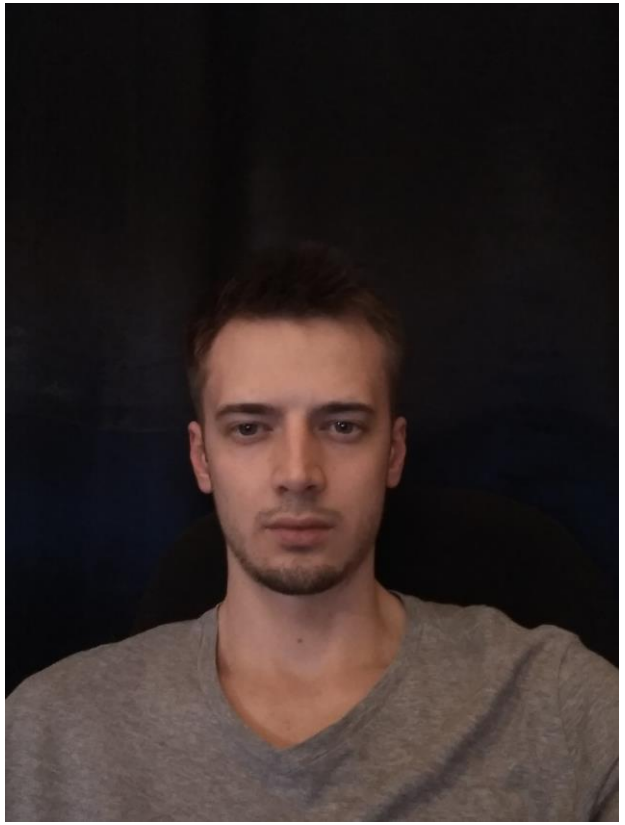
Контейнеры. от Docker до Kubernetes

Luxoft | training
A DXC Technology Company

INTRODUCTION



Никонов Герман

Contacts

gnikonov@it-platforma.ru

DevOps Engineer
Kubernetes/Openshift 2+ year experience

SEMINAR ROADMAP: OVERVIEW

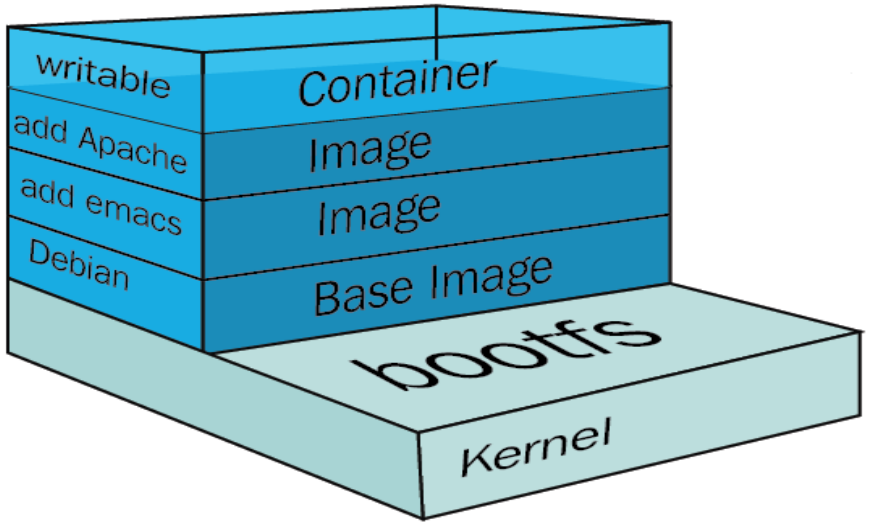
▪ Контейнеры	5
▪ Docker	10
▪ Kubernetes	20
▪ DevOps	30

Семинар раскрывает основные аспекты работы с контейнерами в различных окружениях. Семинар подойдет всем техническим специалистам, кто хочет познакомиться с контейнерами и использовать их в своей работе.

SECTION 1:
Контейнеры

Контейнеры

- **Контейнер – изолированное пространство, в котором запущено приложение, библиотеки, конфигурации и все необходимые компоненты для успешной работы. Изоляцию выполняют механизмы ядра Linux, такие как namespaces (пространства имен) и cgroups (контрольные группы).**



Зачем использовать контейнеры?

- **Контейнеры использует ОС хоста/машины на котором запущен.**
- **Контейнеры потребляют определенное количество ресурсов. Доступна настройка и регулировка потребление CPU, RAM.**
- **Контейнеры по умолчанию ограничены в области видимости. Доступна настройка ограничения/разрешения области видимости, например на сетевом, файловом и т.д. уровне.**
- **Контейнеры работают одинаково в любых окружениях.**

Как начать использовать контейнеры?

SECTION 2:
Docker

Что такое Docker?

Docker – платформа для сборки, хранения, запуска и управления контейнеров.

Dockerfile – инструкция по сборке образа

Docker image – контейнерный образ, собранный из Dockerfile, для запуска контейнеров.

Docker daemon – сервер для запуска и управления контейнерами.

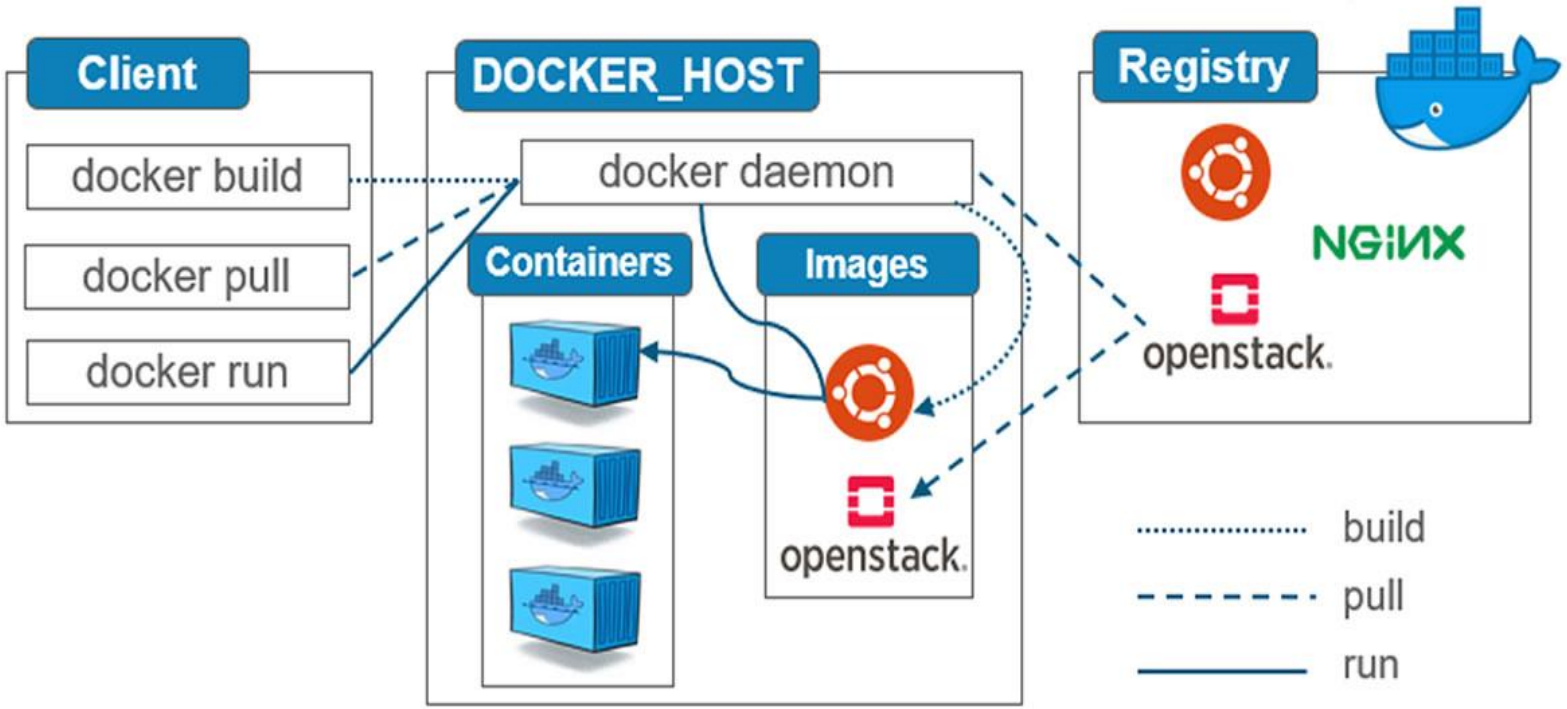
Docker client – клиент, посылающий команды через CLI или REST API серверу.

Dockerhub – репозиторий (docker registry) для хранения контейнеров.

Dockerfile

```
FROM python:3.8.1-slim-buster

WORKDIR /usr/src/app

ENV PYTHONDONTWRITEBYTECODE 1
ENV PYTHONUNBUFFERED 1

RUN rm -rf services

RUN pip install --upgrade pip
COPY ./requirements.txt /usr/src/app/requirements.txt
RUN pip install -r requirements.txt

COPY . /usr/src/app/

CMD ["gunicorn"  , "--bind", "0.0.0.0:8000", "wsgi:app"]
```

Работа с Docker



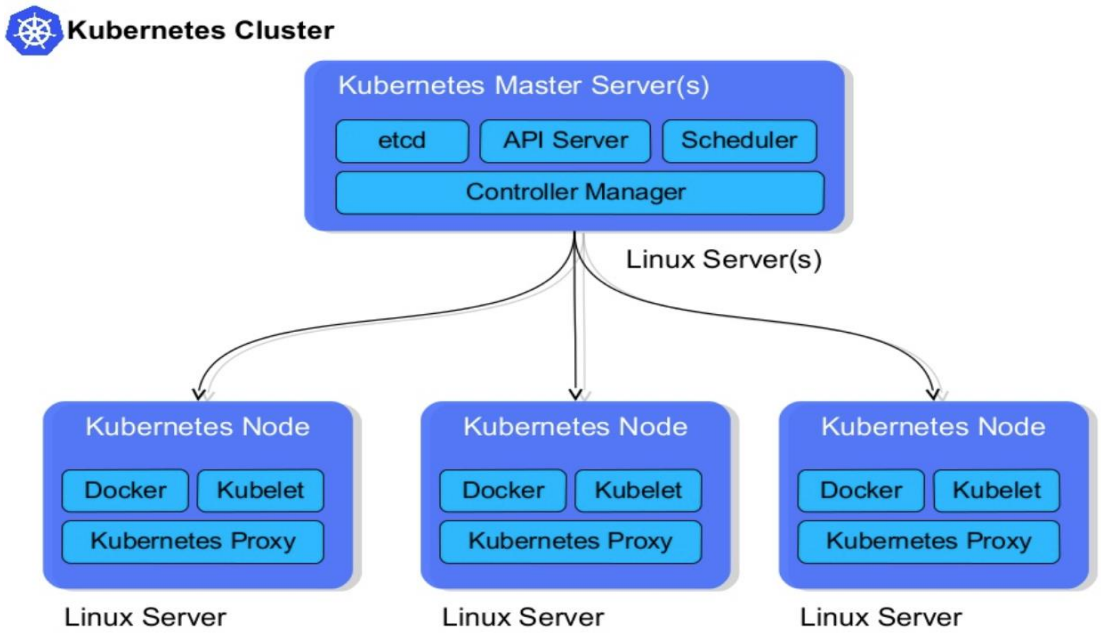
Проблемы при использовании Docker контейнеров

- Сложность управления - как управлять большим количеством контейнеров?
- Отказоустойчивость - что делать если контейнер перестал работать?
- Масштабирование - как их масштабировать? А если нужно откатиться на
- предыдущую версию?
- Балансировка и распределение нагрузки - нужно самому думать как распределять сетевую нагрузку и равномерно распределять контейнеры на хосты
- Stateful приложения - как сохранять и где хранить данные для stateful приложений?

SECTION 3:
Kubernetes

Что такое Kubernetes?

Kubernetes - платформа для управления контейнерами на множесте хостов из единой точки, которая также позволяет настраивать и управлять сетью, конфигурациями для приложений, хранить данные и настраивать работу пользователей.

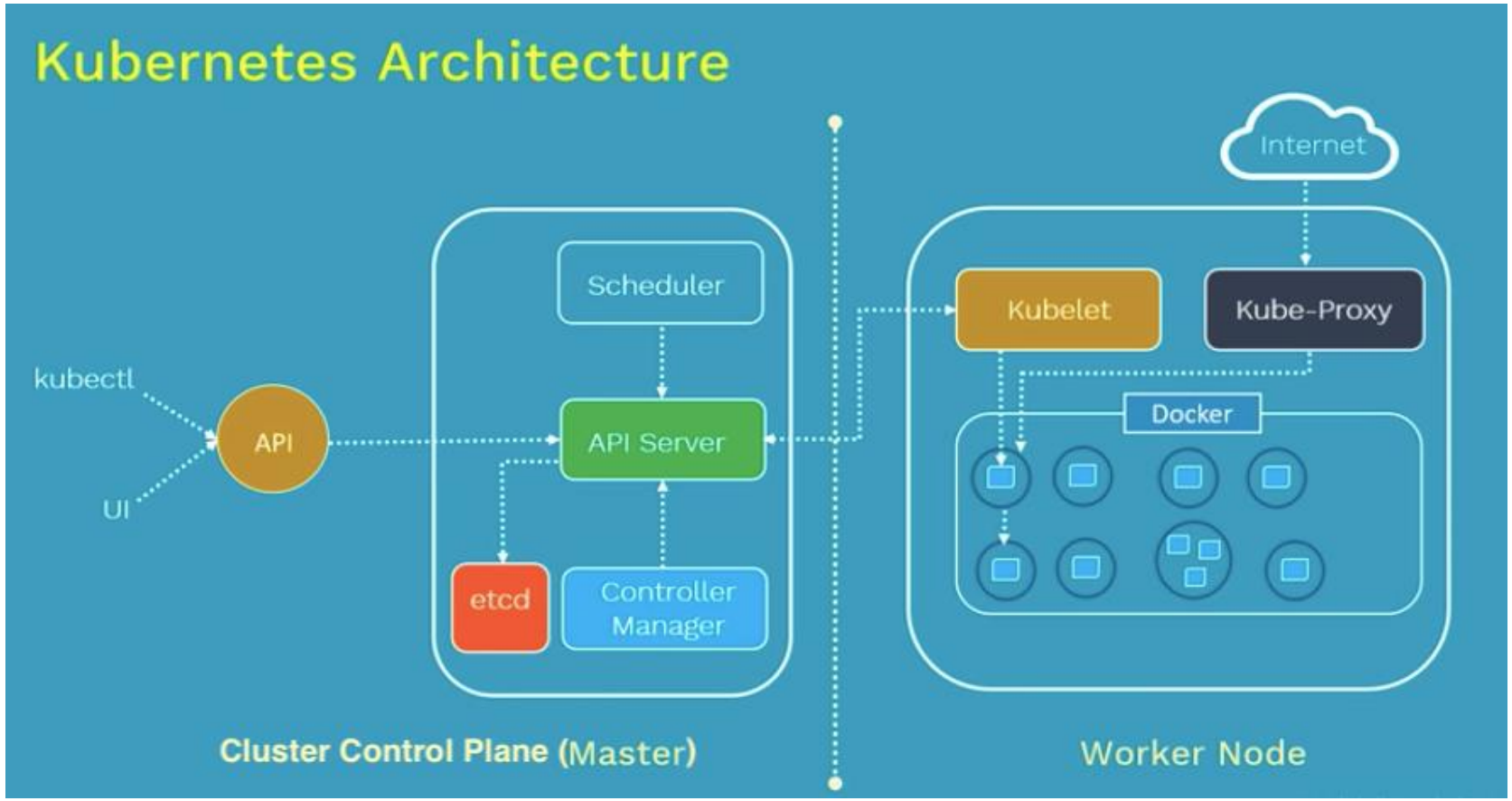


Преимущества kubernetes

- Декларативный подход
- Infrastracture-as-a-code
- Распределение приложений по нодам на основе нагруженности
- Балансировка запросов на реплики приложений
- Self-healing
- Обновление приложений без downtime / Rollback в случае ошибки
- RBAC (управление доступом на основе ролей)
- Интегрируется с разными хранилищами (в том числе облачными)
- Интеграция с разными балансировщиками (в том числе облачными)

Работа с Kubernetes

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hello-app
spec:
  replicas: 1
  selector:
    matchLabels:
      app: hello-app
  template:
    metadata:
      labels:
        app: hello-app
    spec:
      containers:
      - name: hello-app
        image: hello-app
```



SECTION 4:
DevOps

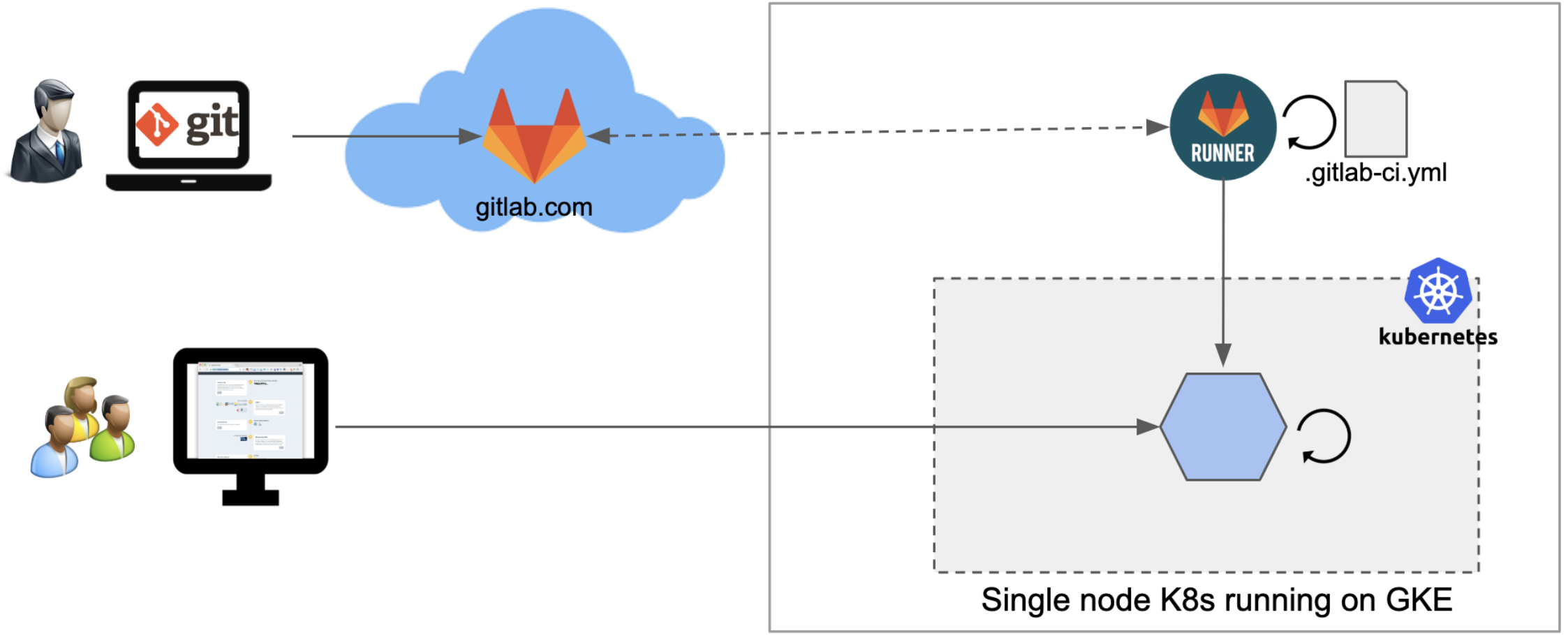
Что такое DevOps ?

DevOps - методология активного взаимодействия специалистов по разработке со специалистами по информационно-технологическому обслуживанию и взаимная интеграция их рабочих процессов друг в друга для обеспечения качества продукта. Предназначена для эффективной организации создания и обновления программных продуктов и услуг. Основана на идее тесной взаимозависимости создания продукта и эксплуатации программного обеспечения, которая прививается команде как культура создания продукта.

Преимущества Kubernetes в разработке и эксплуатации

- **Разработчик - просто пишет код и отправляет его в репозиторий, не отвлекаясь даже на создание и развертывание локальных контейнеров**
- **Инженер - настраивает CI/CD для автоматической сборки, тестирования и выкатки приложений в кластер, применяет методологию инфрастуктура как код и пишет манифесты для всех приложений и настроек(сетевая,конфигурационная), снижая трату времени на ручные действия.**

Как это работает ? Kubernetes + GitlabCI



Практическая часть

Полезные ссылки

https://docs.docker.com/get-started/

https://minikube.sigs.k8s.io/docs/start/

https://github.com/geerlingguy/ansible-role-gitlab

https://docs.gitlab.com/ee/ci

https://github.com/evsq

Thank You!

think.
create.
accelerate.

Пройди опрос

- поделись своим мнением!

https://ru.surveymonkey.com/r/Docker_Kubernetes

Luxoft | training
A DXC Technology Company