



Проектная работа: MVP инфраструктурной платформы

Добро пожаловать в мир современной разработки! Эта презентация познакомит вас с нашей проектной работой по созданию MVP инфраструктурной платформы. Мы рассмотрим цели, требования и результаты, которые помогут вам закрепить знания и создать впечатляющее портфолио.

Цели проектной работы

1

Закрепление знаний

Практическое применение полученных знаний и навыков в области DevOps и облачной инфраструктуры.

2

Портфолио

Создание реального проекта для демонстрации работодателям, подтверждающего ваши компетенции.

3

Продолжение обучения

Глубокое погружение в тему, стимулирующее дальнейшее изучение и развитие в выбранной области.

5

Представление работы

Оptionальная возможность представить результаты на последнем вебинаре, получить обратную связь и признание.



Требования к реализации MVP платформы

Обязательный функционал:

- ***Kubernetes (managed или self-hosted) – основа для развертывания.***
- ***Мониторинг кластера и приложения с алертами и дашбордами.***
- ***Централизованное логирование для кластера и приложений.***
- ***CI/CD пайплайн (Gitlab CI, Github Actions и т.д.) для автоматизации развертывания.***

Дополнительно (для продвинутых):

- ***Инфраструктура для трейсинга.***
- ***Хранилище артефактов.***
- ***Хранилище секретов.***
- ***И многое другое по вашему выбору для расширения функционала.***

Требования к оформлению и доступу

1

Код в репозитории

Всё должно быть описано в виде кода: автоматизация создания кластера, развертывание сервисов, настройки мониторинга и CI/CD пайплайны.

2

Публичный доступ

Необходим публично доступный репозиторий с кодом, пайплайнами и ingress кластера (белый IP, домен, например, через nip.io).

3

Документация

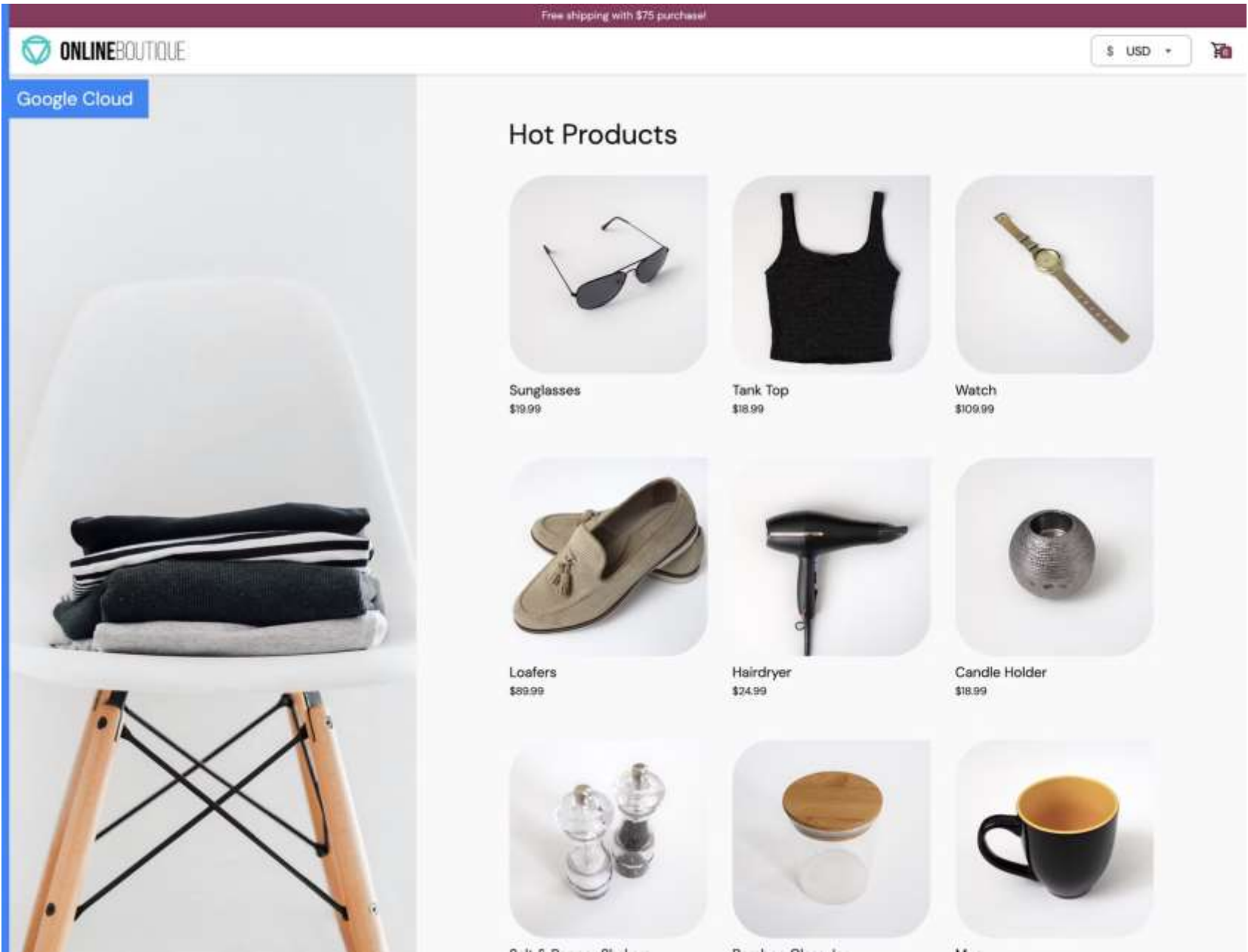
Подробный README с описанием решения, шагами развертывания, настройками и дополнительной информацией.



Выбор тестового приложения

Вариант 1: Наш пример

*Приложение **Online Boutique** by Google — отличный выбор для старта. Это микросервисное приложение, которое позволяет глубоко погрузиться в архитектуру и взаимодействие сервисов.*



Наша архитектура решения

Мы развернули самоуправляемый Kubernetes кластер (self-hosted) с высокой доступностью (HA Control Plane) и тремя рабочими узлами. Сетевая топология включает маршрутизацию через Windows Host и Router с NAT, обеспечивая внешний доступ к сервисам.

Основные компоненты платформы



ArgoCD для
GitOps

*Система
декларативно
го управления
Kubernetes,
обеспечивающ
ая
автоматичес
кую
синхронизаци
ю состояния
кластера с Git-
репозиторием
.*



Ingress
Controller
(nginx)

*Обрабатывае
т входящий
трафик и
маршрутизир
ует запросы к
соответству
ющим
микросервиса
м в кластере.*



Monitoring
Stack (Loki
+
Prometheus
+ Grafana)

*Комплексное
решение для
сбора метрик,
централизованного
логирования и
визуализации
данных о
состоянии
кластера и
приложений.*



Local Path
Provisioner

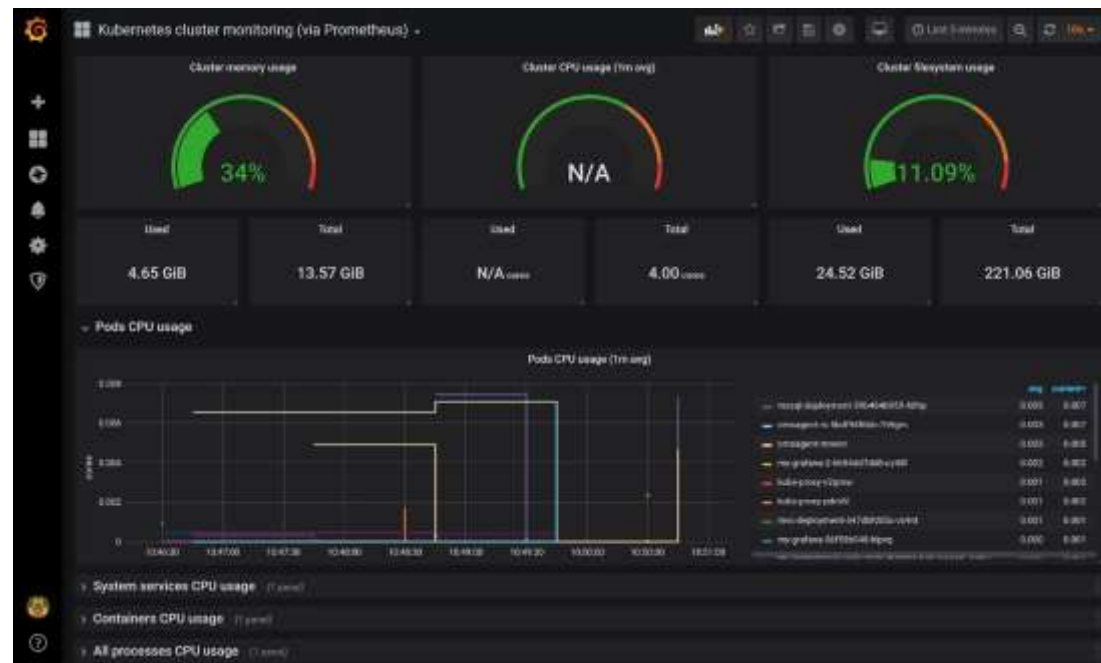
*Автоматичес
кое создание
хранилищ для
персистентн
ых данных,
что
упрощает
управление
хранением в
кластере.*

Мониторинг и логирование

Метрики

С помощью Prometheus собираются и анализируются ключевые метрики:

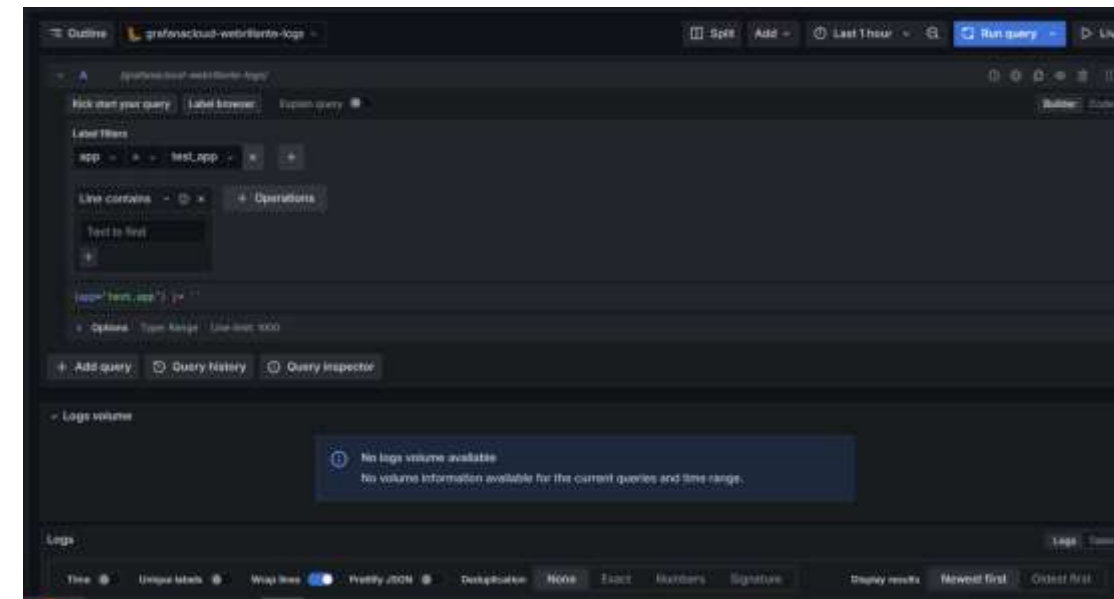
- *Использование CPU и памяти узлов.*
- *Пропускная способность сети и дисковые операции.*
- *Метрики Pod'ов (CPU, память, сетевые операции).*
- *Метрики приложения (запросы, задержки, ошибки) для быстрой диагностики.*



Логирование

Loki обеспечивает централизованный сбор и хранение логов всех контейнеров. Это позволяет легко:

- *Фильтровать логи по namespace, pod, контейнеру.*
- *Проводить глубокий анализ для выявления проблем.*
- *Использовать дашборды Grafana для визуализации логов (Loki Kubernetes Logs).*



Доступ к сервисам и безопасность

Сервис	URL	Описание
<i>Microservices Demo (HTTP)</i>	<i>http://kibakus.duckdns.org</i>	<i>Приложение Online Boutique</i>
<i>Microservices Demo (HTTPS)</i>	<i>https://kibakus.duckdns.org</i>	<i>С самоподписанным сертификатом</i>
<i>Grafana</i>	<i>http://kibakus.duckdns.org:32000</i>	<i>Мониторинг и дашборды</i>

Важное замечание по безопасности:

Данное решение предназначено для тестирования и обучения. Для production среды рекомендуются меры:

- *Использование Let's Encrypt с cert-manager.*
- *Настройка RBAC и NetworkPolicies.*
- *Внедрение Secret Management (Sealed Secrets, Vault).*



Ключевые выводы и полученные знания



Администрирование **Kubernetes**

Глубокое понимание развертывания HA кластера, управления узлами и сетевыми политиками.



Инфраструктура как код

Практический опыт с Ansible/Kubespray и GitOps подходом через ArgoCD.



Мониторинг и наблюдаемость

Уверенное использование Prometheus, Loki и Grafana для сбора, анализа и визуализации данных.



DevOps и автоматизация

Закрепление концепций CI/CD, управления конфигурациями и автоматической синхронизации состояния.

Этот проект стал ценным опытом, демонстрирующим практические навыки в построении современной инфраструктурной платформы.