Redis Backup and Recovery Performance Testing Report

# Objective

The objective of this test was to measure the performance of Redis backup and recovery operations using a dataset representative of a production workload. The test focused on export (backup) and import (restore) durations to assess their efficiency and reliability.

# Test Environment

- Redis Configuration: 2 shards  
- Total Dataset Size: 5 GB (approximately 2.5 GB per shard)  
- Environment Type: [Specify if it was cloud, on-premises, or containerized]  
- Redis Version: [Insert version here, e.g., Redis 7.2]  
- Export Method: [e.g., RDB snapshot, `redis-cli --rdb`, `bgsave`, or custom script]  
- Import Method: [e.g., Loading RDB file, replication-based restore, or custom process]  
- Hardware Specs:  
 - CPU: [Insert details]  
 - RAM: [Insert details]  
 - Disk Type: [e.g., SSD/NVMe]  
 - Network Bandwidth: [Insert if applicable]

# Results

## Export (Backup)

- Total Time: 2 minutes and 12 seconds  
- Notes:  
 - Export was consistent across multiple runs.  
 - No significant CPU or memory spikes observed.  
 - Disk I/O remained within acceptable limits during the operation.

## Import (Restore)

- Total Time: 1 minute and 8 seconds  
- Notes:  
 - Import was successful without data integrity issues.  
 - Redis nodes resumed normal operations immediately after restore.  
 - Initial performance lag post-import was negligible (< 5 seconds).

# Conclusion

The Redis backup and recovery process proved to be efficient and reliable for a 5 GB dataset spread across two shards. Export and import durations were well within acceptable timeframes for operational use, making this approach suitable for scheduled backups and disaster recovery scenarios.

# Recommendations

- Automation: Integrate the backup and restore process into CI/CD or scheduled jobs for improved resilience.  
- Monitoring: Ensure metrics are captured during backup/restore for future tuning.  
- Scale Testing: Perform further testing with larger datasets and more shards to assess scalability.