

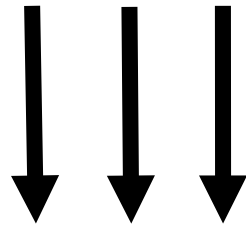
Университет ИТМО
Факультет Безопасности Информационных
Технологий

**Интеллектуальная система управления
версиями программного обеспечения в
Kubernetes**

Березина Анна Сергеевна

Проблема

- При выпуске новой версии иногда возникает ухудшение качества работы на каком-либо устройстве
- Причинами могут быть:
 - недостаток памяти
 - особые условия использования
 - несовместимость каких-либо компонентов и т.д.



- **Возрастает количество ошибок**
- **Возможна потеря пользовательских данных**

Цель проекта

Создать умную систему контроля качества сервиса, реализующую автоматический возврат к стабильной версии в случае ухудшения качества работы на конкретном устройстве

Задачи

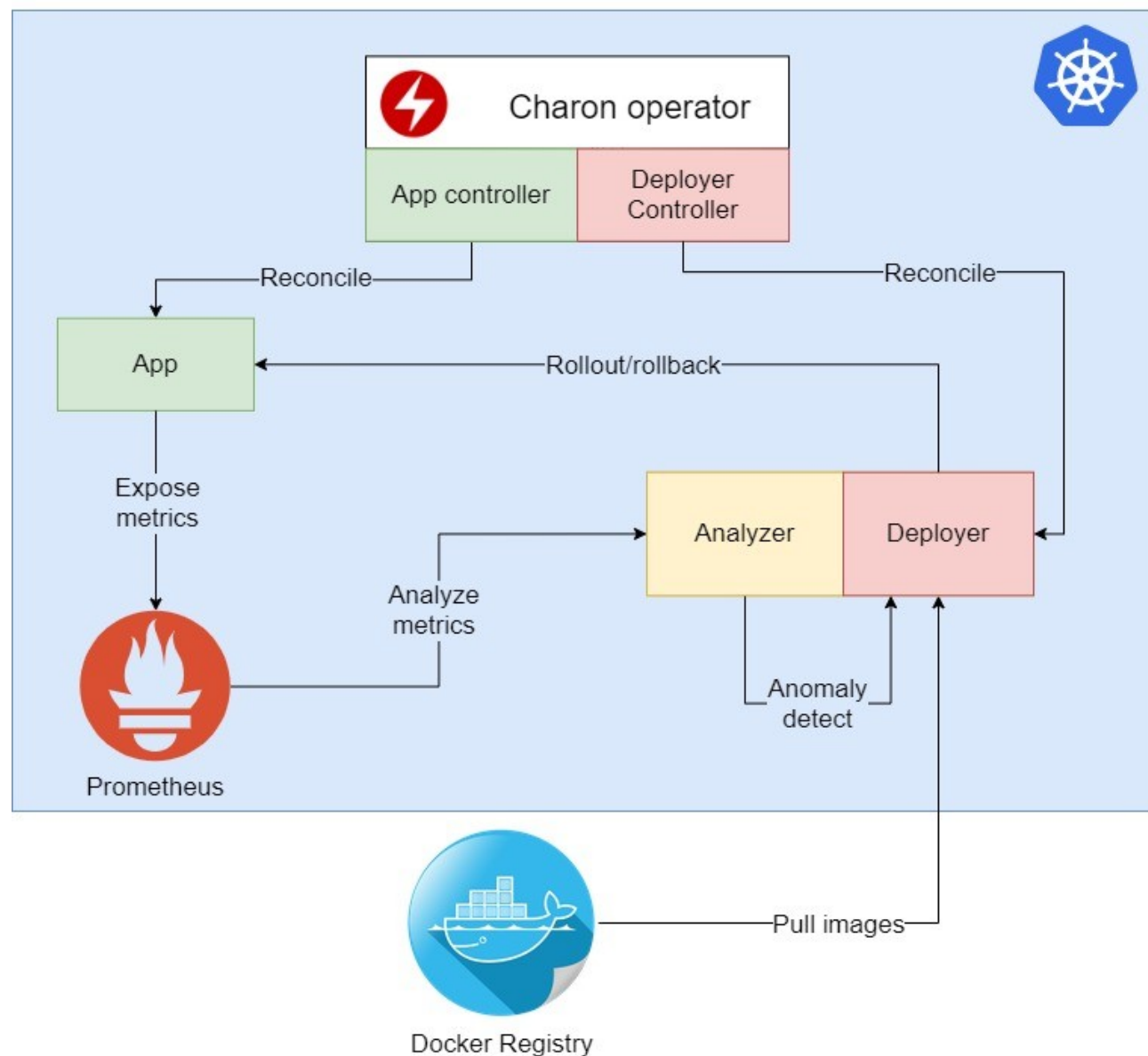
- Разработать модуль развертывания определенных версий сервисов
- Настроить получение уведомлений в случае загрузки новой версии
- Реализовать получение метрик сервисов с помощью Prometheus
- Реализовать анализ трендов изменения метрик и выявление аномалий

Анализ существующих решений

- **KubeRBS** (Kubernetes Rollback System) - система, реализующая возврат к предыдущей версии, если количество ошибок превышает некий заданный порог
- Работает только с метриками из Stackdriver и Datadog
- Определяет аномалии только по количеству ошибок

<https://github.com/doitintl/kuberbs>

Архитектура системы



Оператор

- Это ПО, расширяющее базовый функционал Kubernetes и позволяющее автоматизировать управление приложениями
- Следит за жизненным циклом тестового приложения и модуля, отвечающего за развертывание (deployer)

Deployer

- Модуль, отвечающий за развертывание необходимых версий тестового приложения в Kubernetes
- Получает уведомления от Docker Registry о том, что была загружена новая версия
- Получает уведомления от анализатора о необходимости отката к стабильной версии
- Отправляет на Kubernetes API Server PATCH-запрос на обновление конфигурации тестового сервиса

Анализатор

- Модуль, анализирующий тренды изменения метрик тестового приложения и обнаруживающий аномалии
- Библиотека Anomalyzer
 - U-критерий Манна — Уитни
 - Критерий согласия Колмогорова
 - Сравнение относительной разности между средними значениями

Kubernetes

```
root@charon-project:~/charon/deployer# kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
charon-curl-tester	1/1	Running	0	4d13h
charon-deployer	2/2	Running	1	21s
charon-operator-7888697667-78vkz	1/1	Running	0	4d
charon-registry	1/1	Running	0	20h
grafana-6d7cc69ffb-9t6xc	1/1	Running	0	4d19h
prometheus-deployment-5c4f4f5779-szbmw	1/1	Running	0	4d19h
test-app	1/1	Running	17	4d

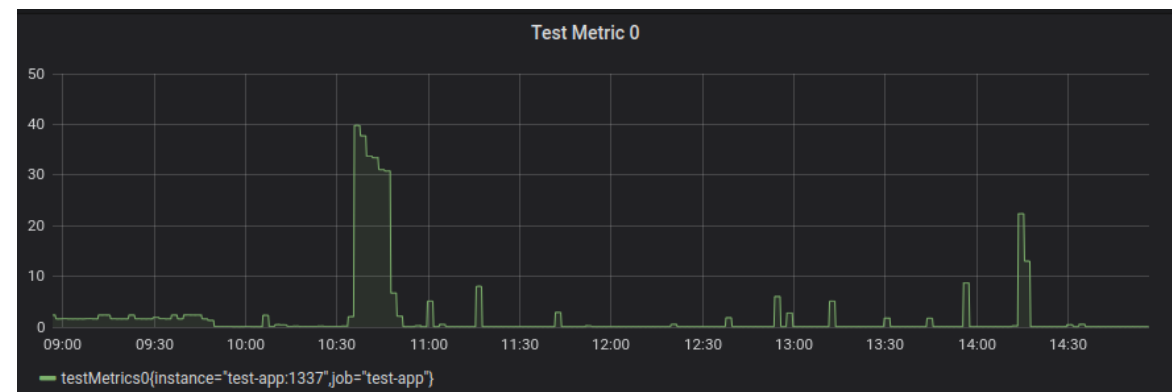
Deployer

```
[GIN] 2020/04/10 - 13:51:26 | 200 | 595.807µs | 192.168.184.119 | POST /rollout
Processing event 1 of 1
test-app:v1.1
[GIN] 2020/04/10 - 13:51:26 | 200 | 85.624939ms | 192.168.184.119 | POST /rollout
2020/04/10 14:00:52 Received a rollback request for image: test-app
[GIN] 2020/04/10 - 14:00:52 | 200 | 53.994035ms | 192.168.184.125 | POST /rollback
```

Анализатор

```
2020/04/10 13:55:53 Metric: testMetrics1{instance="test-app:1337",job="test-app"}; Probability: 0.997252
2020/04/10 13:55:53 Connected to Prometheus... Querying metrics...
2020/04/10 13:58:25 Metric: testMetrics2{instance="test-app:1337",job="test-app"}; Probability: 0.792211
2020/04/10 13:58:25 Connected to Prometheus... Querying metrics...
2020/04/10 14:00:52 Metric: testMetrics0{instance="test-app:1337",job="test-app"}; Probability: 0.922672
2020/04/10 14:00:52 ANOMALY! 0.922672
2020/04/10 14:00:52 Sent rollback request!
2020/04/10 14:00:52 Connected to Prometheus... Querying metrics...
2020/04/10 14:04:23 Metric: testMetrics1{instance="test-app:1337",job="test-app"}; Probability: 0.000000
2020/04/10 14:04:23 Connected to Prometheus... Querying metrics...
```

Визуализация метрик



Использованные инструменты

- Golang 1.14
- Kubernetes 1.17
- OperatorSDK
- Prometheus
- Grafana

Выводы

- В ходе работы над проектом был реализован рабочий прототип системы, осуществляющий все основные функции
- Выбранное решение показало свою актуальность для поставленной проблемы и целесообразность дальнейшего развития данного проекта

Дальнейшие перспективы

- Реализовать получение новых версий из публичных registry и Git
- Добавить возможность работы с более сложными приложениями
- Реализовать отправку уведомлений
- Развивать механизм обнаружения аномалий

Спасибо за внимание!

<https://github.com/f0m41h4u7/Charon>