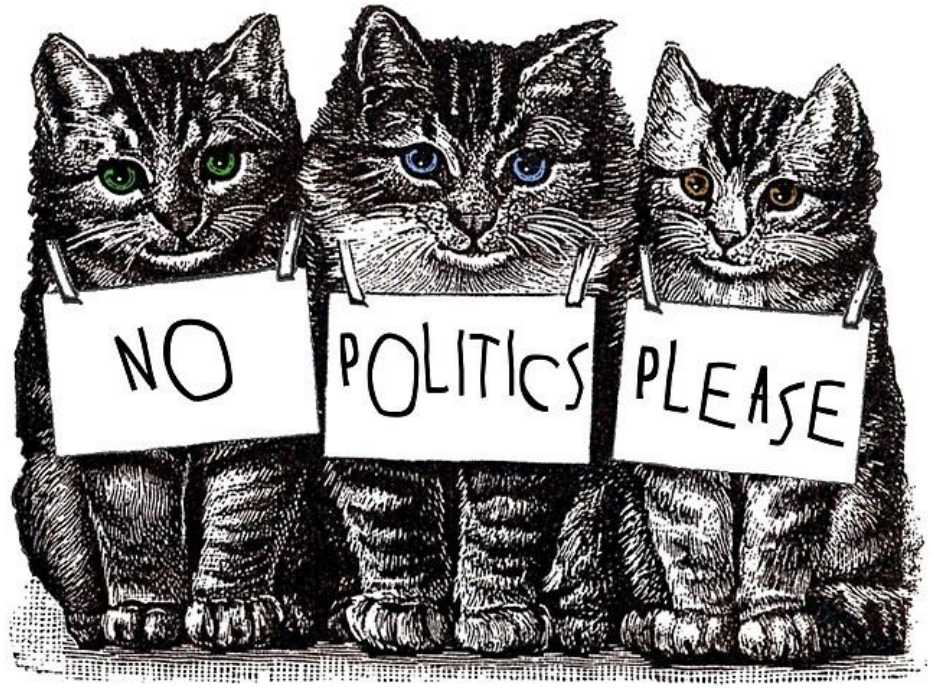
Kubernetes. Начало

РФ/6.09.2022

IBS Training Center


NO
POLITICS
PLEASE

Вступление

Плахутин Евгений

Контакты

Eplakhutin@IBS.ru

Начинал карьеру в ИТ, когда настройка почтового сервера была must have для системного администратора

SEMINAR ROADMAP: OVERVIEW

▪ История проблем	5
▪ Kubernetes, что за зверь?	14
▪ И что дальше ?	32

Кратко коснёмся истории появления оркестраторов и коротко о выгодах использования kubernetes (k8s)

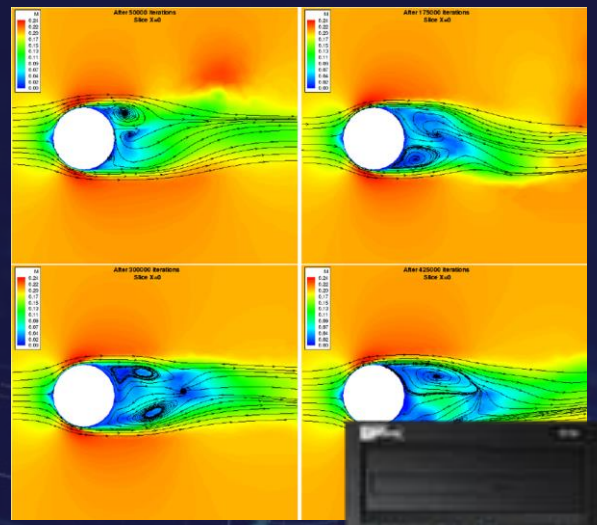
Этот вебинар ориентирован на тех, кто ещё не знает что такое оркестраторы (в частности k8s) и зачем их использовать

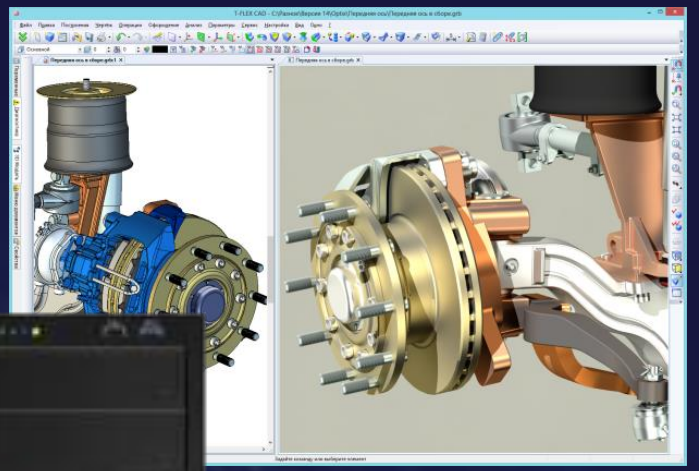
Часть 1:
Краткая история проблем

Сервера большие

Приложения прожорливы

Можно останавливать сервера для обслуживания, никаких SLA





Сколько виртуальных машин, столько проблем и добавляется. Плюс накладные расходы на память, жёсткий диск, лицензии

Hypervisor

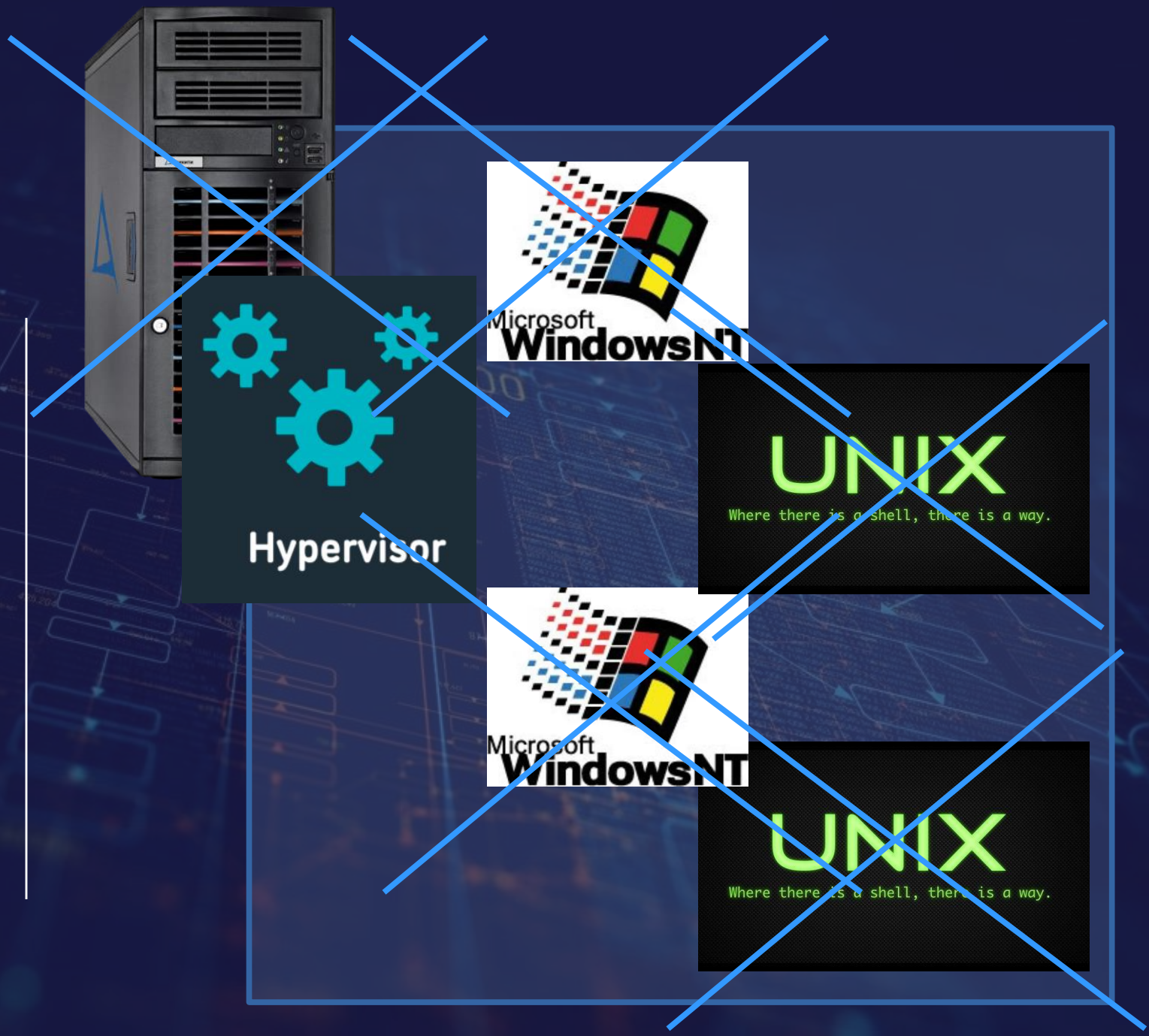
Microsoft WindowsNT

UNIX
Where there is a shell, there is a way.

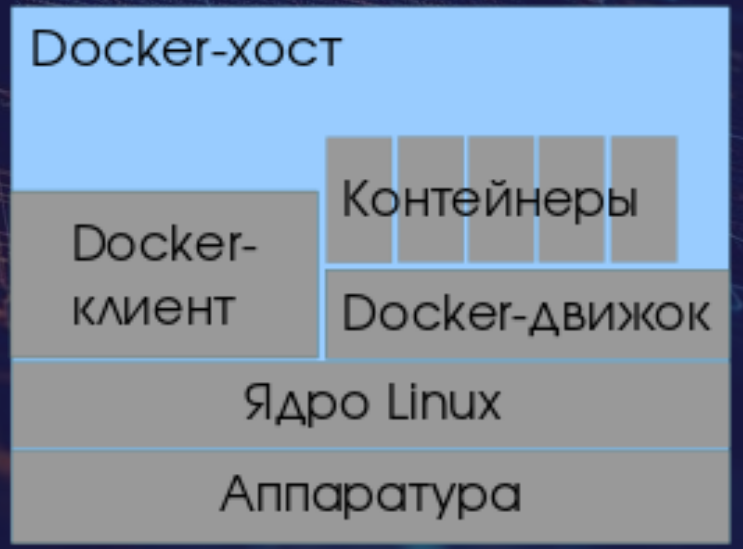
Microsoft WindowsNT

UNIX
Where there is a shell, there is a way.

Единая, глобальная точка отказа — host система или hypervisor



Docker доставит и запустит приложения за пару минут.



Где логи?

Как зайти по ssh?

Не работает localhost!

Как это нельзя поправить руками? Мне надо!

docker run -p 6379:6379 -d redis

```
docker run --restart=always -d --name redis \
  -v /opt/redis/redis.conf:/etc/redis/redis.conf \
  -v /opt/redis/data:/data \
  -p 127.0.0.1:6379:6379 redis redis-server \
  /usr/local/etc/redis/redis.conf
```

Классическая связка

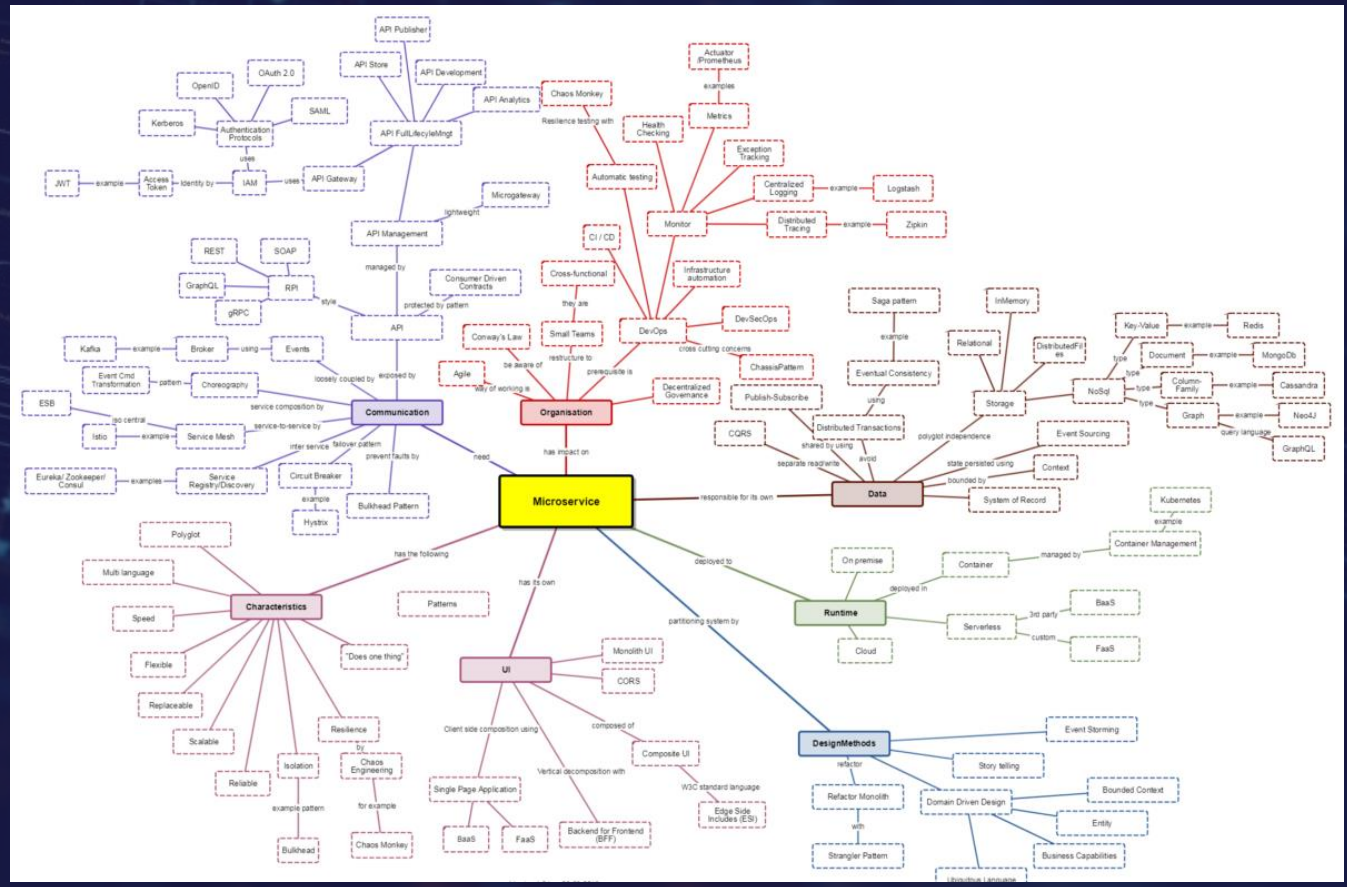
Nginx-phpfpm-mysql

для docker-compose.yaml

```
version: '3'
  services:
    php:
      build:
        context: ./docker/php
      ports:
        - 9000:9000
      volumes:
        - ./:/srv/www/api
        - /php/ww.conf:/etc/ww.conf
      environment:
        MYSQL_USER: ${MYSQL_USER}
        MYSQL_PASSWORD: ${MYSQL_PASSWORD}
    nginx:
      image: nginx:1.13.8
      ports:
        - 80:80
      volumes:
        - ./:/srv/www/api
        - /conf:/etc/nginx/conf.d/.conf
      depends_on:
        - php
    mysql:
      image: mysql:5.7
      ports:
        - 3306:3306
      depends_on:
        - php
      environment:
        MYSQL_ROOT_PASSWORD: ${MYSQL_ROOT_PASSWORD}
        MYSQL_DATABASE: ${MYSQL_DATABASE}
        MYSQL_USER: ${MYSQL_USER}
        MYSQL_PASSWORD: ${MYSQL_PASSWORD}
```

Рост бизнеса — рост нагрузки — рост проблем

Теперь структура сайта компании выглядит так

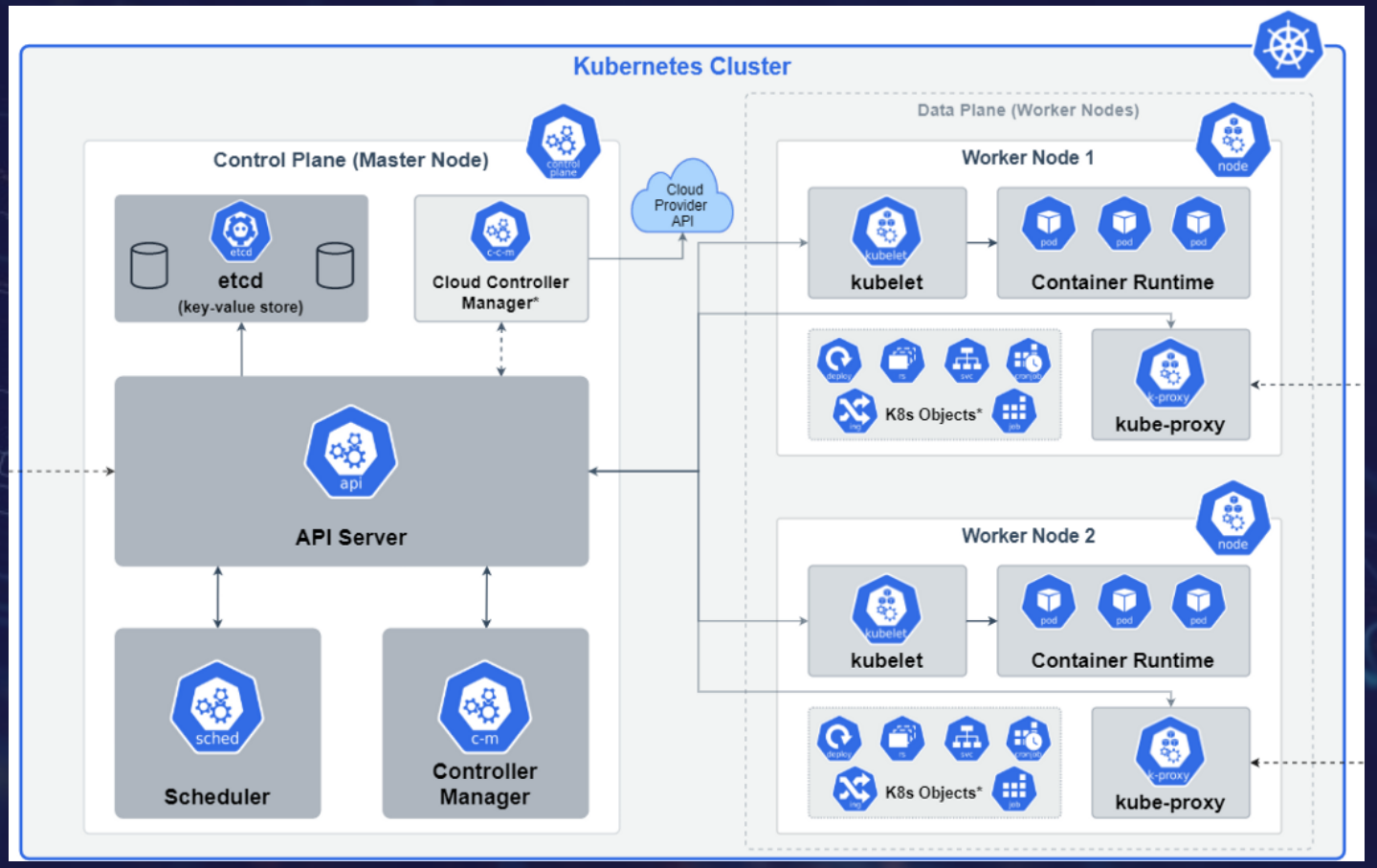


IBS Training Center

Вопросы

Часть 2:
Kubernetes (k8s), что за зверь?

kubernetes


Kubernetes Cluster
Control Plane (Master Node)
etcd
(key-value store)
Cloud Controller Manager*
API Server
Scheduler
Controller Manager
Cloud Provider API
Data Plane (Worker Nodes)
Worker Node 1
kubelet
Container Runtime
K8s Objects*
kube-proxy
Worker Node 2
kubelet
Container Runtime
K8s Objects*
kube-proxy


```
Services:
nginx
    image: nginx:1.13.8
        ports:
            - 80:80
        volumes:
            - /conf:/etc/nginx/conf.d/.conf
```


```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-nginx
spec:
  selector:
    matchLabels:
      run: my-nginx
  replicas: 1
  template:
    metadata:
      labels:
        run: my-nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.13.8
        ports:
        - containerPort: 80
        volumeMounts:
        - mountPath:
/etc/nginx/conf.d
          name: configmap-volume
```

docker run -p 8080:8080 -d MyApp

Kubectl apply -f ./manifest.yaml

cat ./manifest.yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: MyApp
spec:
  containers:
  - name: MyApp
    image: MyApp:0.1
    ports:
    - containerPort: 8080
```

Pod новая сущность, которая запускает контейнер

POD - это абстрактный объект Kubernetes, представляющий собой группу из *одного или нескольких контейнеров* приложения и совместно используемых ресурсов для этих контейнеров.

```
apiVersion: v1
kind: Pod
metadata:
  name: podtest
spec:
  containers:
    - name: container_1
      image: MyApp
      volumeMounts:
        - name: shared_log
          mountPath: /log
    - name: container_2
      image: fluentbit
      volumeMounts:
        - name: shared_log
          mountPath: /log
  volumes:
    - name: shared_log
      emptyDir: {}
```

Общий каталог /log и общий localhost

kubernetes

kubectl apply -f ./manifest.yaml

```
cat ./manifest.yaml

apiVersion: v1
kind: Pod
metadata:
  name: MyApp
spec:
  containers:
  - name: MyApp
    image: MyApp:0.1
    ports:
    - containerPort: 8080
```

Запущенный POD где-то здесь

Важные моменты!

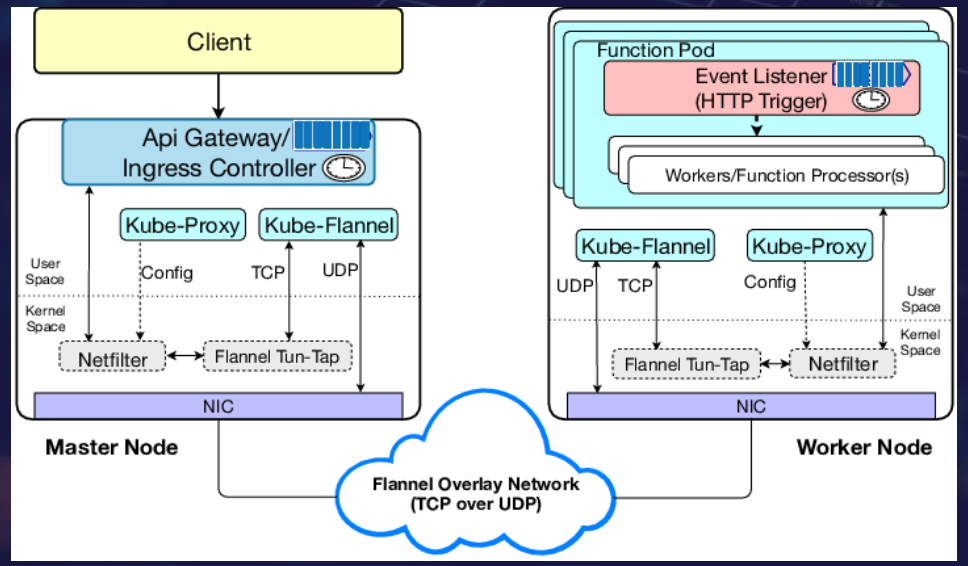
1. По умолчанию, ваш pod может стартовать на почти любом сервере!

2. Поды эфемерны - т.е. под может быть перезапущен в любой момент, и как было сказано в п.1 на любом сервере.

3. Сеть kubernetes это не сеть между серверами! IP адрес сервера это не IP адрес pod.

kubernetes

K8s и сеть – отдельная тема

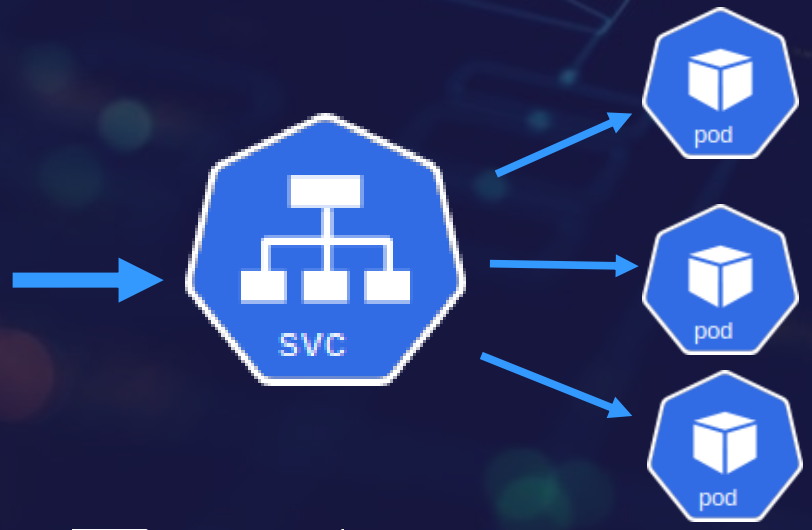


IP для сервера 192.168.0.0/24

IP для pod 10.244.0.0/24

И может меняться при перезапуске

Service — это уровень абстракции, который определяет логический набор pod-ов, перенаправляет внешний трафик и балансирует нагрузку



cat ./manifest.yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: MyApp
  labels:
    app: MyApp
spec:
  containers:
  - name: MyApp
    image: MyApp:0.1
    ports:
    - containerPort: 8080
```

cat ./service.yaml

```
apiVersion: v1
kind: service
metadata:
  name: ServiceMyApp
spec:
 selector:
    app: MyApp
  type: NodePort
  ports:
    - protocol: TCP
      port: 8080
      targetPort: 8080
      nodePort: 31122
```

Условие при котором трафик будет направляться из service в pod

curl 10.244.0.36:8080 → curl ServiceMyApp:8080

Я смог в K8s !
Теперь я девопс

```
docker run -p 80:8080 -d MyApp:0.1
```


```
Services:
nginx
    image: nginx:1.13.8
          ports:
              - 80:80
          links:
            -myapp
          volumes:
              -
/conf:/etc/nginx/conf.d/.conf

myapp
    image: MyApp:0.1
          ports:
              - 8080:8080
```

Можно и так, но это не наш путь

```
kubectl run --image Myapp:0.1 Myapp
kubectl expose --type NodePort --port 8080 pod
Myapp
```

```
cat ./manifests.yaml

apiVersion: v1
kind: Pod
metadata:
  name: MyApp
  labels:
    app: MyApp
spec:
  containers:
  - name: MyApp
    image: MyApp:0.1
    ports:
    - containerPort: 8080

---
apiVersion: v1
kind: service
metadata:
  name: ServiceMyApp
spec:
 selector:
    app: MyApp
  type: NodePort
  ports:
    - protocol: TCP
      port: 8080
      targetPort: 8080
      nodePort: 31122
```

kubernetes

Алгоритм балансировки трафика знает только администратор кластера

svc

```
apiVersion: v1
kind: service
metadata:
  name: ServiceMyApp
spec:
 selector:
   app: MyApp
  type: NodePort
  ports:
    - protocol: TCP
      port: 8080
      targetPort: 8080
      nodePort: 31122
```

pod

```
apiVersion: v1
kind: Pod
metadata:
  name: MyApp_1
  labels:
    app: MyApp
spec:
  containers:
  - name: MyApp
    image: MyApp:0.1
    ports:
    - containerPort: 8080
```

pod

```
apiVersion: v1
kind: Pod
metadata:
  name: MyApp_2
  labels:
    app: MyApp
spec:
  containers:
  - name: MyApp
    image: MyApp:0.1
    ports:
    - containerPort: 8080
```

ReplicaSet — это уровень абстракции, который поддерживает количество pod-ов в любой момент времени

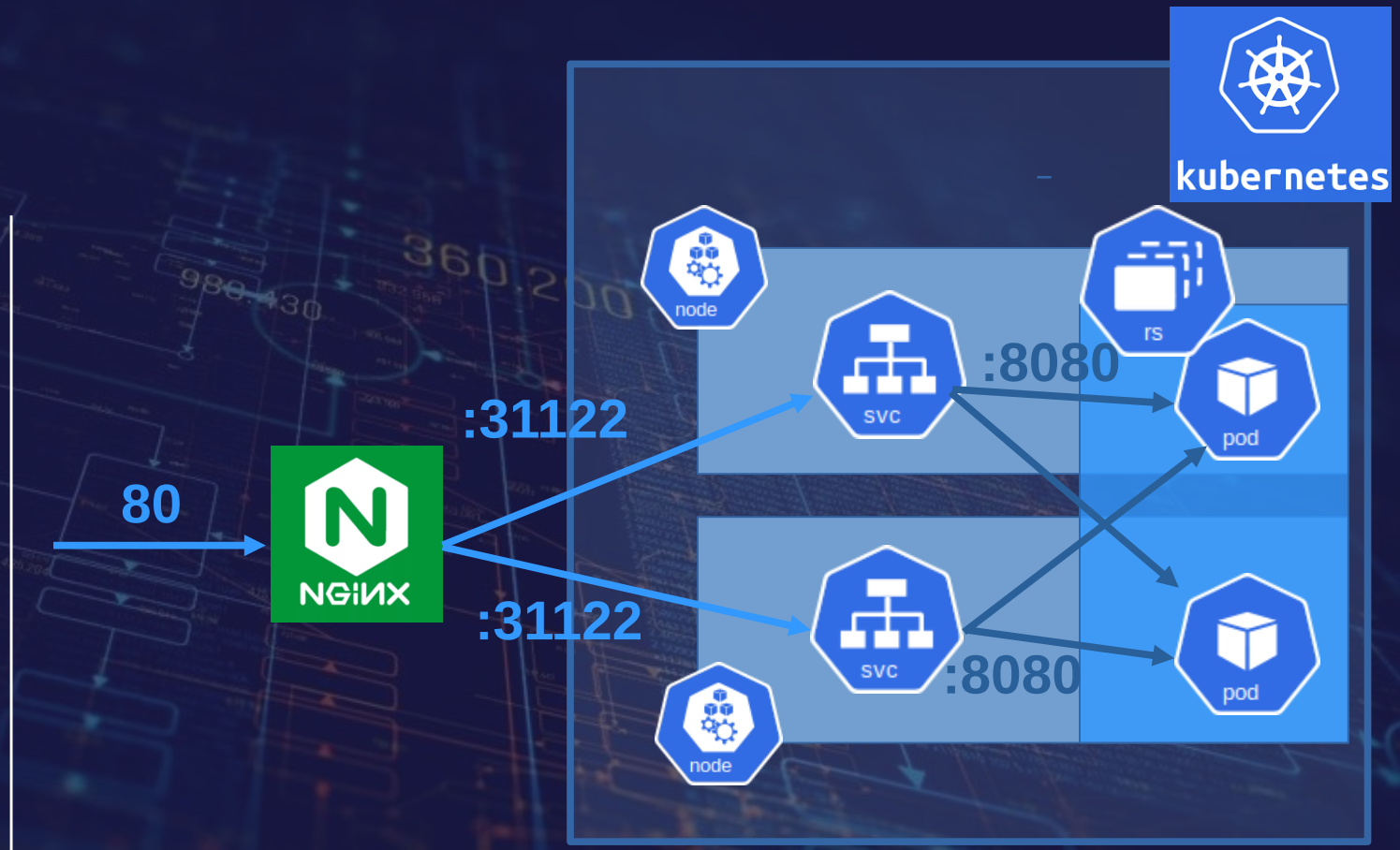
Количество экземпляров (реплик) приложения


```
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: RS-MyApp
spec:
  replicas: 2
  selector:
    matchLabels:
      app: MyApp
  template:
    metadata:
      labels:
        app: MyApp
    spec:
      containers:
            - name: MyApp
              image: MyApp:0.1
              ports:
                - containerPort:
8080
```

По умолчанию управление трафиком не очень богатое. Для расширенного управления необходимо использовать продукты класса service mesh

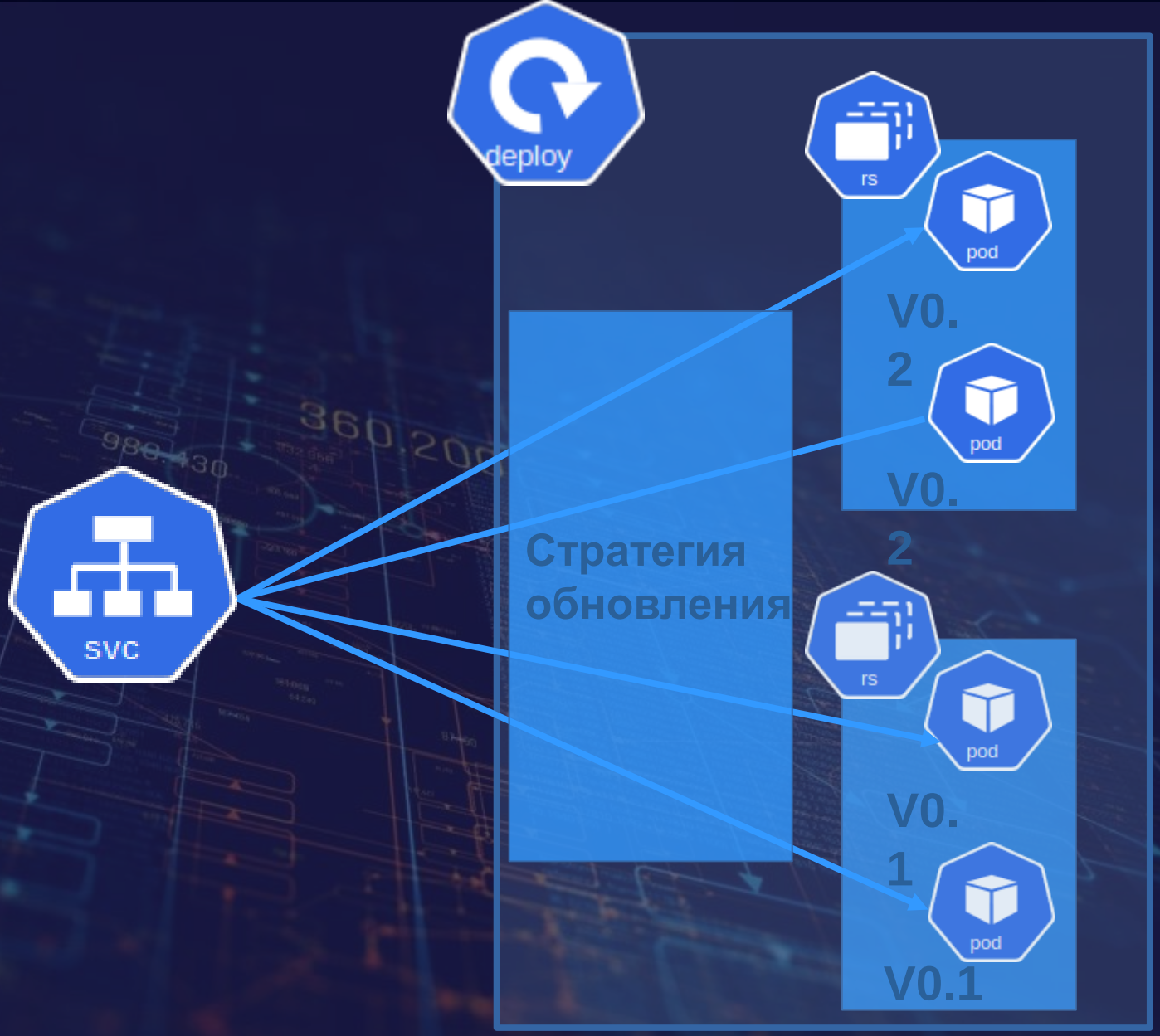
kubernetes

Почти рабочая схема, но нет предела совершенства



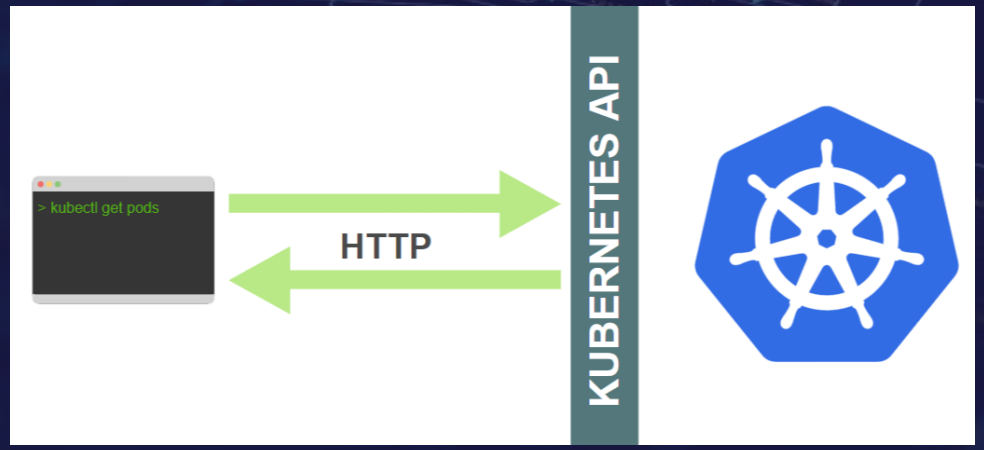
Deployment — это уровень абстракции, который предназначен для развёртывания приложений и их обновлений.

Является предпочтительным способом развёртывание приложения в K8s



kubectl set image deployment/myapp myapp=myapp:0.2

Рассмотренные преимущества



1. При установке приложения kubernetes определит на каком физическом сервере лучше разместить pod-ы, согласно нагрузке серверов и указаных нами требованиях.

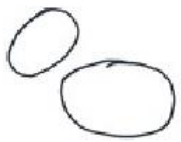
2. Kubernetes следит за количеством рабочих pod-ов, и меняет их согласно указанному в манифесте. (сюда же можно добавить и автомасштабирование)

3. Предаставляет единый доступ к портам приложения через сущность service, в не зависимости на каком физическом сереверe находится поды.

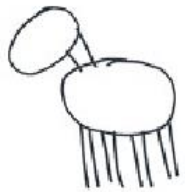
4. При обновлении приложения, согласно написанным политикам, kubernetes поднимит новые pod-ы, убедится что они работают и выключит старые pod-ы. Также дает возможность откатится

Как нарисовать лошадь

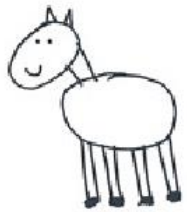
Мы сейчас здесь



1. Сначала 2 круга

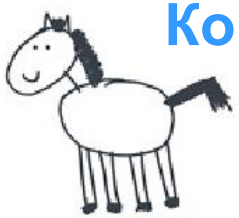


2. Потом 4 ноги



3. Лицо

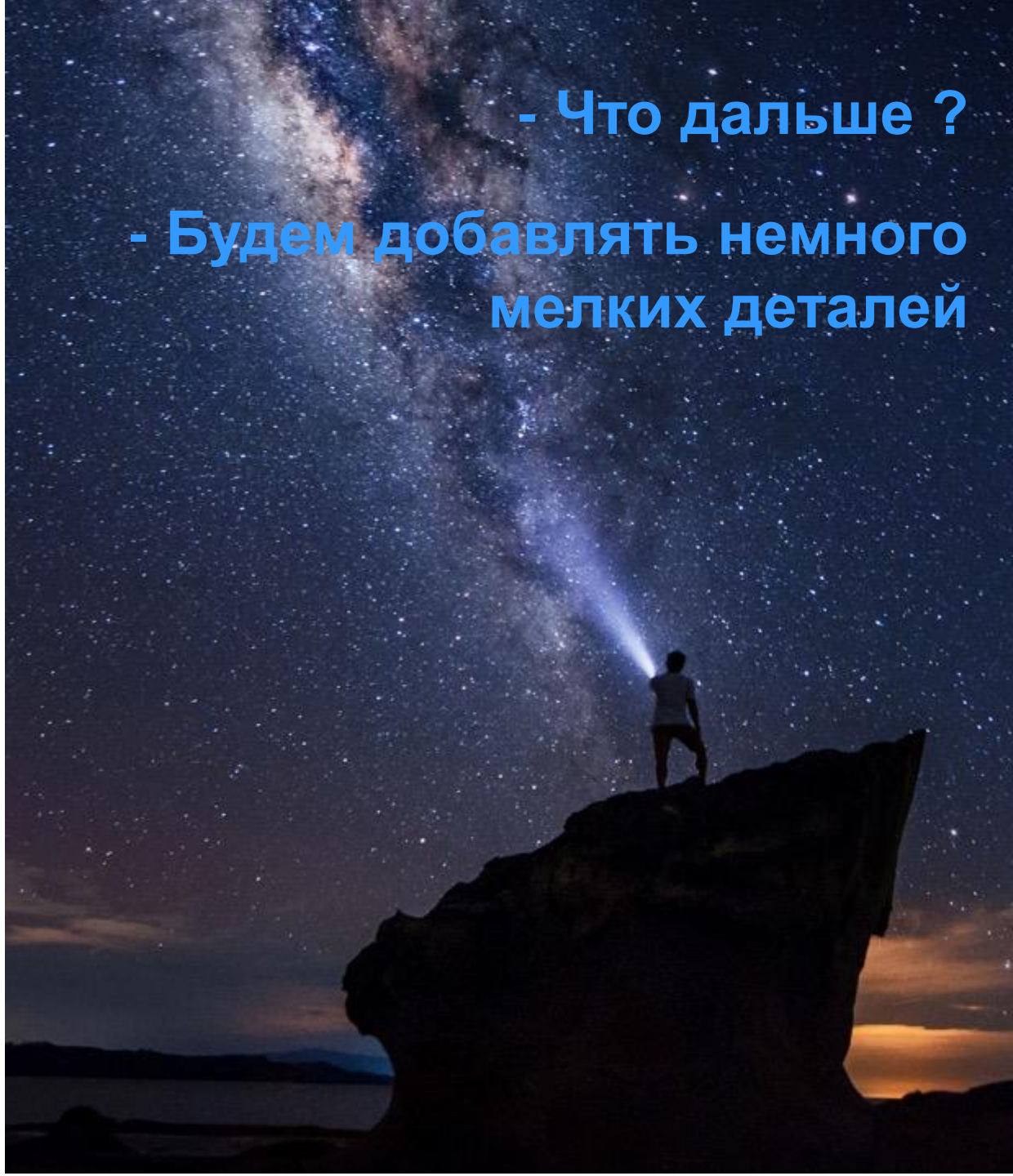
Конец курса



4. Волосы

Самостоятельно

5. Немного мелких деталей



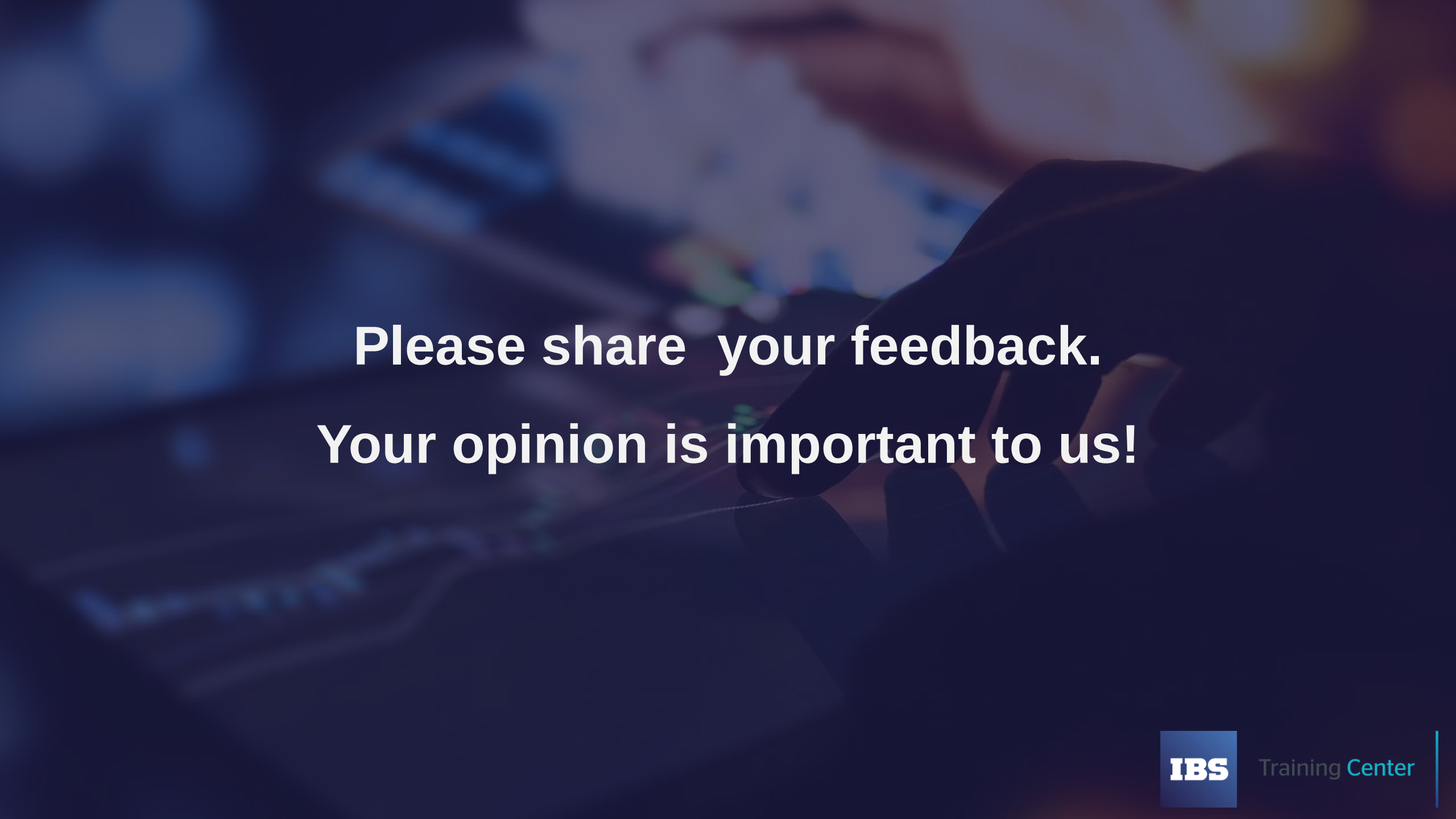
- Что дальше ?

- Будем добавлять немного мелких деталей

Вопросы

Thank you!

IBS Training Center



IBS Training Center