PartA

1. Preparing for the Environment

Create the redis-culster directory:

Prepare the `docker-compose.yml` file

Code block

```yaml
services:
  redis-7001:
    image: redis:7
    container_name: redis-7001
    command: redis-server /usr/local/etc/redis/redis.conf
    ports:
      - "7001:7001"
      - "17001:17001"
    volumes:
      - ./7001/redis.conf:/usr/local/etc/redis/redis.conf
      - ./7001:/data
    networks:
      - redis-net

  redis-7002:
    image: redis:7
    container_name: redis-7002
    command: redis-server /usr/local/etc/redis/redis.conf
    ports:
      - "7002:7002"
      - "17002:17002"
    volumes:
      - ./7002/redis.conf:/usr/local/etc/redis/redis.conf
      - ./7002:/data
    networks:
      - redis-net

  redis-7003:
    image: redis:7
    container_name: redis-7003
```

```
    command: redis-server /usr/local/etc/redis/redis.conf
    ports:
      - "7003:7003"
      - "17003:17003"
    volumes:
      - ./7003/redis.conf:/usr/local/etc/redis/redis.conf
      - ./7003:/data
    networks:
      - redis-net

  redis-7004:
    image: redis:7
    container_name: redis-7004
    command: redis-server /usr/local/etc/redis/redis.conf
    ports:
      - "7004:7004"
      - "17004:17004"
    volumes:
      - ./7004/redis.conf:/usr/local/etc/redis/redis.conf
      - ./7004:/data
    networks:
      - redis-net

  redis-7005:
    image: redis:7
    container_name: redis-7005
    command: redis-server /usr/local/etc/redis/redis.conf
    ports:
      - "7005:7005"
      - "17005:17005"
    volumes:
      - ./7005/redis.conf:/usr/local/etc/redis/redis.conf
      - ./7005:/data
    networks:
      - redis-net

  redis-7006:
    image: redis:7
    container_name: redis-7006
    command: redis-server /usr/local/etc/redis/redis.conf
    ports:
      - "7006:7006"
      - "17006:17006"
    volumes:
      - ./7006/redis.conf:/usr/local/etc/redis/redis.conf
      - ./7006:/data
    networks:
```

```
      - redis-net

networks:
  redis-net:
    driver: bridge
```

Prepare the `setup_redis_cluster.sh` ,For each node, create a `redis.conf` :

Code block

```
#!/bin/bash

# List of Redis ports for cluster nodes
ports=(7001 7002 7003 7004 7005 7006)

# Loop through each port and create configuration
for port in "${ports[@]}"; do
  # Create folder for this node
  mkdir -p ${port}

  # Write redis.conf file
  cat > ${port}/redis.conf <<EOF
port ${port}
bind 0.0.0.0
cluster-enabled yes
cluster-config-file nodes.conf
cluster-node-timeout 5000
cluster-announce-ip redis-${port}
cluster-announce-port ${port}
cluster-announce-bus-port 1${port}
appendonly yes
protected-mode no
EOF

  echo "Created configuration for node ${port}"
done

echo "All Redis configurations have been created."
```

2. Create Redis Node Directories

Execute the `setup_redis_cluster.sh`

We will use 6 nodes (3 masters, 3 replicas):

```
pen_warren@wus-MacBook-Pro redis-culster % ./setup_redis_cluster.sh
Created configuration for node 7001
Created configuration for node 7002
Created configuration for node 7003
Created configuration for node 7004
Created configuration for node 7005
Created configuration for node 7006
All Redis configurations have been created.
pen_warren@wus-MacBook-Pro redis-culster % ls
7001                  7003                  7005                  docker-compose.yml     setup_redis_cluster.sh
7002                  7004                  7006                  part_a.md
pen_warren@wus-MacBook-Pro redis-culster %
```

to execute the `docker-compose up -d`

```
pen_warren@wus-MacBook-Pro redis-culster % docker-compose up -d
[+] Running 7/7
 ✔ Network redis-culster_redis-net  Created
 ✔ Container redis-7005             Started
 ✔ Container redis-7001             Started
 ✔ Container redis-7003             Started
 ✔ Container redis-7004             Started
 ✔ Container redis-7002             Started
 ✔ Container redis-7006             Started
pen_warren@wus-MacBook-Pro redis-culster %
```

The result:

```
pen_warren@wus-MacBook-Pro redis-culster % docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS                                                                                        NAMES
477033ebf11c   redis:7   "docker-entrypoint.s…"   2 minutes ago   Up 2 minutes   0.0.0.0:7005->7005/tcp, [::]:7005->7005/tcp, 0.0.0.0:17005->17005/tcp, [::]:17005->17005/tcp   redis-7005
bfea557e0e8b   redis:7   "docker-entrypoint.s…"   2 minutes ago   Up 2 minutes   0.0.0.0:7003->7003/tcp, [::]:7003->7003/tcp, 0.0.0.0:17003->17003/tcp, [::]:17003->17003/tcp   redis-7003
6b1454c1ae90   redis:7   "docker-entrypoint.s…"   2 minutes ago   Up 2 minutes   0.0.0.0:7002->7002/tcp, [::]:7002->7002/tcp, 0.0.0.0:17002->17002/tcp, [::]:17002->17002/tcp   redis-7002
dd98796b847c   redis:7   "docker-entrypoint.s…"   2 minutes ago   Up 2 minutes   0.0.0.0:7006->7006/tcp, [::]:7006->7006/tcp, 0.0.0.0:17006->17006/tcp, [::]:17006->17006/tcp   redis-7006
7788f91ef840   redis:7   "docker-entrypoint.s…"   2 minutes ago   Up 2 minutes   0.0.0.0:7004->7004/tcp, [::]:7004->7004/tcp, 0.0.0.0:17004->17004/tcp, [::]:17004->17004/tcp   redis-7004
6a797bda4a23   redis:7   "docker-entrypoint.s…"   2 minutes ago   Up 2 minutes   0.0.0.0:7001->7001/tcp, [::]:7001->7001/tcp, 0.0.0.0:17001->17001/tcp, [::]:17001->17001/tcp   redis-7001
pen_warren@wus-MacBook-Pro redis-culster %
```

3. Create the Cluster

Code block

```
docker-compose exec redis-7001 redis-cli --cluster create \
redis-7001:7001 redis-7002:7002 redis-7003:7003 \
redis-7004:7004 redis-7005:7005 redis-7006:7006 \
--cluster-replicas 1
```

The excution result:

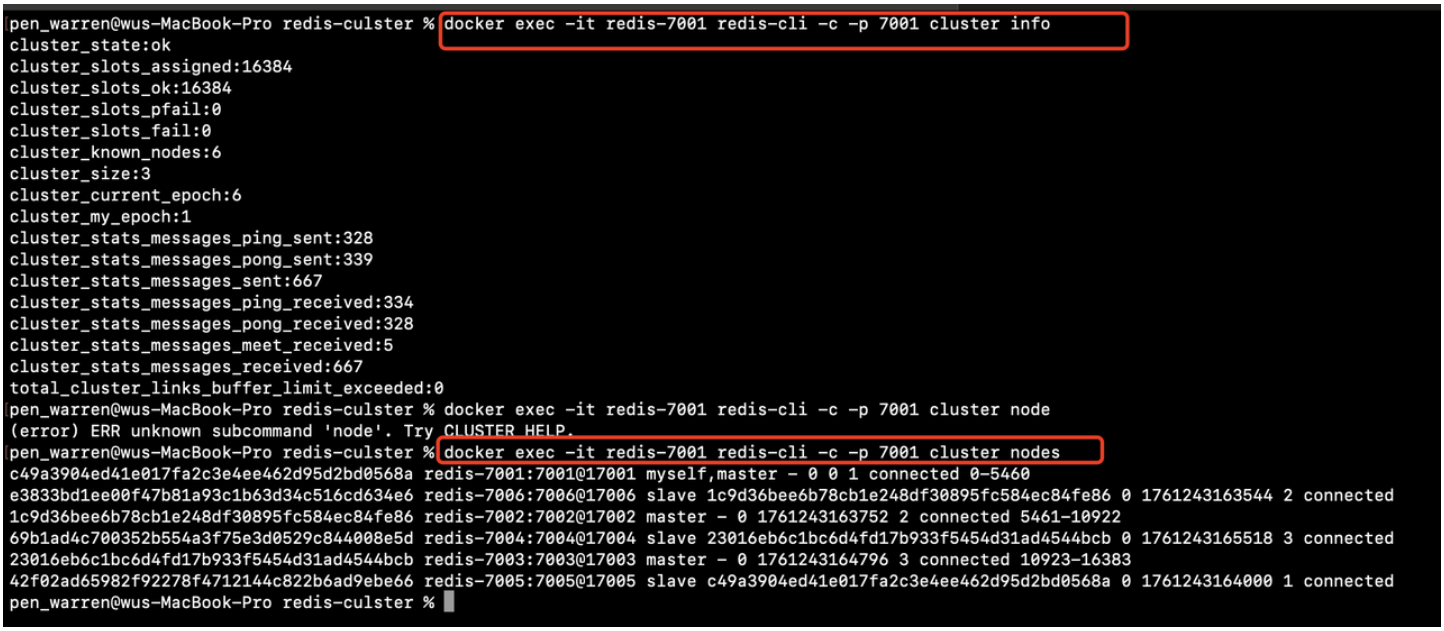
```
pen_warren@wus-MacBook-Pro redis-culster % docker-compose exec redis-7001 redis-cli --cluster create \
redis-7001:7001 redis-7002:7002 redis-7003:7003 \
redis-7004:7004 redis-7005:7005 redis-7006:7006 \
--cluster-replicas 1
>>> Performing hash slots allocation on 6 nodes...
Master[0] -> Slots 0 - 5460
Master[1] -> Slots 5461 - 10922
Master[2] -> Slots 10923 - 16383
Adding replica redis-7005:7005 to redis-7001:7001
Adding replica redis-7006:7006 to redis-7002:7002
Adding replica redis-7004:7004 to redis-7003:7003
M: c49a3904ed41e017fa2c3e4ee462d95d2bd0568a redis-7001:7001
   slots:[0-5460] (5461 slots) master
M: 1c9d36bee6b78cb1e248df30895fc584ec84fe86 redis-7002:7002
   slots:[5461-10922] (5462 slots) master
M: 23016eb6c1bc6d4fd17b933f5454d31ad4544bcb redis-7003:7003
   slots:[10923-16383] (5461 slots) master
S: 69b1ad4c700352b554a3f75e3d0529c844008e5d redis-7004:7004
   replicates 23016eb6c1bc6d4fd17b933f5454d31ad4544bcb
S: 42f02ad65982f92278f4712144c822b6ad9ebe66 redis-7005:7005
   replicates c49a3904ed41e017fa2c3e4ee462d95d2bd0568a
S: e3833bd1ee00f47b81a93c1b63d34c516cd634e6 redis-7006:7006
   replicates 1c9d36bee6b78cb1e248df30895fc584ec84fe86
Can I set the above configuration? (type 'yes' to accept): yes
>>> Nodes configuration updated
>>> Assign a different config epoch to each node
>>> Sending CLUSTER MEET messages to join the cluster
Waiting for the cluster to join

>>> Performing Cluster Check (using node redis-7001:7001)
M: c49a3904ed41e017fa2c3e4ee462d95d2bd0568a redis-7001:7001
   slots:[0-5460] (5461 slots) master
   1 additional replica(s)
S: e3833bd1ee00f47b81a93c1b63d34c516cd634e6 redis-7006:7006
   slots: (0 slots) slave
   replicates 1c9d36bee6b78cb1e248df30895fc584ec84fe86
M: 1c9d36bee6b78cb1e248df30895fc584ec84fe86 redis-7002:7002
   slots:[5461-10922] (5462 slots) master
   1 additional replica(s)
S: 69b1ad4c700352b554a3f75e3d0529c844008e5d redis-7004:7004
   slots: (0 slots) slave
   replicates 23016eb6c1bc6d4fd17b933f5454d31ad4544bcb
M: 23016eb6c1bc6d4fd17b933f5454d31ad4544bcb redis-7003:7003
   slots:[10923-16383] (5461 slots) master
   1 additional replica(s)
S: 42f02ad65982f92278f4712144c822b6ad9ebe66 redis-7005:7005
   slots: (0 slots) slave
   replicates c49a3904ed41e017fa2c3e4ee462d95d2bd0568a
[OK] All nodes agree about slots configuration.
```

4. Verify Cluster

Code block

```
docker exec -it redis-7001 redis-cli -c -p 7001 cluster info
docker exec -it redis-7001 redis-cli -c -p 7001 cluster nodes
```



5. Insert data into cluster

Design a UserProfile Entity

Field	Type	Description
user_id	string	Unique ID of the user
username	string	User’ s display name
email	string	User email address
last_login_time	string	Timestamp of last login

Example key-value in Redis:

Code block

```
Key: user:1
Value:
{"user_id":"1","username":"pengwu","email":"pengwu@example.com","last_login_tim
e":"2025-10-23 10:00:00"}
```

Connect to a master node and insert data:

Code block

```
docker exec -it redis-7001 redis-cli -c -p 7001
```

The execution results:

```
pen_warren@wus-MacBook-Pro redis-culster % docker exec -it redis-7001 redis-cli -c -p 7001
127.0.0.1:7001>
```

Code block

```
SET user:1 '{"user_id":"1","username":"pengwu","email":"pengwu@example.com","last_login_time":"2025-10-23 10:00:00"}'
SET user:2 '{"user_id":"2","username":"alice","email":"alice@example.com","last_login_time":"2025-10-23 10:05:00"}'
SET user:3 '{"user_id":"3","username":"bob","email":"bob@example.com","last_login_time":"2025-10-23 10:10:00"}'
```

Inter data:

```
pen_warren@wus-MacBook-Pro redis-culster % docker exec -it redis-7001 redis-cli -c -p 7001
127.0.0.1:7001> SET user:1 '{"user_id":"1","username":"pengwu","email":"pengwu@example.com","last_login_time":"2025-10-23 10:00:00"}'
-> Redirected to slot [10778] located at redis-7002:7002
OK
redis-7002:7002> SET user:2 '{"user_id":"2","username":"alice","email":"alice@example.com","last_login_time":"2025-10-23 10:05:00"}'
OK
redis-7002:7002> SET user:3 '{"user_id":"3","username":"bob","email":"bob@example.com","last_login_time":"2025-10-23 10:10:00"}'
-> Redirected to slot [2648] located at redis-7001:7001
OK
redis-7001:7001>
```

Check data insertion:

```
OK
redis-7001:7001> GET user:1
-> Redirected to slot [10778] located at redis-7002:7002
"{\"user_id\":\"1\",\"username\":\"pengwu\",\"email\":\"pengwu@example.com\",\"last_login_time\":\"2025-10-23 10:00:00\"}"
redis-7002:7002> GET user:2
"{\"user_id\":\"2\",\"username\":\"alice\",\"email\":\"alice@example.com\",\"last_login_time\":\"2025-10-23 10:05:00\"}"
redis-7002:7002> GET user:3
-> Redirected to slot [2648] located at redis-7001:7001
"{\"user_id\":\"3\",\"username\":\"bob\",\"email\":\"bob@example.com\",\"last_login_time\":\"2025-10-23 10:10:00\"}"
redis-7001:7001>
```