

# В production за 60 секунд

Внедрение kubernetes для динамического создания тестовых окружений



# Об авторе

Свиридов Алексей  
teamlead/senior java developer  
backend-разработки

[https://t.me/sviridov\\_alexey](https://t.me/sviridov_alexey)



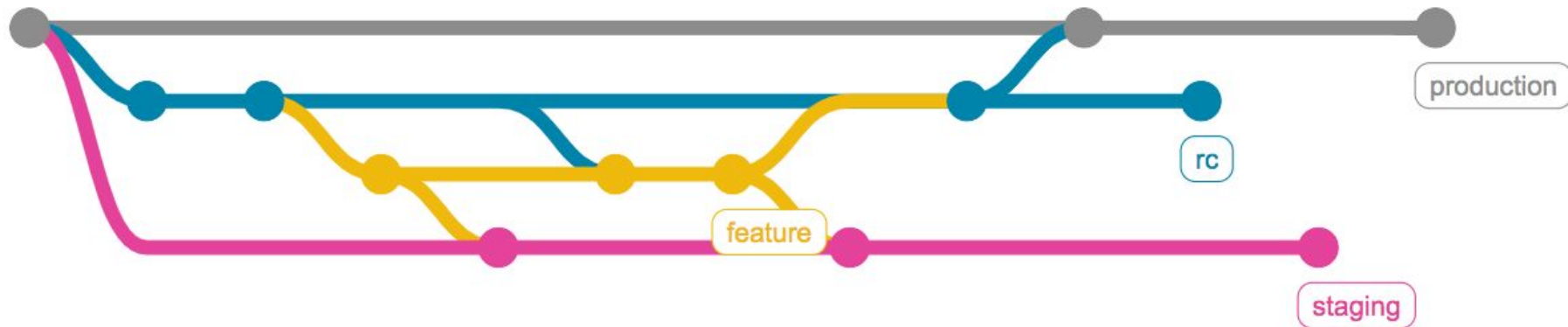
# О проекте

сервер онлайн-касс  
язык разработки - Kotlin  
15 сервисов  
~ 35 000 торговых точек  
(~1500-2000 RPS)

# Что обсудим?

1. Зачем нужно создавать тестовые окружения
2. Проблемы динамического создания окружений существующим стеком
3. Использование Kubernetes, плюсы и минусы
4. Установка и настройка Kubernetes
5. Итоги

# Предыдущая модель разработки



# Проблемы

## Для бизнеса

1. Мелкие фичи долго ждут в RC
2. Неудобно смотреть превью фич

# Проблемы

## Для QA

1. Неизолированное тестирование функционала
2. Проблемы тестирования при появлении новых изменений

# Проблемы

## Для разработчиков

1. Конфликты при merge в staging
2. Несовместимость структур в автотестах
3. Помнить о merge из feature-branch в staging

# Идеальная модель





# Компромиссная модель

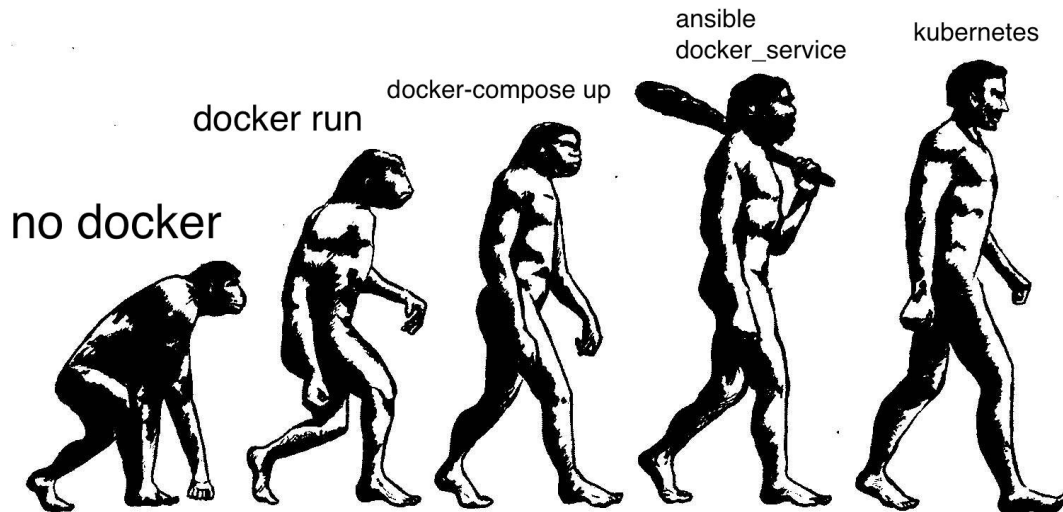


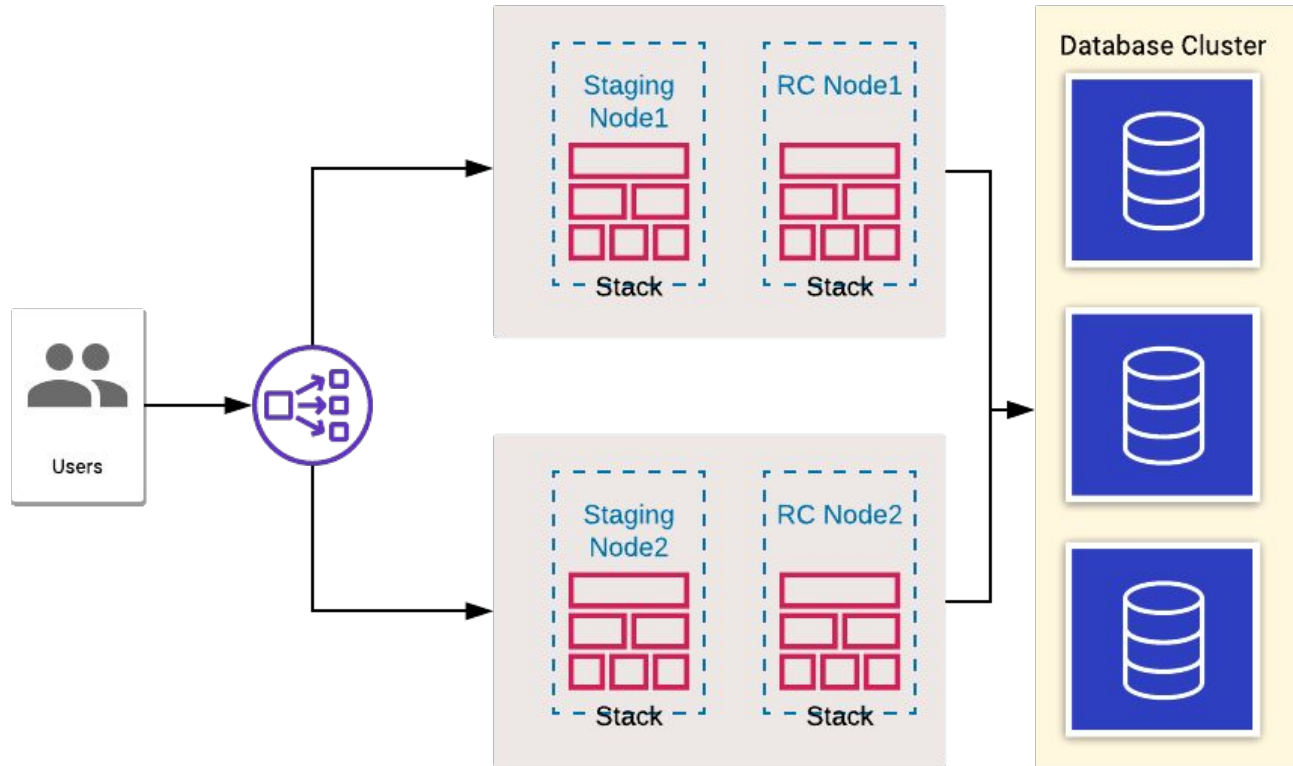
# Что нужно для тестирования feature branch

1. Отдельный домен (mk-1010.dev.modulkassa.local)
2. Сборка сервисов из feature branch
3. Заменить внешние интеграции моками
4. **Создание изолированной инфраструктуры для сервисов**
5. **Развертывание сервисов**
6. **Удалить сервисы и инфраструктуру после тестирования**

# Что имеем на момент внедрения

1. Сборка в docker image
2. Деплой ansible'ом
3. Несколько серверов
4. nginx как LB





# Трудности создания окружения в ansible

1. Как развернуть сервисы на несколько серверов
2. Как избежать конфликта портов на одном сервере
3. Как заводить внешний трафик в окружение
4. Как конфигурировать окружения



# Kubernetes

1. Автоматически распределит сервисы по серверам
2. Namespace'ы чтобы избежать конфликта портов
3. Делает LB для сервисов
4. Не запрещает ходить вовне
5. Можно задавать конфигурацию в namespace'ах

## Проблемы kubernetes на bare metal

1. Где хранить данные
2. Как заводить внешний трафик



# Установка

1. `apt install kubelet kubeadm kubectl && kubeadm init`
2. Сетевой plugin, маска сети pod'ов `--pod-network-cidr=10.244.0.0/16`

## На что обратить внимание

1. Отключить swap
2. Достаточно места на дисках
3. Уникальные hostname (Ubuntu-1804-bionic-64-minimal)
4. `kubeadm reset`
5. Записать вывод `kubeadm init` для добавления серверов



# Настройка

1. Установить сетевой plugin
2. Добавить в кластер сервера (kubeadm join)
3. Доступ извне
  - a. NodePort
  - b. MetalLB
4. Маршрутизация трафика внутри кластера
  - a. Ingress (nginx, traefik, kong, envoy ...)
  - b. ambassador



**Ambassador**

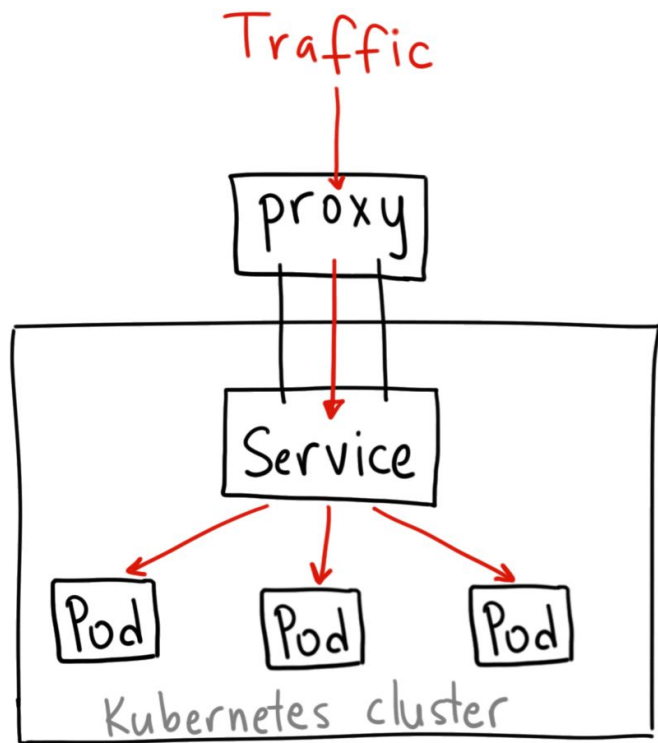


# MetalLB

```
apiVersion: v1
kind: ConfigMap
metadata:
  namespace: metallb-system
  name: config
data:
  config: |
    address-pools:
    - name: my-ip-space
      protocol: layer2
      addresses:
      - 10.7.7.150-10.7.7.200
```

# MetalLB

```
apiVersion: v1
kind: Service
spec:
  ports:
    - port: 9000
      targetPort: 8080
  selector:
    service: ambassador
  type: LoadBalancer
```



# Ingress?

```
apiVersion: networking.k8s.io/v1beta1
kind: Ingress
spec:
  rules:
  - http:
    paths:
    - path: /mk9901/stock
      backend:
        serviceName: stock.mk9901
        servicePort: 8080
    - path: /mk9950/stock
      backend:
        serviceName: stock.mk9950
        servicePort: 8080
```

# Ingress? Ambassador!

```
apiVersion: v1
kind: Service
metadata:
  name: "stock"
  namespace: "{{ env }}"
  annotations:
    getambassador.io/config: |
      ---
      apiVersion: ambassador/v1
      kind: Mapping
      name: "stock-{{ env }}_mapping"
      prefix: "/{{ env }}/stock/"
      service: "stock.{{ env }}:8083"
```

1

2

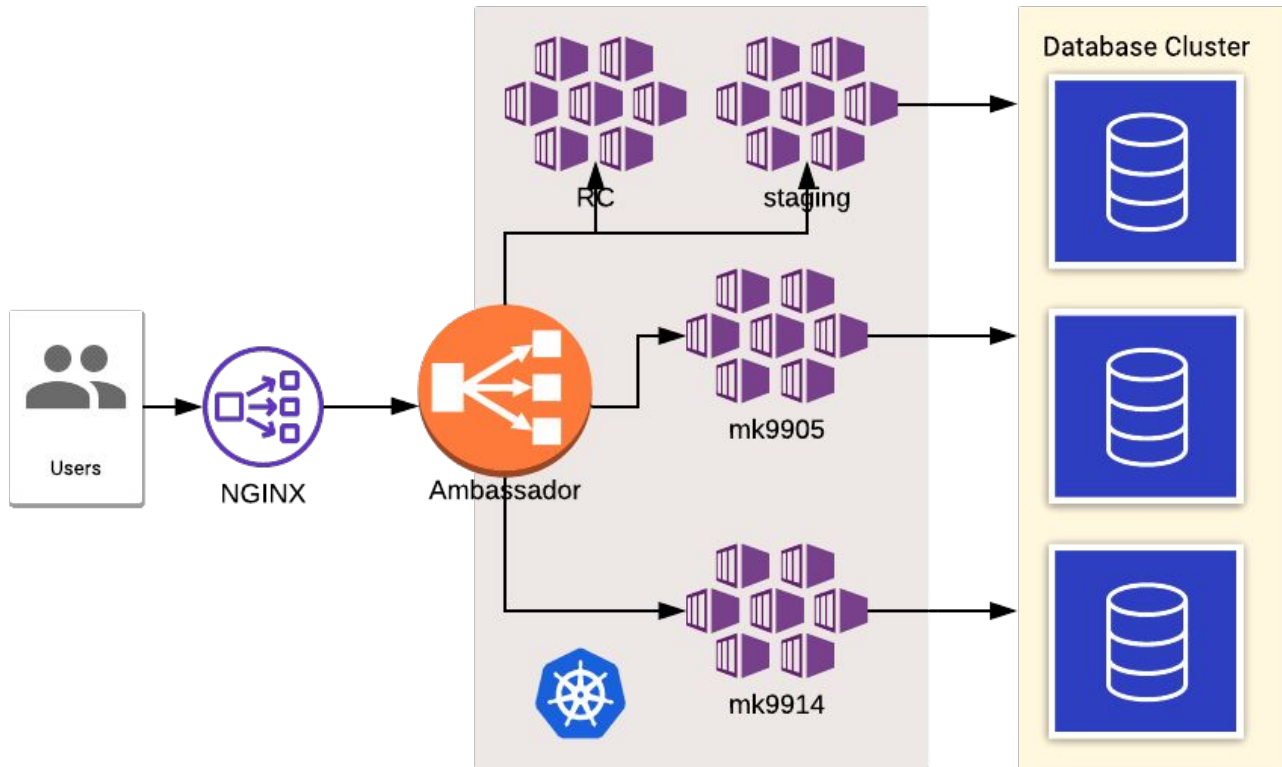
# Запускаем трафик в кластер

```
[fred Mac-mini] - [~/tmp] - [2019-07-20 06:44:17]
[0] <> kubectl get svc
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
ambassador	LoadBalancer	10.109.200.184	10.7.7.151	9000:32425/TCP	39d
ambassador-admin	NodePort	10.101.106.104	<none>	8877:31122/TCP	39d
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	42d

```
location /api/sas/ {
    proxy_pass http://10.7.7.151:9000/mk9838/sas/;
}
```

```
location /api/stock/ {
    proxy_pass http://10.7.7.151:9000/mk9838/stock/;
}
```



# Отличие деплоя ansible и k8s



ansible



k8s



# Удаляем окружение

1. Удалить namespace k8s
2. Удалить созданные данные
3. Удалить домен



# Результаты и выводы

1. Значительно сократилось количество конфликтов
2. Сократилось время вывода мелких фич/фиксов в production
3. Более надежные результаты тестирования
4. Технология доступная, практические результаты достигаются очень быстро

# Q&A

[https://t.me/sviridov\\_alexey](https://t.me/sviridov_alexey)

A solid orange horizontal bar spanning the width of the slide at the bottom.