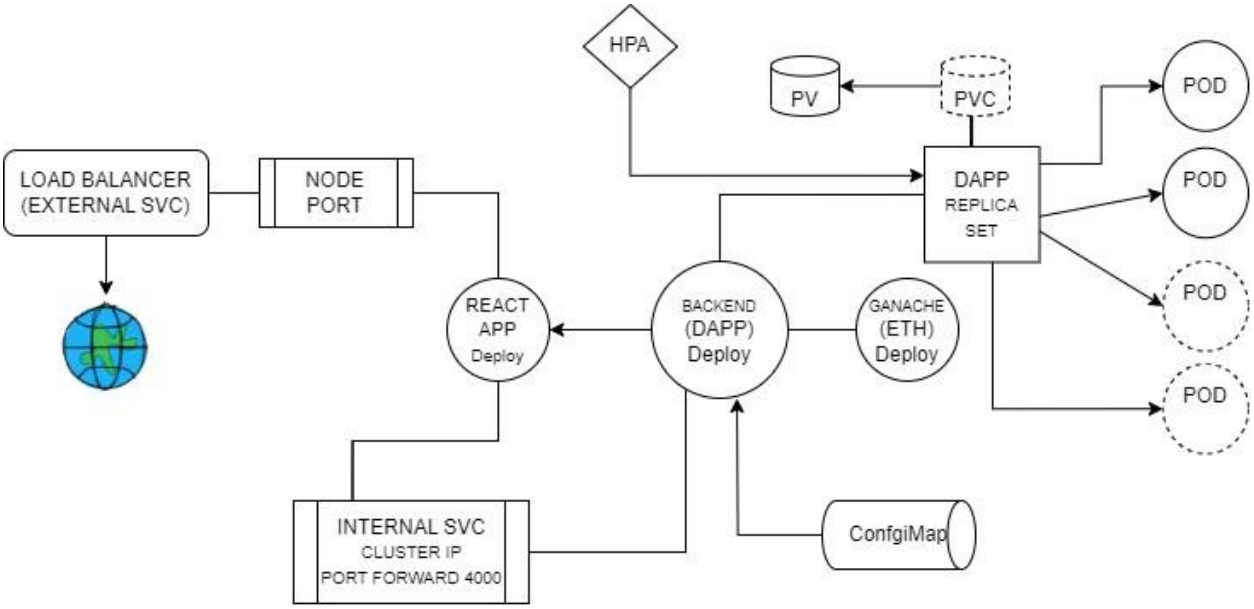
Kubernetes Deployment Architecture Design Document for Docker-Ethereum Application

ARCHITECTURE DIAGRAM:



1. dapp (Backend) Deployment:

- **Stateful Deployment:**
    - Reason: The backend (dapp) is deployed as a StatefulSet to maintain state across pod replicas. The dapp backend uses shared persistent storage to store the deployed contract addresses. This shared storage allows multiple pods to access and update the deployed contract addresses file, facilitating seamless communication between pods.
- **Horizontal Pod Autoscaler (HPA):**
    - Reason: The HPA is configured to automatically scale the number of pod replicas based on resource utilization. This ensures optimal performance and responsiveness, allowing the application to handle varying workloads.
- **Persistent Storage:**
    - Reason: The backend deployment utilizes Persistent Volumes (PV), Persistent Volume Claims (PVC), and a Storage Class to provide persistent storage for the application's data.

- **ConfigMap Usage:**
    - Reason: The use of ConfigMap for storing configuration data, such as the GANACHE_URL_PORT variable, allows for easy configuration changes without modifying the application code.

2. React App Deployment:

- **Service Type: LoadBalancer and NodePort:**
    - Reason: The React app is exposed to external traffic using a LoadBalancer service type, which provides a stable external IP for accessing the application. Additionally, a NodePort service type is used for local access during development. This dual-service approach caters to both external and local access needs.
- **Simplicity in Deployment:**
    - Reason: The React app is deployed using a straightforward Deployment object. Since the frontend is stateless and doesn't require persistent storage a basic Deployment suffices for ensuring application availability.

3. Ganache App Deployment:

- **Simplicity in Deployment:**
    - Reason: Similar to the React app, the Ganache app is stateless and doesn't require persistent storage. Therefore, a basic Deployment object is used for simplicity and efficiency.