Load Balancer with Service Turn-Off

Resources and microservices

Project statement

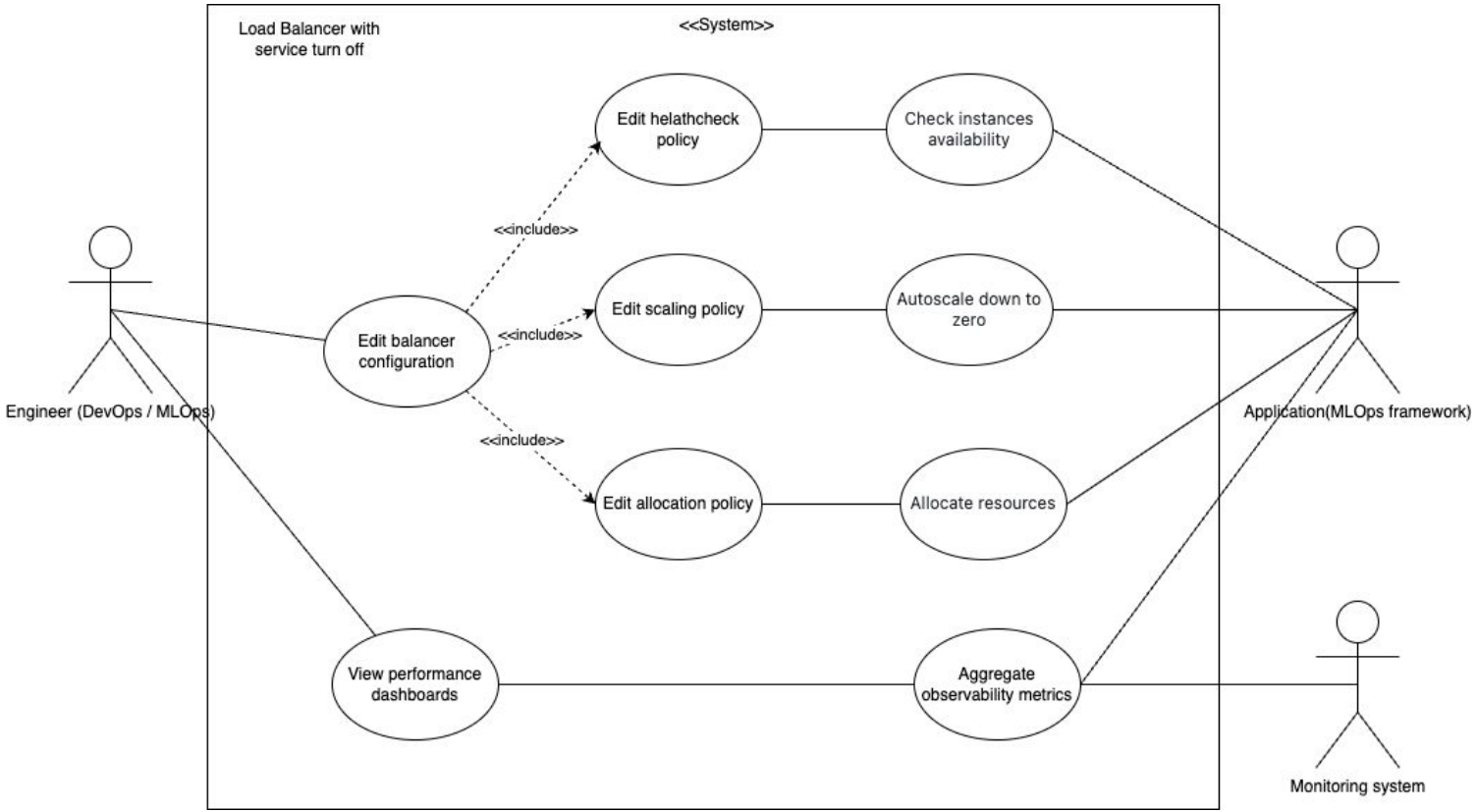
The project aims to enhance the functionality of a vanilla Kubernetes by enabling it to scale applications down to zero instances when they are not in use. This is relevant for ML applications that can be slow to start. The service will use event-driven automata to manage application scaling in response to real-time monitoring data and optimizing resource usage.

Team: Dmitry Kara, Daniil Mikulik, Ekaterina Karavayeva, Nikita Dumkin

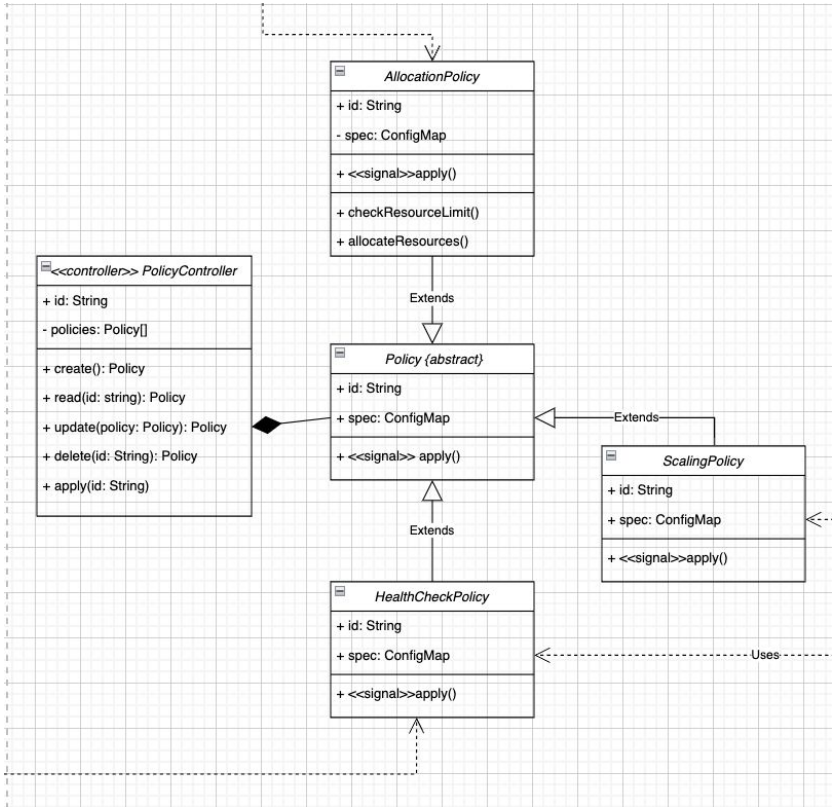
Project repo: https://github.com/dmitriykara/ads-tech-tornados-project

This report: https://docs.google.com/presentation/d/1IaeXMkLVIqLq7L1JCghJjV8ALhiMkqwj1jUcEOtdsyg/edit?usp=sharing

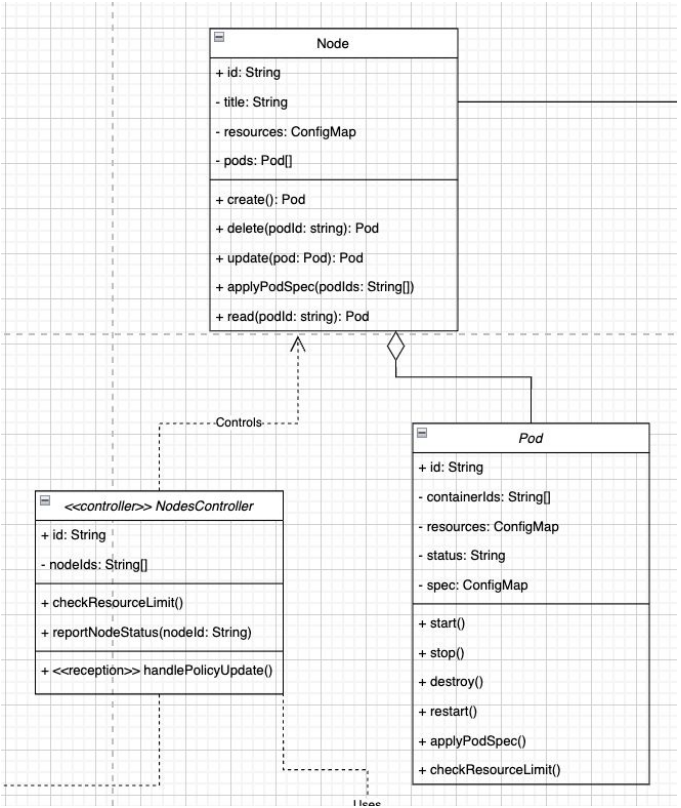
Use case diagram



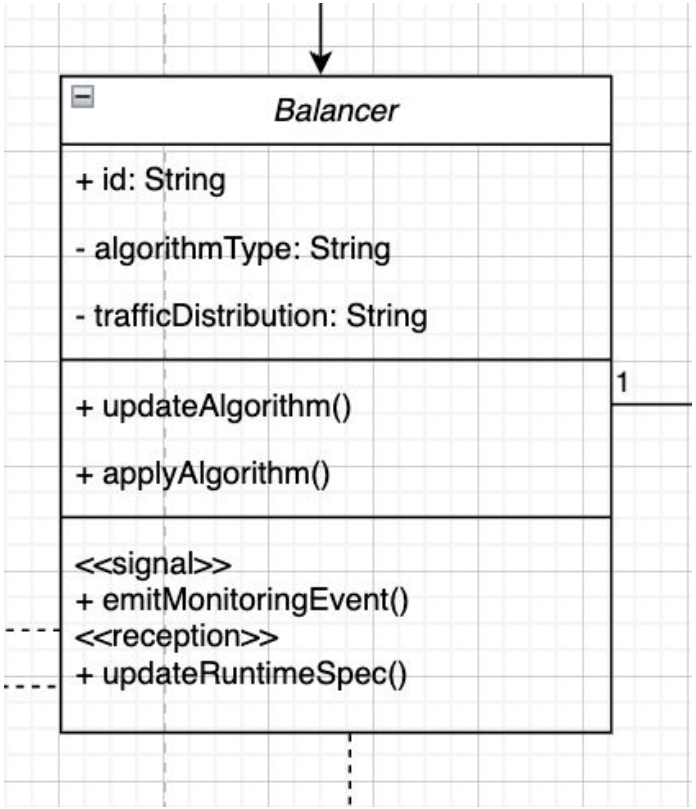
Detailed class diagram - Policies manager service



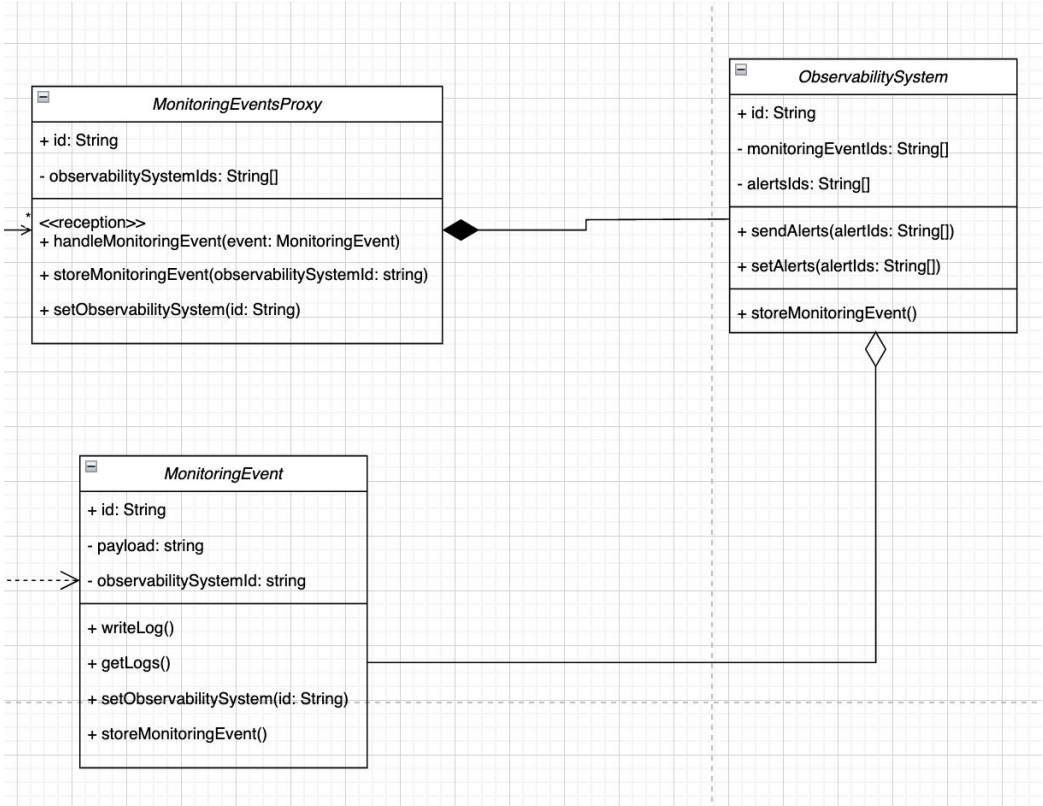
Detailed class diagram - Node management service



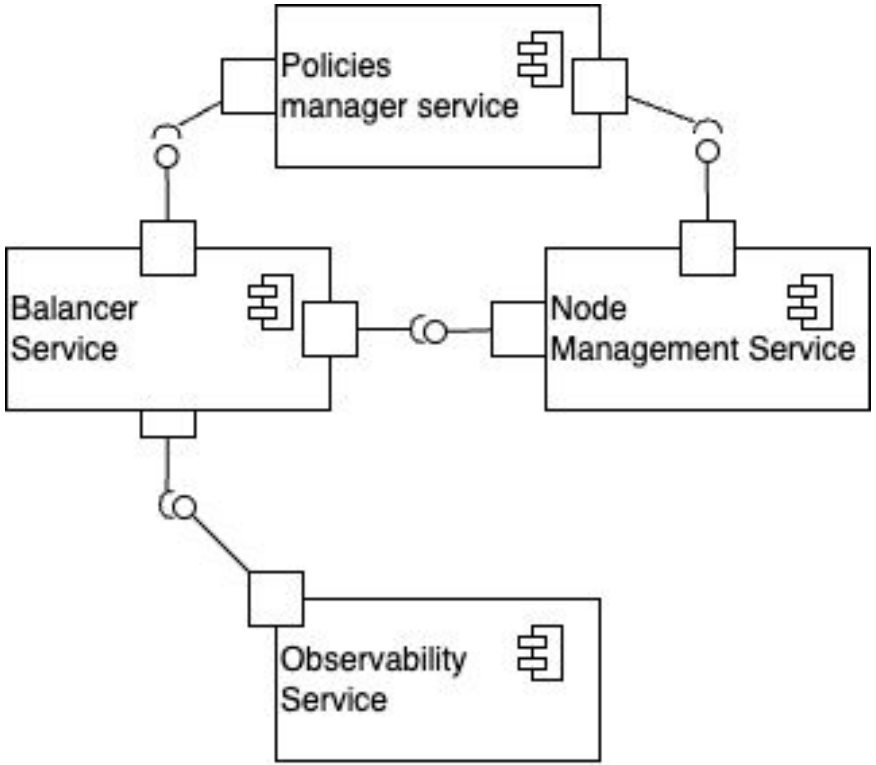
Detailed class diagram - Balancer service



Detailed class diagram - Observability service



Service diagram



Traceability

Microservice (component)	Resource (interface)	Domain classes in the microservice	Use case that use the microservice
Balancer Service	- Check Node & Pod Status - Send Metrics	- TrafficBalancer - LoadBalancerConfig	- Balance Traffic - Monitor Pod Health
Policies Manager Service	- Apply Policies to Nodes - Apply Policies to Balancer	- PolicyController - AllocationPolicy - ScalingPolicy	- Manage Policies - Apply Resource Allocation Policies
Node Management Service	- Send Node & Pod Data - Send Metrics	- Node - Pod - Kubelet	- Manage Node Lifecycle - Collect Node and Pod Metrics
Observability Service	- Receive Metrics - Store Node & Pod Data	- MetricsCollector - Logger	- Monitor Node and Pod Health - Track System Metrics