Task 4: Docker Disaster Recovery Report Date: 14 Jan 2026 Prepared by: Janak Singh

1. Objective

Design and implement a backup & restore system for containerized applications that supports automated failover and disaster recovery. The system ensures application availability even in case of container or node failures.

2. Application Setup

- 3-tier application deployed in Docker:
  1. Database: PostgreSQL
  2. Backend: Node.js
  3. Frontend: Nginx/HTML
- Docker Compose used for orchestration
- Persistent volumes for database
- Network: docker-dr-assignment_dr-network

3. Backup & Restore Procedure

1. backup.sh - Automated backup script:
   - Backs up PostgreSQL database using pg_dump
   - Backs up Docker volumes using tar + gzip
   - Stores backups in ./backups/volumes with timestamp
2. restore.sh - Disaster Recovery script:
   - Stops containers
   - Removes old DB volume
   - Restores latest DB volume snapshot
   - Restores latest SQL dump
   - Starts backend and frontend containers
3.

Failover Simulation & Validation

Scenario 1: Container Crash

- Command: docker kill docker-dr-backend
- Result: Backend DOWN detected in monitor.log
- Restore: ./restore.sh executed successfully
- Post-recovery: Backend UP, Frontend UP, Database UP

Scenario 2: Node Failure - Command: sudo systemctl stop docker - Command: minikube stop - Result: All containers stopped - Restore: sudo systemctl start docker; minikube start; ./restore.sh - Post-recovery: All services running

Scenario 3: Data Corruption - Delete DB files manually or remove volume - Restore executed: ./restore.sh - Post-recovery: Database restored, application functional

5. Monitoring & Alerts

- Scripts: health_check.sh, alert_check.sh
- monitor.log: Tracks service health every minute
- alert.log: Records alerts if any service is DOWN
- Example logs:

===== Health Check at Wed Jan 14 12:23:23 UTC 2026 ===== Backend DOWN Frontend DOWN Database DOWN

[ALERT] Service DOWN detected at Wed Jan 14 12:23:54 UTC 2026

6. Logs & Screenshots

janaksingh@DESKTOP-LAI1NV8:~/docker-dr-assignment$ cat monitoring/monitor.log ===== Health Check at Wed Jan 14 12:23:23 UTC 2026 ===== Backend DOWN Frontend DOWN Database DOWN

===== Health Check at Wed Jan 14 12:23:54 UTC 2026 ===== Backend DOWN Frontend DOWN Database DOWN

===== Health Check at Wed Jan 14 12:29:27 UTC 2026 ===== Backend DOWN Frontend DOWN Database DOWN

===== Health Check at Wed Jan 14 12:29:47 UTC 2026 ===== Backend DOWN Frontend DOWN Database DOWN

===== Health Check at Wed Jan 14 12:39:43 UTC 2026 ===== Backend OK Frontend OK Database OK

===== Health Check at Wed Jan 14 12:41:43 UTC 2026 ===== Backend DOWN Frontend OK Database OK

===== Health Check at Wed Jan 14 12:41:51 UTC 2026 ===== Backend DOWN Frontend OK Database OK

===== Health Check at Wed Jan 14 12:42:58 UTC 2026 ===== Backend DOWN Frontend OK Database OK

===== Health Check at Wed Jan 14 12:53:00 UTC 2026 ===== Backend OK Frontend OK Database OK

===== Health Check at Wed Jan 14 12:53:17 UTC 2026 ===== Backend OK Frontend OK Database OK

janaksingh@DESKTOP-LAI1NV8:~/docker-dr-assignment$ cat monitoring/alert.log [ALERT] Service DOWN detected at Wed Jan 14 12:23:54 UTC 2026

7. Conclusion

- Disaster recovery workflow is fully functional.
- Automated backup and restore with minimal downtime verified.
- Monitoring and alerting system correctly detects failures.
- Application fully restored after container or node failure.