

The Kubernetes deployment for the Docker-Ethereum application consists of several parts, each with its own design choices:

1. Ethereum Server (eth-server):
 - Deployment: Stateless (each instance is independent).
 - Replicas: 2 (ensures high availability and load balancing).
 - Service Type: ClusterIP (internal connectivity, not accessed from outside).
2. Ganache Server (eth-ganache):
 - Deployment: Stateless (for scalability and independence).
 - Replicas: 2 (sufficient for development and testing).
 - Service Type: ClusterIP (internal component, no external access).
3. React Frontend (eth-react):
 - Deployment: Stateless (easy scaling and management).
 - Replicas: 1 (can be scaled based on requirements).
 - Service Type: LoadBalancer (exposed to external traffic for user access).
 - Horizontal Pod Autoscaler (HPA): Enabled (automatically adjusts replicas based on CPU utilization).
4. Permissions and RBAC:
 - Role and RoleBinding (pod-reader) for reading pods with specified permissions.
 - ClusterRole and ClusterRoleBinding (pod-reader-global) allowing global pod reading for a specific user reetika.

5. Secrets:

- ConfigMap (node-app-config) storing Ethereum server configuration (NODE_PORT set to "8080").

Summary: This Kubernetes setup aims for scalability, resilience, and organization. Stateless deployments of Ethereum and Ganache servers provide flexibility, while the React frontend efficiently handles external traffic. RBAC controls enhance security, and a ConfigMap for configuration adds flexibility. The architecture is designed for optimal resource usage and responsiveness.