Task: Set Up a Local Kubernetes Cluster and Automate Disaster Recovery

Objective:

Create a local Kubernetes environment using kubeadm, deploy a sample app, and implement an automated disaster recovery (DR) pipeline with dynamic backup/restore capabilities using GitLab CI/CD. Demo the solution once complete.

Requirements:

1. **Local Cluster Setup:**
    a. Use kubeadm to set up a single-node Kubernetes cluster on your machine.
    b. Deploy a stateful app (e.g., WordPress with MySQL, using a PersistentVolume).
2. **Backup Automation:**
    a. Install Velero to manage backups of the app's configuration and data.
    b. Configure dynamic backups: Enable selective backup/restore for specific namespaces or resources (e.g., by labels or names).
    c. Schedule automated backups every 24 hours for critical workloads (e.g., the app's namespace) to a local MinIO instance (set up as an S3-compatible store).
    d. After each backup, delete the previous backup to optimize storage space.
3. **Recovery Pipeline:**
    a. Create a GitLab CI/CD pipeline (new repo or local Git setup):
        i. Add a backup job to trigger manually or dynamically (e.g., for a specific namespace).
        ii. Add a restore job to recover selected resources from a backup.
    b. Test the pipeline by simulating a failure (e.g., delete the app's namespace) and restoring it dynamically.
4. **Demo Prep:**
    a. Simulate a disaster (e.g., delete the app or a specific resource) and recover it live using dynamic restore.
    b. Target recovery within 15 minutes.

Deliverables:

- A running kubeadm cluster with the DR solution.

- Git repo with CI/CD pipeline code.
- A short README explaining:
    - Cluster setup with kubeadm.
    - How to trigger dynamic backups/restores (e.g., for a specific namespace/resource).
    - Steps to test the 24-hour backup and deletion process.
- A 5-minute demo showing setup, dynamic backup, failure, and recovery.

Timeline:

Demo on 09-04-2024(Wednesday) EOD.