
# Deployment and DeploymentConfigs in OpenShift

Denis Moiseev
Senior Quality Engineer

Bohdan Iakymets
Software Engineer



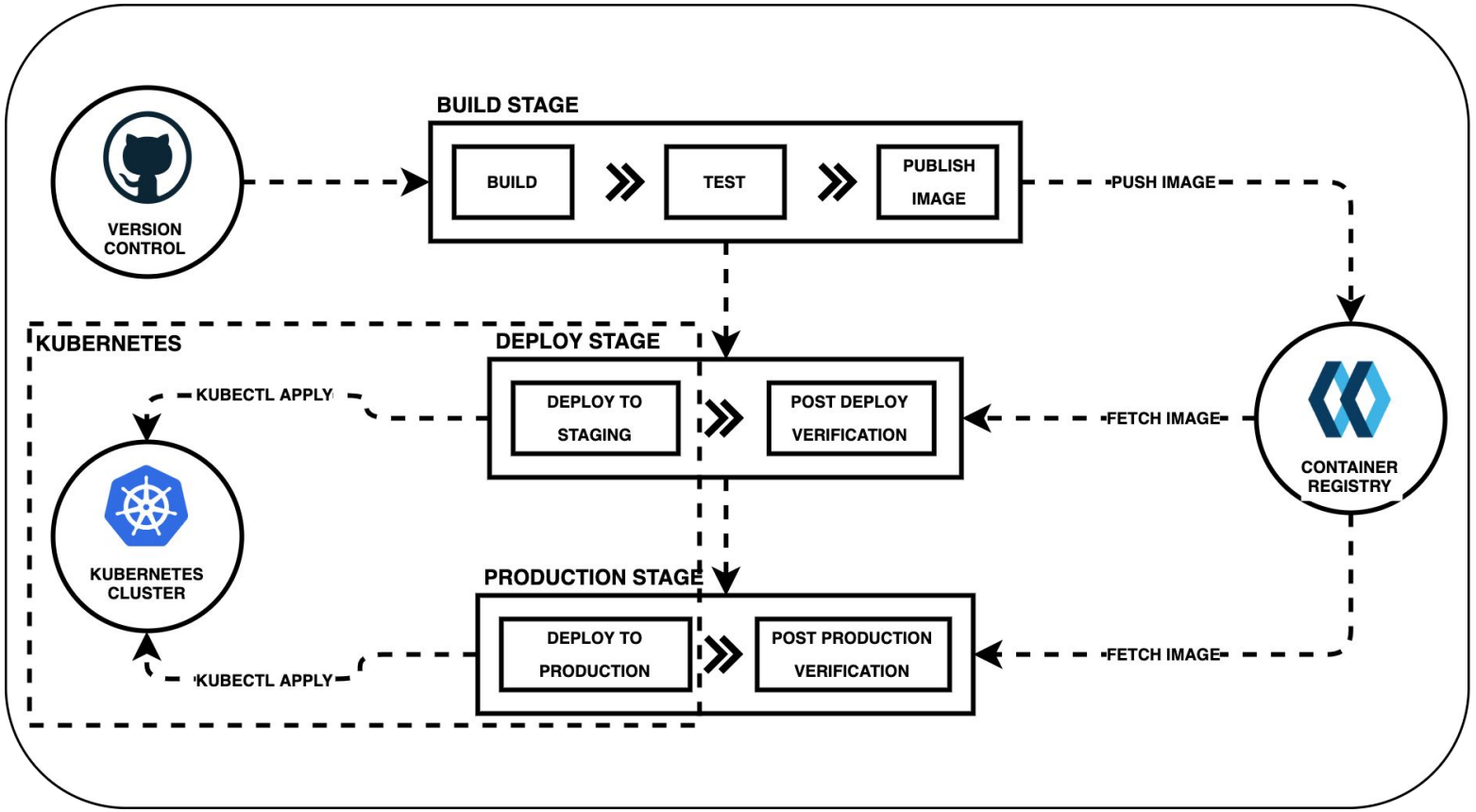
# Agenda

- Сборка и доставка приложений в k8s
- Что может предложить OpenShift?
- Deployment / DeploymentConfigs, Builds, ImageStreams, S2i - что это и зачем?
- Собираем все вместе

# Предполагаемые ограничения

- **Нет** привелегий cluster-wide администратора
- Вы **не** хотите разрабатывать оператор
- Использование 3rd party сервисов нежелательно либо ограничено
- Установка cluster-wide приложений затруднена (см первый пункт)

# App Delivery Workflow with K8s






Red Hat

# Usually in app development and delivery workflow we use a lot of third-party tools and services.

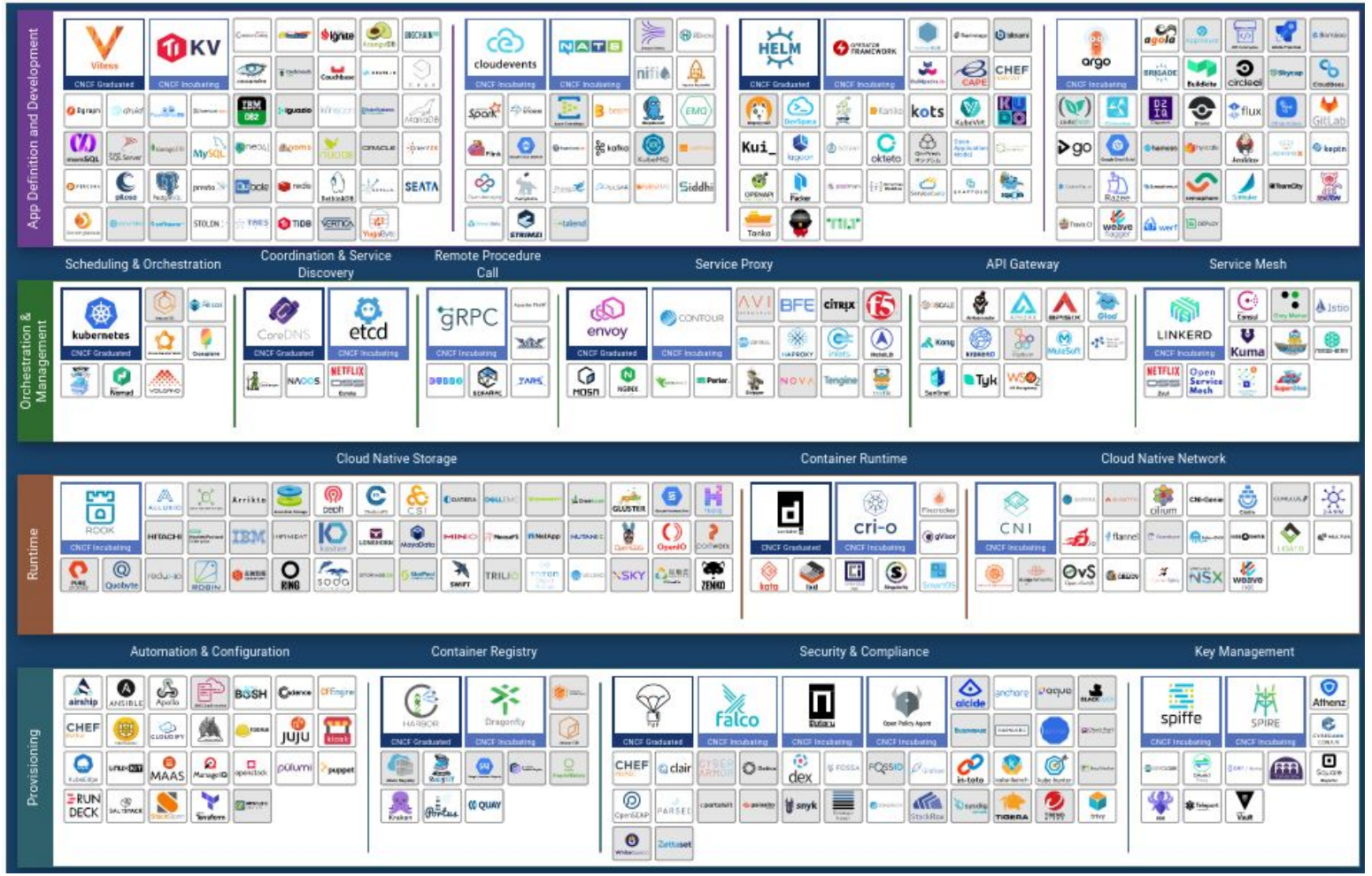
## Build Stage bits

- Gitlab
- Jenkins
- TeamCity
- ...

## Publish and Deploy Stage bits

- Gitlab
- Helm
- Ansible
- Jenkins
- DockerHub
- Quay.io
- ArgoCD
- ...

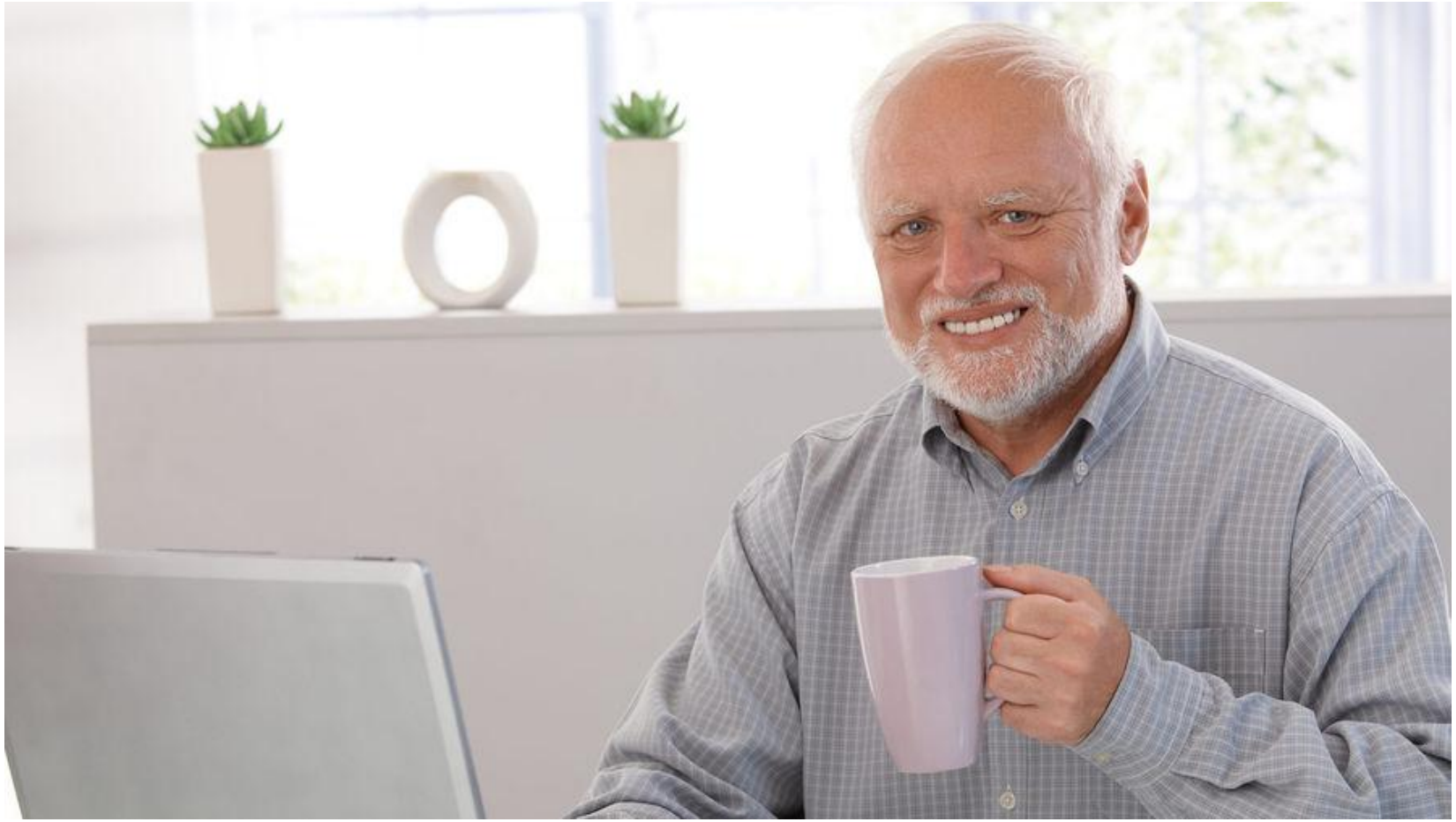
# Even more...



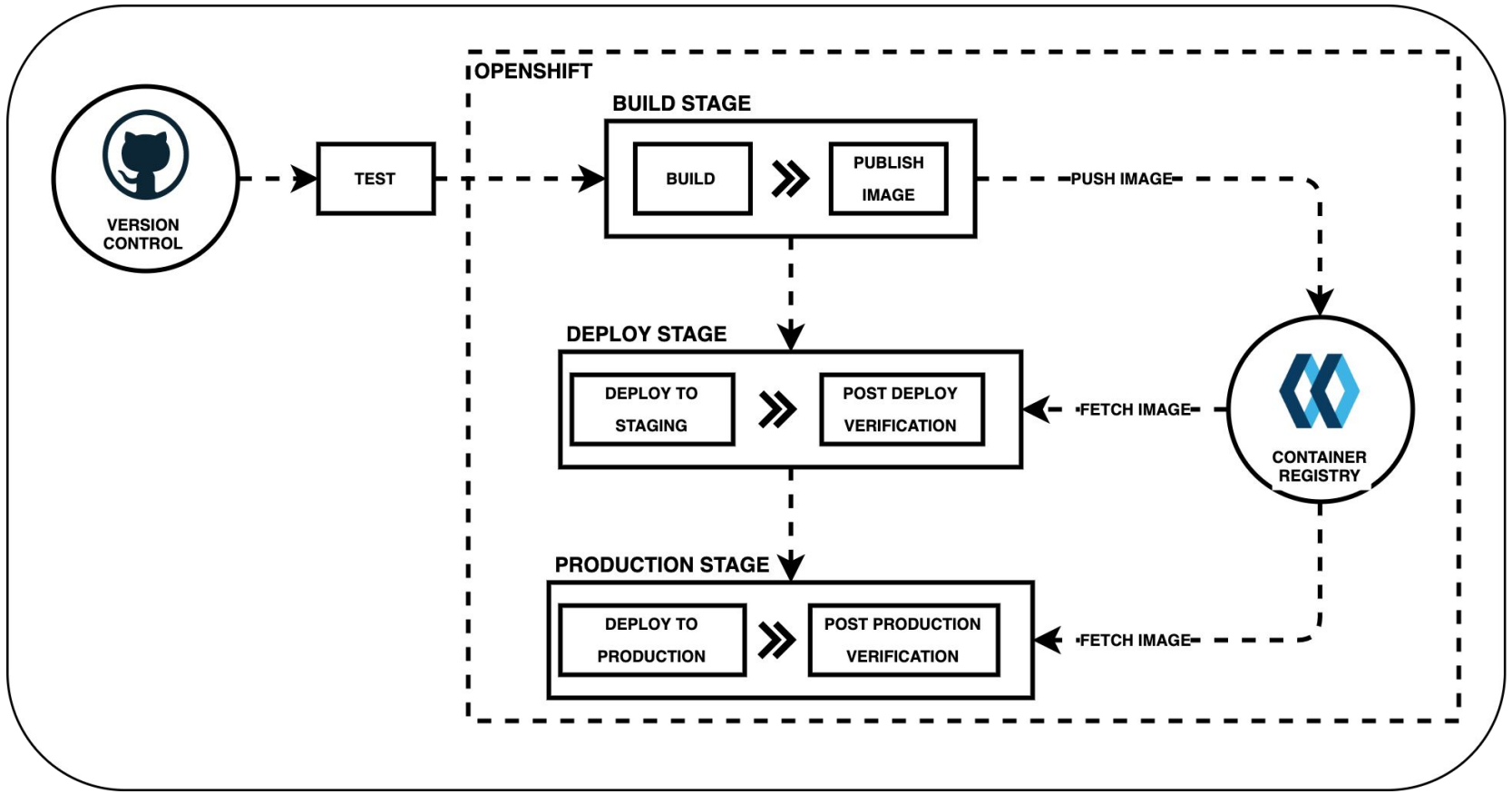



Red Hat

# K8s это просто, весело и интересно...



# OpenShift Delivery Workflow






Red Hat

# Usually in app development and delivery workflow we use a lot of third-party tools and services.

Build Stage bits

Publish and Deploy Stage bits

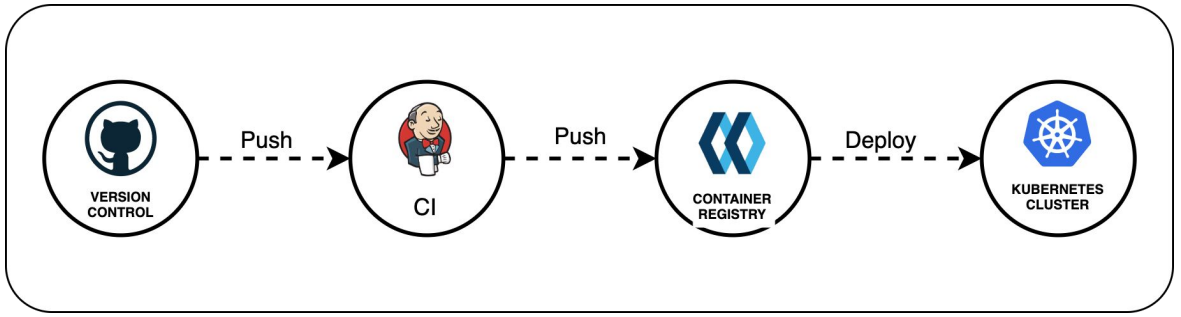
One response for all:

**OpenShift!**

# Собираем / Тестируем

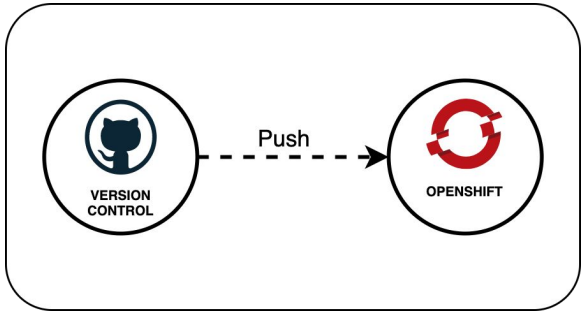
## Pure k8s

- Jenkins
- Travis
- TeamCity
- ...
- ???




## OpenShift

- Builds






Red Hat

# Builds

```yaml
- apiVersion: v1
  kind: BuildConfig
  metadata:
    name: ${APP_NAME}
    labels:
      app: ${APP_NAME}
  spec:
    successfulBuildsHistoryLimit: 1
    failedBuildsHistoryLimit: 1
    output:
      to:
        kind: ImageStreamTag
        name: ${APP_NAME}:latest
    source:
      git:
        uri: ${SOURCE_REPOSITORY_URI}
        ref: master
      contextDir: /
      type: Git
    strategy:
      type: Source
      sourceStrategy:
        from:
          kind: ImageStreamTag
          namespace: openshift
          name: 'golang:latest'
    triggers:
      - type: GitHub
        github:
          secretReference:
            name: ${WEBHOOK_SECRET_NAME}
```

- Создает контейнер в котором (как-правило) собирает Docker образ вашего приложения
- Загружает собранный образ в ImageRegistry / ImageStream
- Имеет разные стратегии для сборки (Из Dockerfile, S2i, Pipeline)
- умеет работать с существующими docker образами
  - (пример - достать бинарник и положить в новый образ)
- Pipeline стратегия позволяет использовать Jenkinsfile для выполнения практически чего угодно
- ( Да, оно поднимет Jenkins. Да, Groovy :) )

# S2i

```yaml
- apiVersion: v1
  kind: BuildConfig
  metadata:
    name: ${APP_NAME}
    labels:
      app: ${APP_NAME}
  spec:
    successfulBuildsHistoryLimit: 1
    failedBuildsHistoryLimit: 1
    output:
      to:
        kind: ImageStreamTag
        name: ${APP_NAME}:latest
    source:
      git:
        uri: ${SOURCE_REPOSITORY_URI}
        ref: master
      contextDir: /
      type: Git
    strategy:
      type: Source
      sourceStrategy:
        from:
          kind: ImageStreamTag
          namespace: openshift
          name: 'golang:latest'
    triggers:
      - type: GitHub
        github:
          secretReference:
            name: ${WEBHOOK_SECRET_NAME}
```

- Одна из стратегий для Build
- Заранее подготовленный контейнер сборщик
- Управляется с помощью shell скриптов (скрипты для сборки, запуска, сохранения артефактов)
- Умеет в инкрементальную сборку, может переиспользовать артефакты или установленные зависимости с предыдущих запусков

# Build Webhooks and Triggers

```
- apiVersion: v1
  kind: BuildConfig
  metadata:
    name: ${APP_NAME}
    labels:
      app: ${APP_NAME}
  spec:
    successfulBuildsHistoryLimit: 1
    failedBuildsHistoryLimit: 1
    output:
      to:
        kind: ImageStreamTag
        name: ${APP_NAME}:latest
    source:
      git:
        uri: ${SOURCE_REPOSITORY_URI}
        ref: master
      contextDir: /
      type: Git
    strategy:
      type: Source
      sourceStrategy:
        from:
          kind: ImageStreamTag
          namespace: openshift
          name: 'golang:latest'
    triggers:
      - type: GitHub
        github:
          secretReference:
            name: ${WEBHOOK_SECRET_NAME}
```

- Билды можно запускать с помощью HTTP запросов
- Есть поддержка популярных сервисов (Gitlab, Github, Bitbucket)
- Может следить за вашими ImageStreams, запускаться если появился новый образ

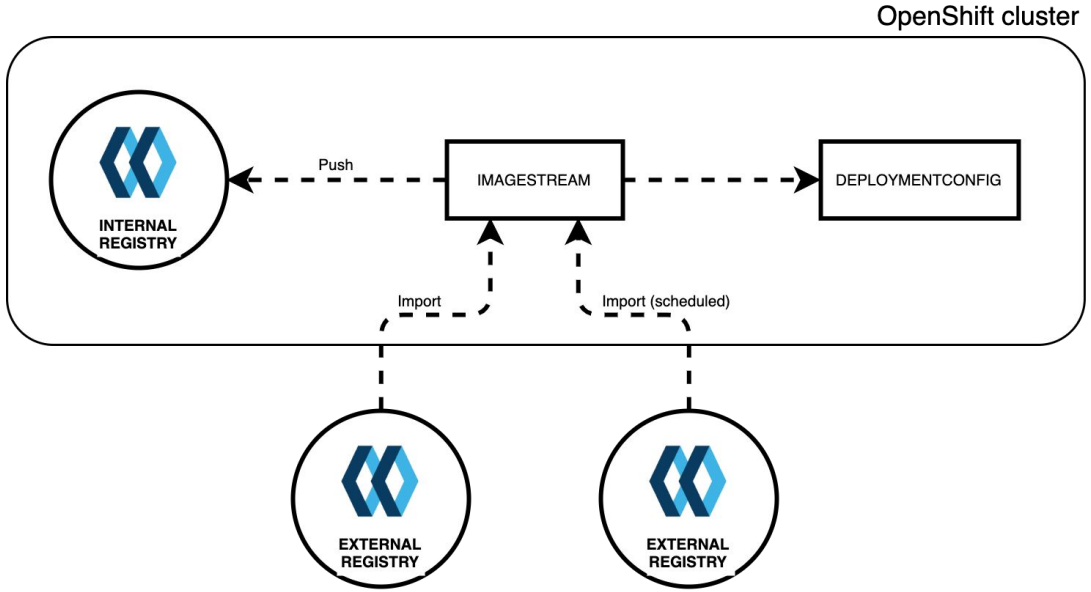
# Публикуем

## Pure k8s

- DockerHub
- Quay.io
- Gitlab.com
- Azure
- Aws
- ...
- собственное registry

## OpenShift

- ImageStreams






Red Hat

# ImageStreams

```
- apiVersion: v1
  kind: ImageStream
  metadata:
    name: ${APP_NAME}
    labels:
      app: ${APP_NAME}
```

- Встроенный registry
- Привязан к неймспейсу
- Есть возможность синхронизации с внешними registry (DockerHub)
- Можно использовать теги
- Настроив RBAC / сервисаккаунт возможно использовать ImageStream из другого неймспейса

# Deployments

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.14.2
        ports:
        - containerPort: 80
```

- Декларативное описание *желаемого* состояния ваших подов / репликасетов
- Работает с ReplicaSet
- Откат к предыдущей версии (хранит историю)
- Масштабирование
- Предпочитает доступность консистентности

# DeploymentConfigs

```yaml
apiVersion: v1
kind: DeploymentConfig
metadata:
  name: frontend
spec:
  replicas: 5
  selector:
    name: frontend
  template: { ... }
  triggers:
  - type: ConfigChange
  - imageChangeParams:
      automatic: true
      containerNames:
      - helloworld
      from:
        kind: ImageStreamTag
        name: hello-openshift:latest
    type: ImageChange
  strategy:
    type: Rolling
```

- По сути то же самое что и Deployments, но с дополнительными функциями
- Триггеры
- Lifecycle Hooks
- Предпочитает консистентность доступности

# ReplicationController / ReplicaSet

- Следит за количеством подов (реплик)
- Для выборки подов использует селектор (лейблы)
- Редко используются напрямую, создаются с помощью Deployment / DeploymentConfig

ReplicationController

- Выборка по точному совпадению селектора

ReplicaSet

- Преемник ReplicationController
- Семантика множеств для селекторов

# Deployment / DeploymentConfig

## Deployment

- Prefer availability over consistency
- Compatible with vanila k8s
- Using ReplicaSets

## DeploymentConfig

- Prefer consistency
- Feature only of Openshift
- Using ReplicationControllers
- Lifcecycle hooks
- Triggers
- Automatic Rollbacks

*Оба варианта поддерживаются в Openhsift Container Platform, однако рекомендуется использовать Deployment, если нет нужды в DeploymentConfig-специфичных функциях / поведении.*

# Consistency and Availability

- Согласно теореме CAP возможно обеспечить два из трех следующих свойств
  - C - consistency
  - A - availability
  - P - partition tolerance

- DeploymentConfig - остановка узла на котором запущен под с деплоером - остановит весь процесс до тех пор пока узел не вернется или не будет удален из кластера

- Deployment - процесс деплоймента может быть обработан другим мастер-узлом

# Triggers

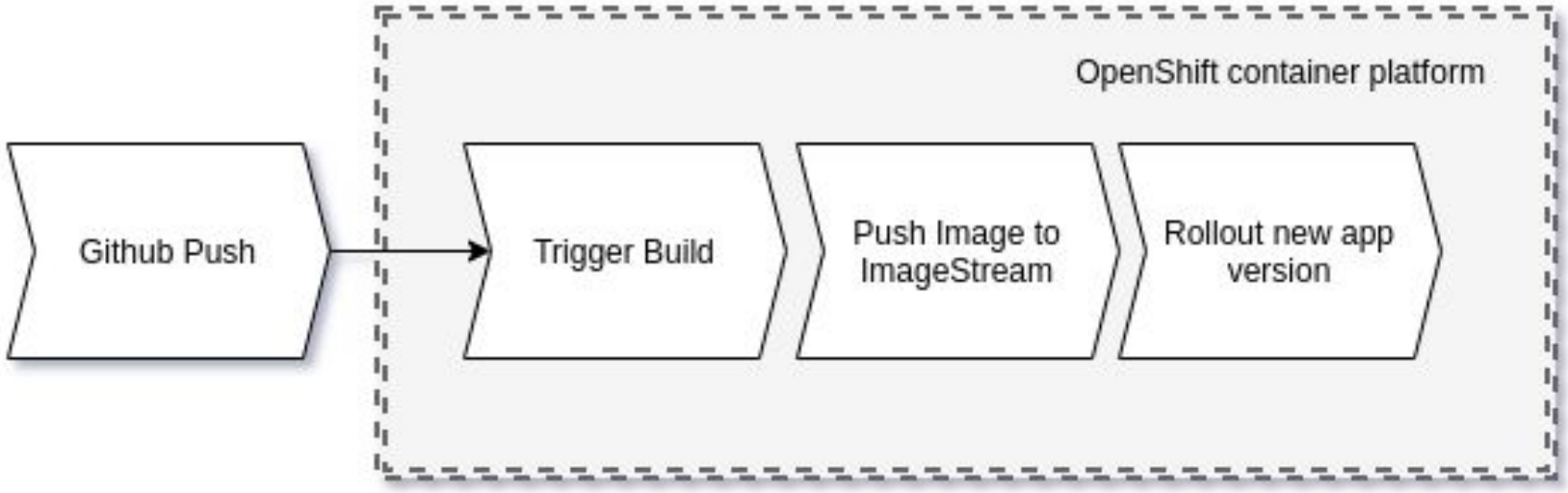
- Реакция на некоторые события внутри кластера
- ImageChange - новый образ в ImageStream
- ConfigChange - изменение самого ресурса
- Довольно удобно в процессе разработки, не требует сложной конфигурации

# Lifecycle Hooks

- Для Rolling и Recreate стратегий
- Поднимает Pod для  выполнения чего либо на определенном этапе развертывания DeploymentConfig
- Примеры задач которые можно этим решать
    - Миграция бд
    - Оповещения об изменениях

# Demo Time

# Put it together



# Где поиграть?

OCP Кластер:

- https://www.openshift.com/products/online/

  Или

- https://learn.openshift.com/playgrounds/openshift44/

Используемый в демо репозиторий:

- https://github.com/lobziik/ocp-features-webinar

# Что почитать / посмотреть?

- Builds
- Available S2i builders
- Deployment / DeploymentConfig
- ImageStreams
- Openshift on github
- learn.openshift.com

# Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.


linkedin.com/company/red-hat


youtube.com/user/RedHatVideos


facebook.com/redhatinc


twitter.com/RedHat



Red Hat