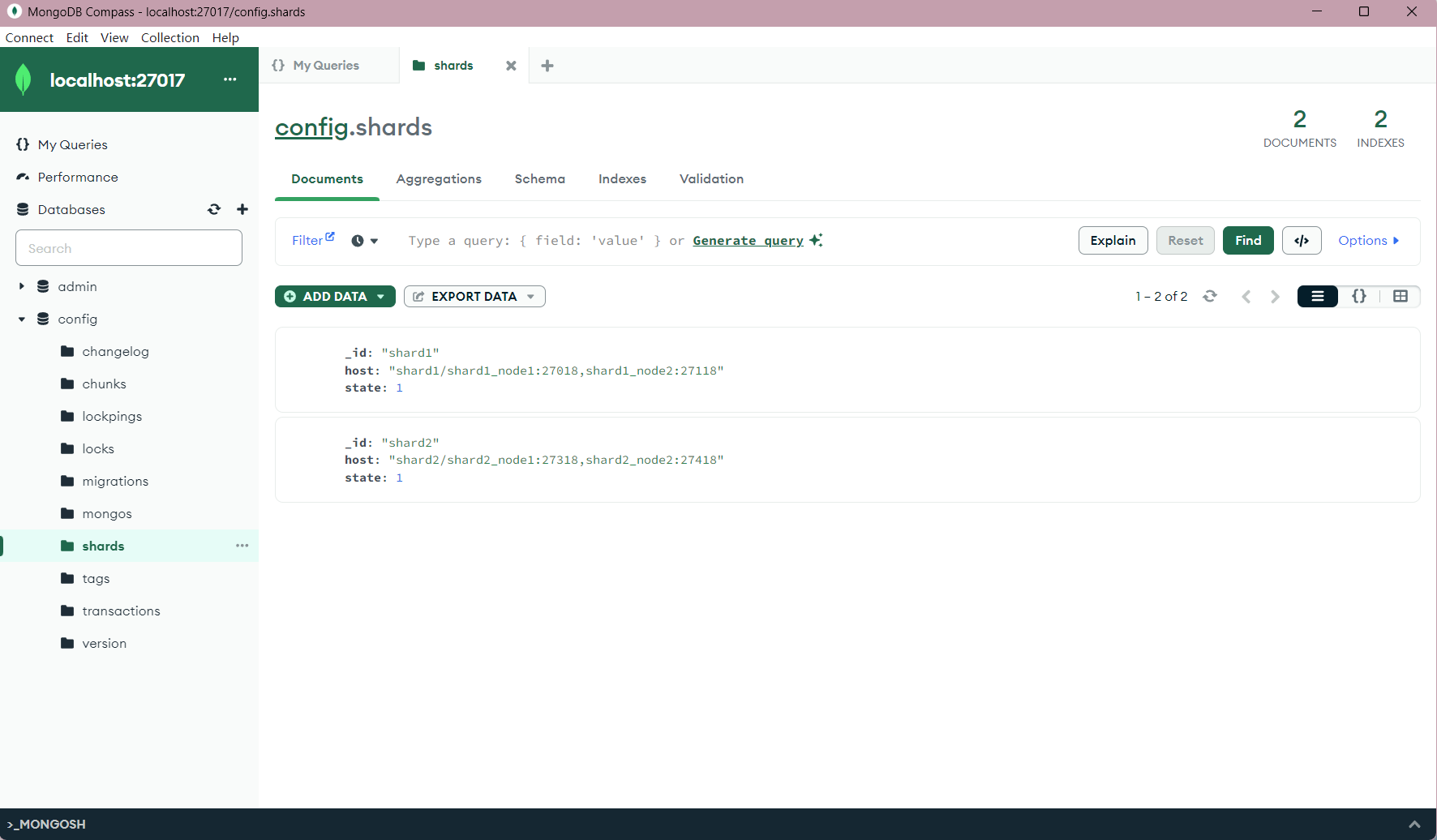


查看数据库各节点状态

完成配置后，在Docker Desktop中查看各节点容器运行状态如下所示



使用MongoDB Compass连接集群如图所示



能够成功连接并查找数据，至此集群搭建已完成

任务二

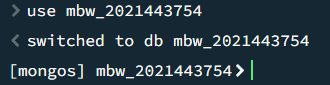
1. 创建数据库

在MongoDB Compass中创建数据库mbw\_2021443754

运行命令

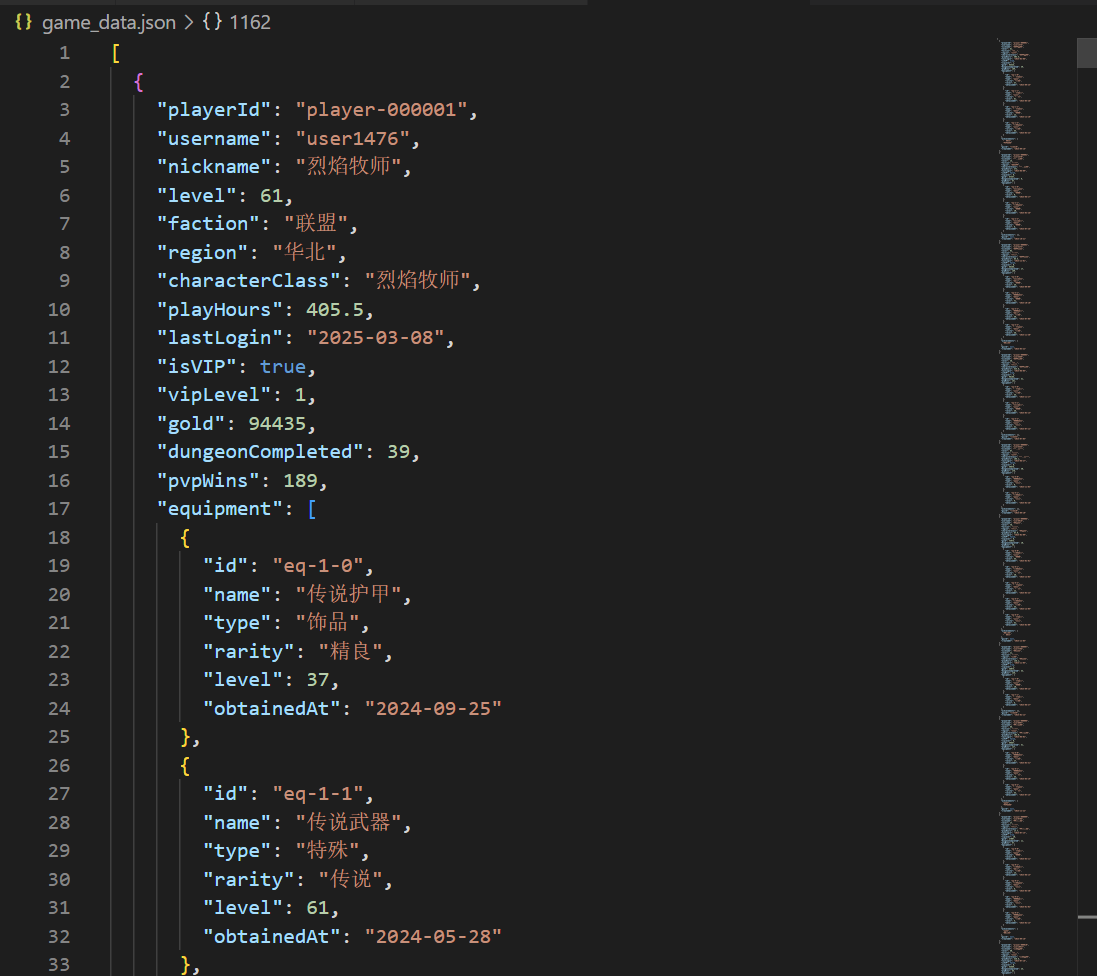


数据库创建结果如下



1. 浏览数据创建片键

打开数据文件game\_data.json，查看文件内容如图所示



为了确保数据的均匀分布，通常选择一个具有高基数（即值的多样性高）的字段作为片键。在game\_data 数据中，playerId字段是一个很好的选择，因为它是一个唯一标识符，能够很好地分散数据。

为数据库mbw\_2021443754启用数据分片功能

运行命令



查看运行结果



1. 导入数据
2. 查看数据分布情况