Обзор на Openshift 4

Не забудь включить запись!

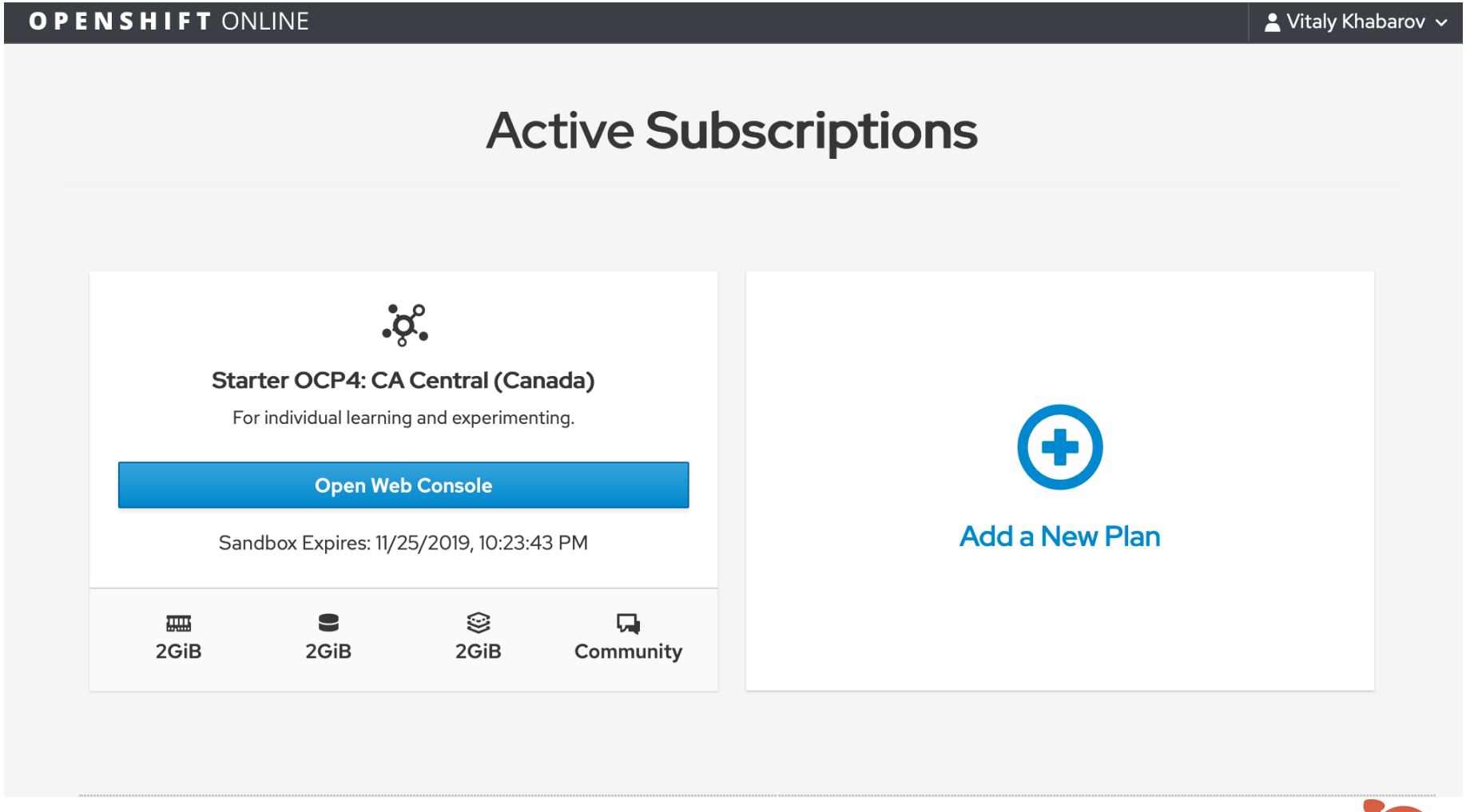
План

- Чем же так хорош Openshift?
- Утилиты для работы с Openshift
- Архитектура Openshift
- Абстракции Openshift
- Дополнительные элементы Openshift
- Отличие Openshift от Kubernetes и что выбрать

Demo - New application

Openshift Online

https://manage.openshift.com/



Создадим инстанс MongoDB

```
$ oc new-app --name=mongodb --template=mongodb-ephemeral \
    -p MONGODB_USER=reddit \
    -p MONGODB_PASSWORD=reddit \
    -p MONGODB_DATABASE=testdb \
    -p MONGODB_ADMIN_PASSWORD=password

--> Deploying template "openshift/mongodb-ephemeral" to project reddit

...

--> Creating resources ...
    secret "mongodb" created
    service "mongodb" created
    persistentvolumeclaim "mongodb" created
    deploymentconfig "mongodb" created
--> Success
    Application is not exposed. You can expose services to the outside world by
executing one or more of the commands below:
     'oc expose svc/mongodb'
    Run 'oc status' to view your app.
```

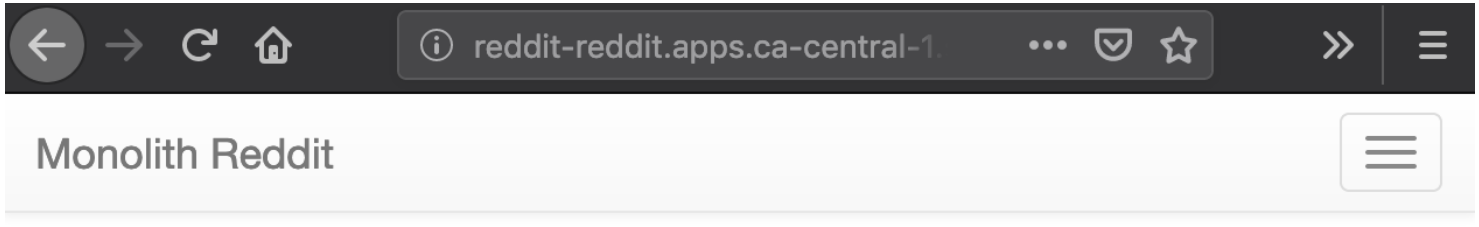
Задеплоим наше приложение

```
oc new-app ruby:2.4~https://github.com/vitkhab/reddit \
    -e DATABASE_URL=mongodb:27017 \
    -e DATABASE_USER=reddit \
    -e DATABASE_PASS=reddit \
    -e DATABASE_NAME=testdb

--> Found image fbe4fe3 (2 weeks old) in image stream "openshift/ruby" under tag "2.4"
for "ruby:2.4"

...

--> Creating resources ...
    imagestream "reddit" created
    buildconfig "reddit" created
    deploymentconfig "reddit" created
    service "reddit" created
--> Success
    Build scheduled, use 'oc logs -f bc/reddit' to track its progress.
    Application is not exposed. You can expose services to the outside world by
executing one or more of the commands below:
     'oc expose svc/reddit'
    Run 'oc status' to view your app.
```

```
$ oc expose svc/reddit

route "reddit" exposed
```

Подождем, пока деплой закончится

```
$ oc status
In project reddit on server https://api.ca-central-1.starter.openshift-online.com:6443

svc/mongodb - 172.30.114.186:27017
  dc/mongodb deploys openshift/mongodb:3.6
    deployment #1 running for 36 seconds - 0/1 pods

http://reddit-reddit.apps.ca-central-1.starter.openshift-online.com to pod port 8080-tcp
(svc/reddit)
  dc/reddit deploys istag/reddit:latest <-
    bc/reddit source builds https://github.com/vitkhab/reddit.git on openshift/ruby:2.4
      build #1 running for 16 seconds - 619ce58: Update app.rb (Vitaly Khabarov
<vitkhab@users.noreply.github.com>)
    deployment #1 waiting on image or update


3 infos identified, use 'oc status -v' to see details.

$ oc get routes
NAME      HOST/PORT                                                    PATH
SERVICES   PORT       TERMINATION   WILDCARD
reddit    reddit-reddit.apps.ca-central-1.starter.openshift-online.com
reddit     8080-tcp                 None
```



Menu

All posts

New post

Project: reddit

Add

reddit-1 > Pod Details

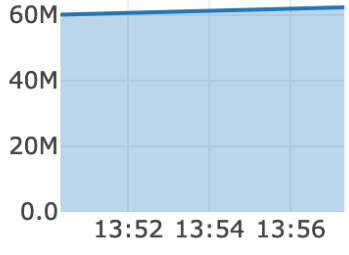
P reddit-1-wsbnl

Actions

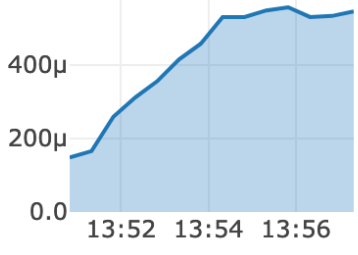
Overview | YAML | Environment | Logs | Events | Terminal

Pod Overview

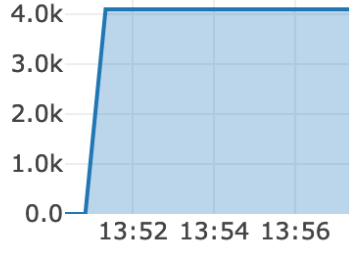
Memory Usage



CPU Usage



Filesystem



NAME
reddit-1-wsbnl

STATUS
Running

Почистим за собой

```
$ oc delete all -l app=reddit

pod "reddit-1-45f8t" deleted
replicationcontroller "reddit-1" deleted
service "reddit" deleted
deploymentconfig.apps.openshift.io "reddit" deleted
buildconfig.build.openshift.io "reddit" deleted
imagestream.image.openshift.io "reddit" deleted
route.route.openshift.io "reddit" deleted

$ oc delete all -l app=mongodb

pod "mongodb-1-z6pff" deleted
replicationcontroller "mongodb-1" deleted
service "mongodb" deleted
deploymentconfig.apps.openshift.io "mongodb" deleted

$ oc delete secret mongodb

secret "mongodb" deleted

$ oc delete pvc mongodb

persistentvolumeclaim "mongodb" deleted
```

End Demo - New application

Утилиты для работы с Openshift

kubectl vs oc

```
brew upgrade openshift-cli
```

- `oc` умеет самонастраиваться, вместо вбивания четырех команд в `kubectl`
- `oc new-app` - киллер фича
- `oc adm` - если хотите поадминить
- `odo` - для разработчиков. Больше в репозитории проекта

Устаревшие методы

- Minishift работает только с 3.X
- `oc cluster up`. Работает только с 3.X. И не нужен, так в Docker for Mac сломан функционал `Bypass proxy`
- Это все про OKD

CodeReady Containers

- Работает в одной VM, которая master и worker нода
- Операторы мониторинга и machine-config выключены. Соответствующие разделы UI не работают
- По причине выше нет возможности плавного обновления на новую версию Openshift
- Польностью Stateless/Ephemeral

Getting Started

Требования

- 4 virtual CPUs (vCPUs)
- 8 GB of memory
- 35 GB of storage space

Работа с CRC

```
# Настройка локальной машины
$ crc setup
# Запуск VM
$ crc start
# Запуск WebUI
$ crc console
# Вывод команды для настройки oc
$ crc oc-env
# Оставновить VM
$ crc stop
# Удалить VM
$ crc delete
```

Редакции Openshift

Бесплатные и без поддержки

- Origin Community Distribution of Kubernetes (OKD)
  - OKD v3.11 и Kubernetes v.1.11
  - OKD v4.X ожидает релиза Fedora CoreOS в конце 2019
  - Есть сборки CI https://origin-release.svc.ci.openshift.org/
  - Дополнительные абстракции и безопасность по отношению к Kubernetes
  - Отсутствует логирование и мониторинг по отношению к OCP
  - Поддержка от сообщества. Отсутствует какая-либо поддержка RedHat
  - Только Opensource. Никаких RHEL и RHCOS

Openshift as a Service

- Openshift Online
  - Ресурсы в облаке RedHat
  - Starter - бесплатный с кучей ограничений
  - Pro - платный, имеет поддержку функций OCP
- Openshift Dedicated
  - Приватный full managed кластер
  - Уже на AWS, в 2020 в GCP

Hosted Openshift

- Openshift Container Engine
  - Урезанная версия
- Openshift Container Platform
  - Advanced cluster management. Логгирование, Kiali и Chargeback.
  - Advanced networking. Istio, Multi-tenant SDN, Ingress на нестандартные порты.
  - Developer experience. Developer console, CI/CD пайплайны, утилита odo, Source-to-Image Container factory.

Поддержка RedHat

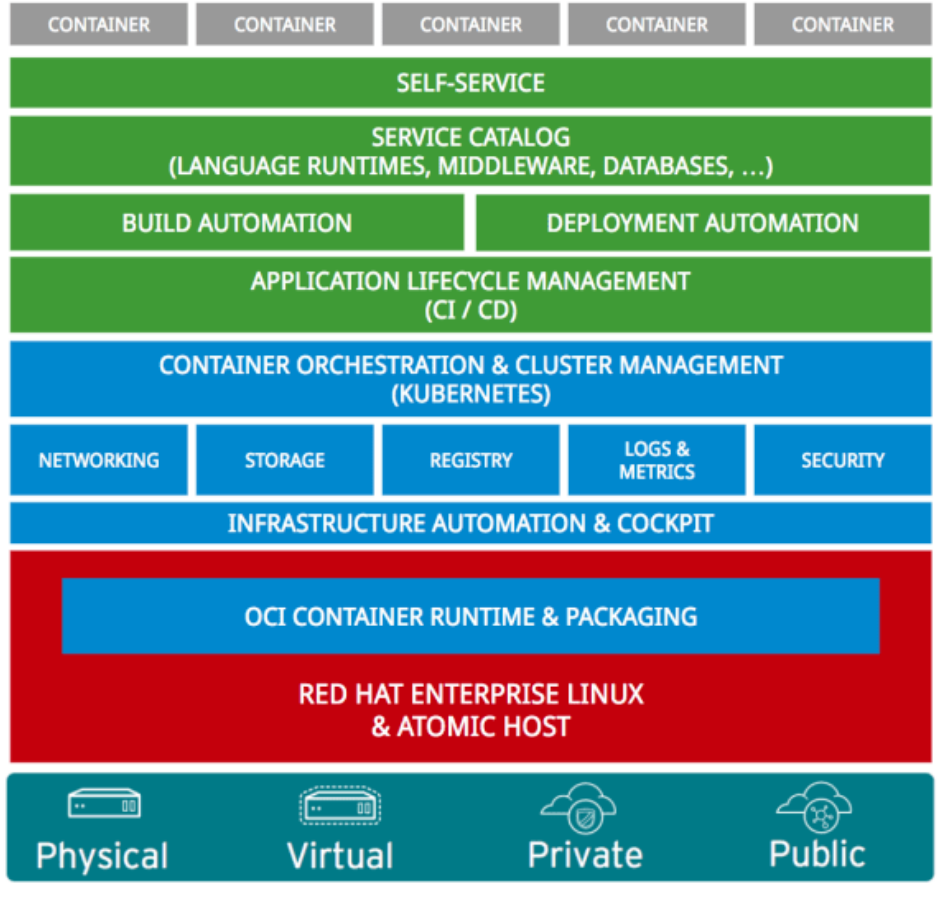
- Есть Scope of Coverage и Service Level Agreement
- Full support - RHSAs, RHBAs, RHEAs
- Maintenance support - RHSAs, RHBAs
- Extended update support - Продленный Maintenance support
- По подписке
- При расширении кластера нужно доплачивать
- Подробнее
  https://access.redhat.com/support/policy/updates/openshift

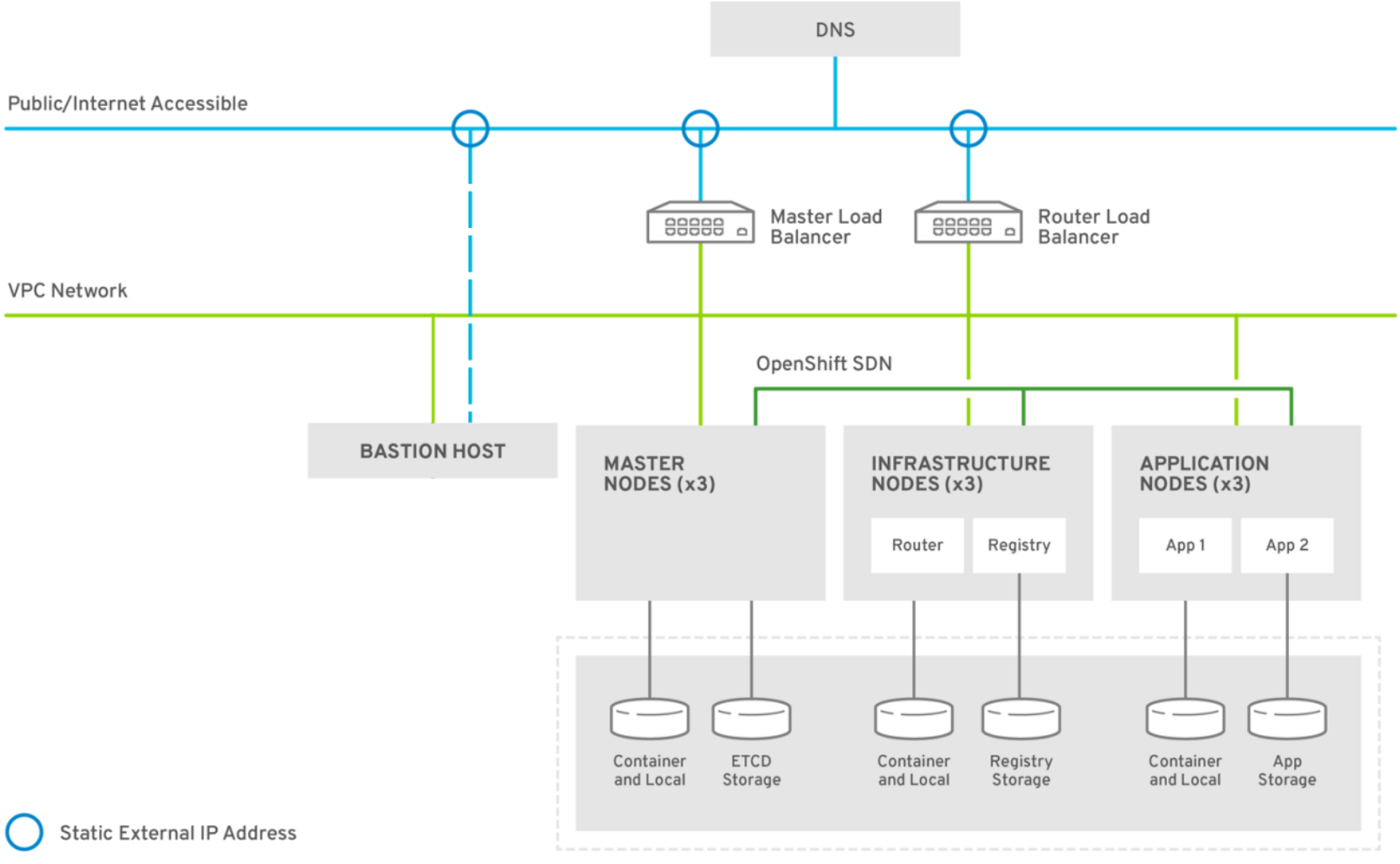
Архитектура Openshift

OpenShift = Enterprise Kubernetes+

Build, Deploy and Manage Containerized Apps

RED HAT®
OPENSHIFT




DNS
Public/Internet Accessible
Master Load Balancer
Router Load Balancer
VPC Network
OpenShift SDN
BASTION HOST
MASTER NODES (x3)
INFRASTRUCTURE NODES (x3)
Router
Registry
APPLICATION NODES (x3)
App 1
App 2
Container and Local
ETCD Storage
Container and Local
Registry Storage
Container and Local
App Storage
Static External IP Address

Инфра ноды

- На них запускаются инфраструктурные компоненты:
  - Router'ы
  - Container Image Registry
  - Cluster metrics collection или monitoring service
  - Cluster aggregated logging
- В принципе, это выделенные Worker ноды со специальными лейблами

Соответствие версий

- Openshift 4.0 <-> Kubernetes 1.12
- Openshift 4.1 <-> Kubernetes 1.13
- Openshift 4.2 <-> Kubernetes 1.14
- Openshift 4.3 <-> Kubernetes 1.16
- План релизов - раз в квартал с отставанием от K8s на одну версию

Demo - Install

Подготовка

- Service Account с нужными ролями
- DNS зона
- Подготовленная инфраструктура или поддерживаемое облако

```
./init.sh

export GOOGLE_CLOUD_KEYFILE_JSON=/Users/vitkhab/Documents/git/express42-
lab/scenario04/openshift-installer.json
```

Скачиваем инсталлер

```
$ wget http://mirror.openshift.com/pub/openshift-v4/clients/ocp/4.2.0/openshift-install-mac-4.2.0.tar.gz
$ mkdir ./oi
$ tar xf openshift-install-mac-4.2.0.tar.gz -C oi
$ rm openshift-install-mac-4.2.0.tar.gz
```

Быстрая установка

```
$ ./oi/openshift-install create cluster

? SSH Public Key /Users/vitkhab/.ssh/id_rsa.pub
? Platform gcp
? Project ID openshift-example-e42
? Region europe-west1
? Base Domain openshift.express42.io
? Cluster Name os-e42
? Pull Secret [? for help]

INFO Creating infrastructure resources...
...
```

Установка с дополнительными настройками

```
$ ./oi/openshift-install create install-config --dir=gcp

? SSH Public Key /Users/vitkhab/.ssh/id_rsa.pub
? Platform gcp
? Project ID openshift-example-e42
? Region europe-west1
? Base Domain openshift.express42.io
? Cluster Name os-e42
? Pull Secret [? for help]
```

Установка с дополнительными настройками

./gcp/install-config.yaml

```
compute:
- hyperthreading: Enabled
  name: worker
  platform:
    gcp:
      type: n1-standard-2
  replicas: 2
```

```
$ ./oi/openshift-install create manifests --dir=gcp
```

Установка с дополнительными настройками

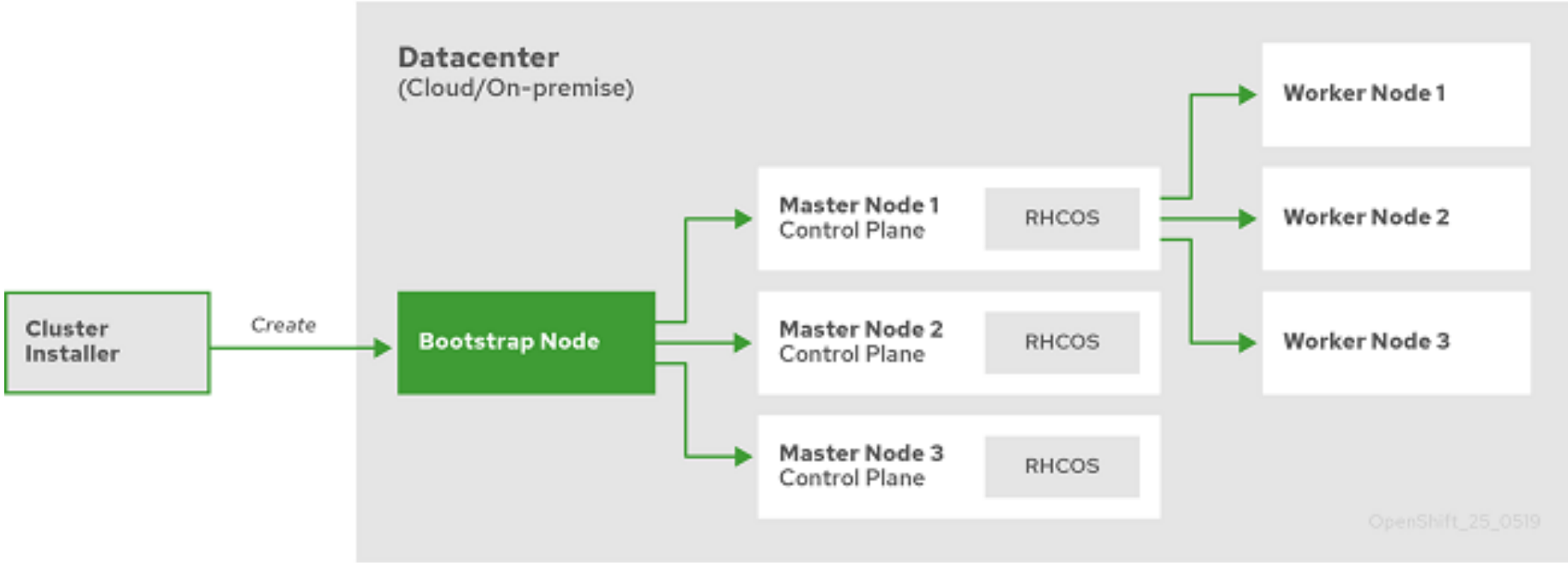
```
$ sed -e 's/        sizeGb: 128/        sizeGb: 64/g' -i "" ./gcp/openshift/*

$ ./oi/openshift-install create ignition-configs --dir=gcp
$ ./oi/openshift-install create cluster --dir=gcp
```

Чистка кластера

```
./oi/openshift-install destroy cluster --dir=gcp
```

End Demo - Install

Установка


Datacenter
(Cloud/On-premise)
Cluster Installer
Create
Bootstrap Node
Master Node 1
Control Plane
RHCOS
Master Node 2
Control Plane
RHCOS
Master Node 3
Control Plane
RHCOS
Worker Node 1
Worker Node 2
Worker Node 3
OpenShift_25_0519

Минимум

- Один bootstrap сервер
- Три master ноды, они же control plane
- Не менее двух worker ноды

Требования к ресурсам

Тип	Operating System	vCPU	RAM	Storage
Bootstrap	RHCOS	4	16 GB	120 GB
Control plane	RHCOS	4	16 GB	120 GB
Compute	RHCOS or RHEL 7.6	2	8 GB	120 GB

Поддерживаемые платформы

	Installer Provisioned Infrastructure (IPI)	User Provisioned Infrastructure (UPI)
OPENSHIFT by Red Hat 4.0	amazon web services	—
OPENSHIFT by Red Hat 4.1	—	amazon web services, vmware vSphere, Baremetal
OPENSHIFT by Red Hat 4.2	Microsoft Azure, RED HAT OPENSTACK PLATFORM, Google Cloud Platform, Baremetal On RHHI**	—
OPENSHIFT by Red Hat 4.3	Alibaba Cloud	Microsoft Azure, Google Cloud Platform, RED HAT OPENSTACK PLATFORM

*** On qualified hardware stack*

Как устанавливать

- 4.X - openshift-install
  - http://mirror.openshift.com/pub/openshift-v4/clients/ocp/
  - http://mirror.openshift.com/pub/openshift-v4/clients/ocp-dev-preview/

Установка на Bare metal

- Persistent storage. Для private image registry.
- Сетевая доступность к Red Hat OpenShift Cluster Manager и Quay.io
- Сервера и сеть. Балансировщики, DHCP и настроенная связь между серверами
- Удостоверяющий центр который сможет автоматически подписать запросы кластера
- Заморочиться с Ignition конфигурацией (аналог cloud-init) и манифестами

Полный гайд

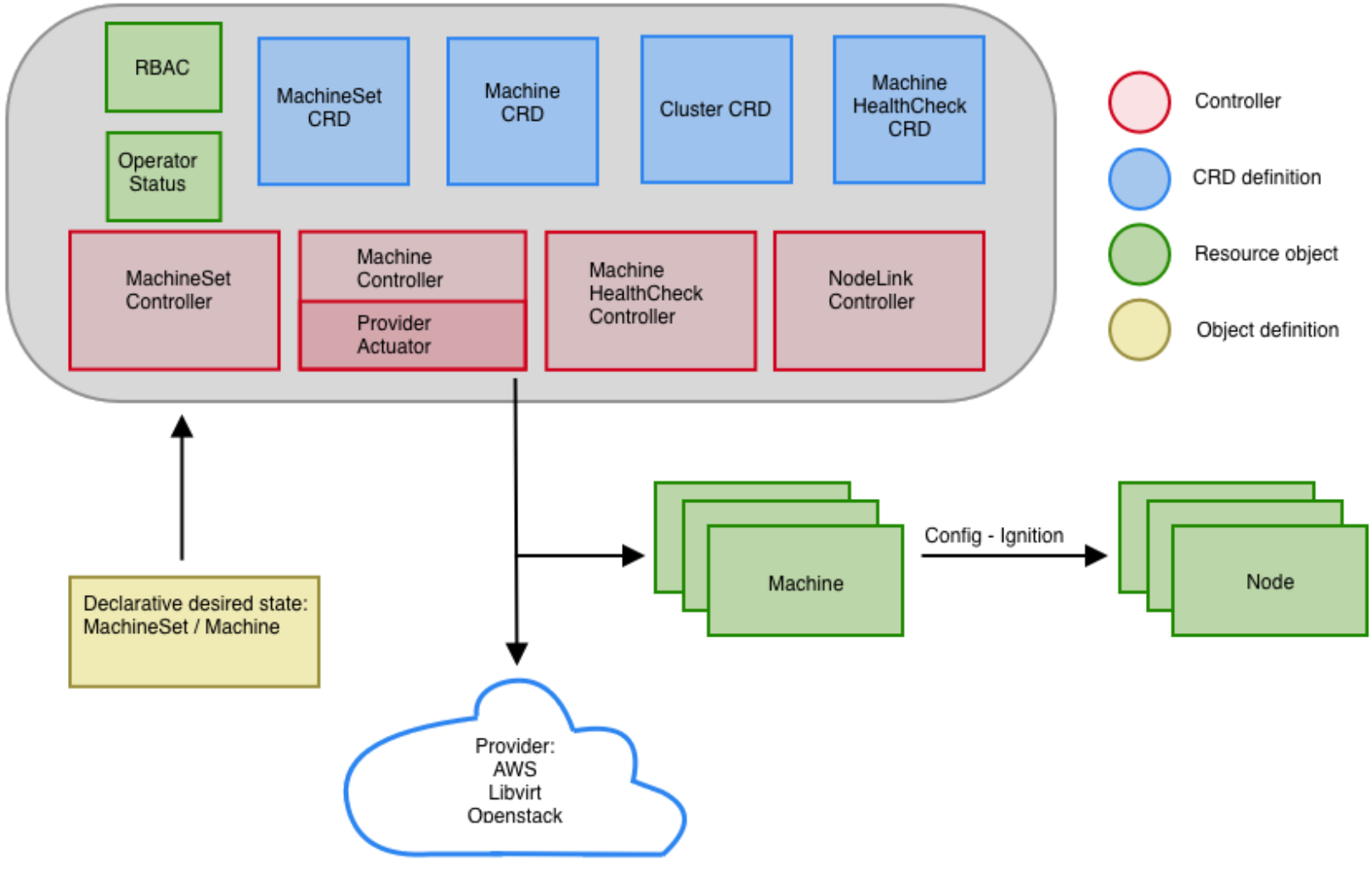
Как расширять

- Machines и MachineSet и MachineHealthCheck
- MachineAutoscaler и ClusterAutoscaler

```
spec:
  providerSpec:
    value:
      apiVersion: gcpprovider.openshift.io/v1beta1
      canIPForward: false
      credentialsSecret:
        name: gcp-cloud-credentials
      deletionProtection: false
      disks: ...
      kind: GCPMachineProviderSpec
      machineType: n1-standard-4
      networkInterfaces: ...
      projectID: <project_name>
      region: us-central1
      serviceAccounts: ...
      zone: us-central1-a
```

Документация по GCP

Machine API Operator



RHEL

- RHEL немного сбоку. Вы сами отвечаете за их поддержку.
- Нужно подготовить машины, подключить подписку, оставить только нужные yum-репозитории.
- RHEL настраивается по старинке Ansible'ом.
- И нужно подписать сертификаты.

Документация по RHEL

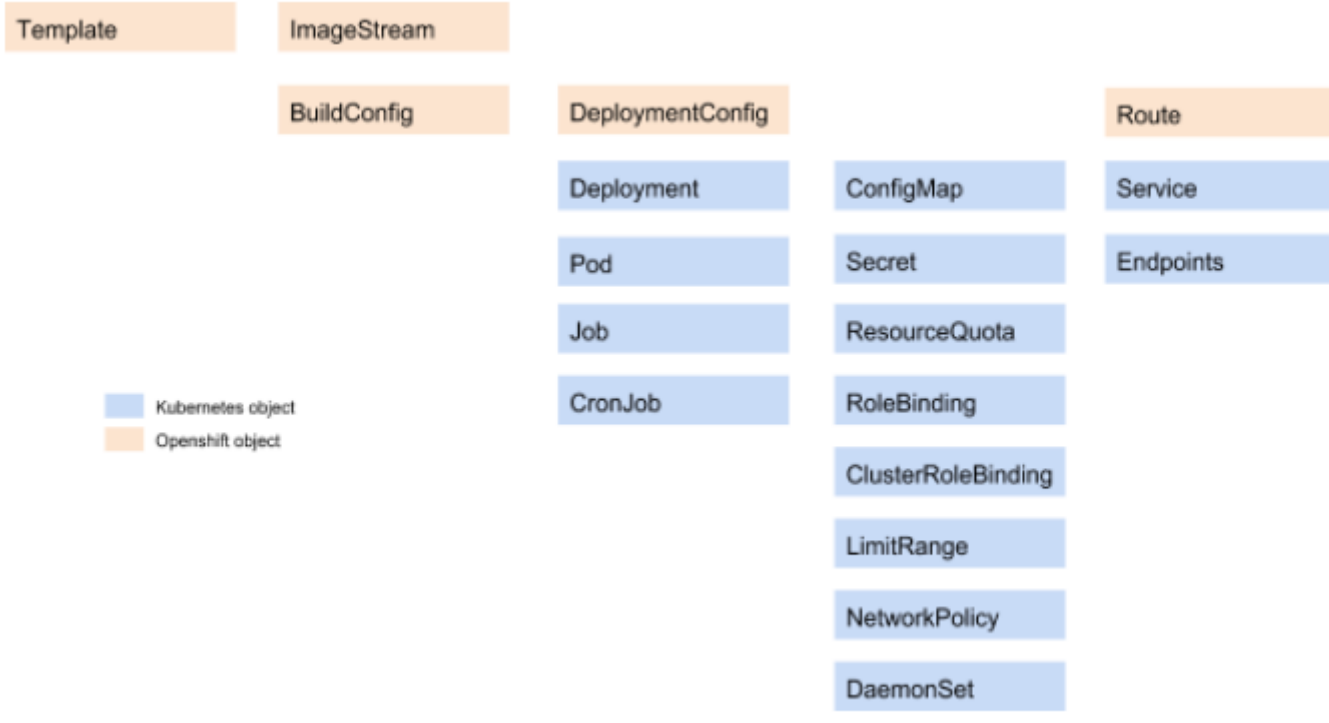
Безопасность в Openshift

- Политики Openshift по умолчанию более строгие
- По умолчанию включены Security Context Constraints
- Openshift умеет интегрироваться с AD из коробки
- RBAC по-умолчанию

OAuth

- Механизм реализуемый oauth proxy в sidecar.
- Позволяет аутентифицироваться во все компоненты запущенные в Openshift под одной учетной записью.
- SSO в в инфраструктурные сервисы Jenkins, Prometheus, Grafana, ElasticSearch и так далее.

Абстракции Openshift


Template
ImageStream
BuildConfig
DeploymentConfig
Route
Deployment
ConfigMap
Service
Pod
Secret
Endpoints
Job
ResourceQuota
CronJob
RoleBinding
ClusterRoleBinding
LimitRange
NetworkPolicy
DaemonSet
Kubernetes object
Openshift object

Templates vs Helm

- Templates очень простые и не поддерживают версионирования
- Для полноценной работы с Template'ами нужны wrapper, которые будут обеспечивать дополнительную функциональность и логику
- Для работы с Helm'ом нужно настраивать права Tiller
- Red Hat у себя в блоге тоже пишут про Helm

```yaml
apiVersion: template.openshift.io/v1
kind: Template
metadata:
  name: project-request
objects:
- apiVersion: project.openshift.io/v1
  kind: Project
  metadata:
    annotations:
      openshift.io/description: ${PROJECT_DESCRIPTION}
      openshift.io/display-name: ${PROJECT_DISPLAYNAME}
      openshift.io/requester: ${PROJECT_REQUESTING_USER}
    creationTimestamp: null
    name: ${PROJECT_NAME}
  spec: {}
  status: {}
- apiVersion: rbac.authorization.k8s.io/v1
  kind: RoleBinding

  ...

parameters:
- name: PROJECT_NAME
- name: PROJECT_DISPLAYNAME
- name: PROJECT_DESCRIPTION
- name: PROJECT_ADMIN_USER
- name: PROJECT_REQUESTING_USER
```

Projects

- Проект = Namespace + дополнительная функциональность.
- Описание и Display name
- Создаются из шаблона и имеют набор элементов по умолчанию.
- Можно добавить RoleBinding, сетевые политики и квоты по умолчанию

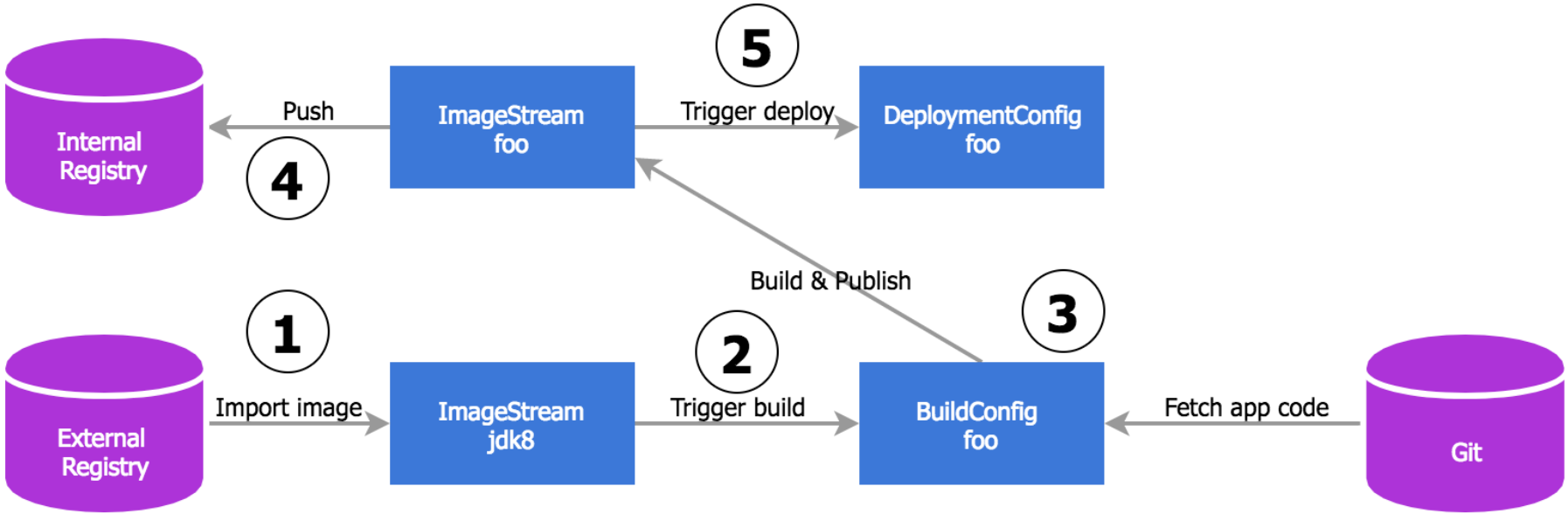
Builds

- Docker build
  - Dockerfile и контекст
- Source-to-Image (S2I) build
  - Закидывает код внутрь подготовленных контейнеров
  - Запускает buildah
  - Умеет инкрементальные сборки
- Custom build
  - Можно использовать Custom образ для сборки котрый сам будет руководить сборкой приложения
  - RedHat предупреждает, не раздавайте права на Custom build налево и направо

- Pipeline build

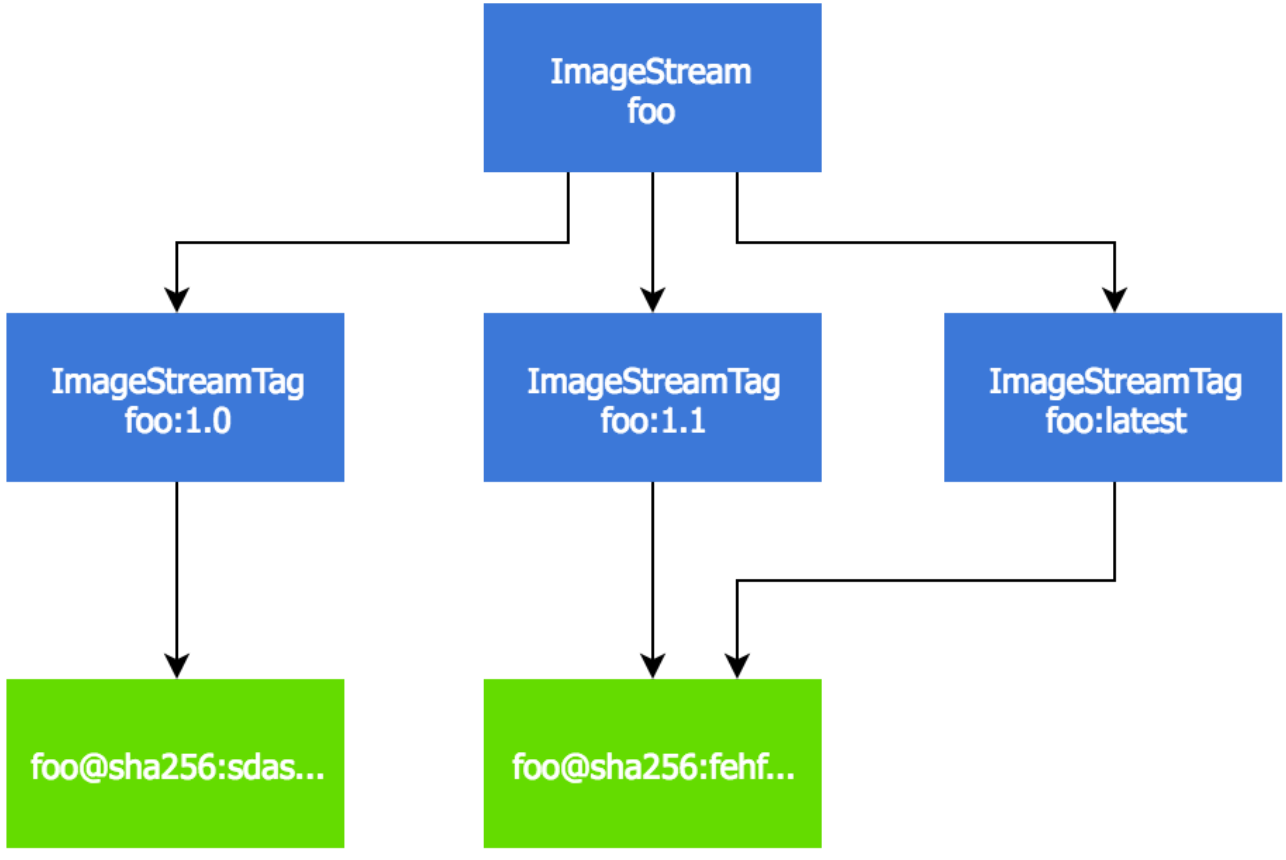
Triggers

- Можно использовать для BuildConfig и DeploymentConfig.
- Вы можете всегда держать актуальным Прод на теге 'latest'
- Или пересобирать образа приложений при изменении базовых образов
- Возможно все становится не так прозрачно с Trigger'ами



ImageStreams

- Коллекция референсов на реальные образа. Можно представить в виде директории с символьными ссылкаи
- ImageStreamTags указывают на конкретные образа по SHA256 идентификатору
- Можно менять образа не меняя манифесты DeploymentConfig
- Можно управлять тем, какой релиз приложения куда будет задеплоен (Dev, Test, Prod)
- Можно повесить триггер на изменение Imagestream
- Можно ссылаться на внешние и внутренние образа



```
kind: ImageStream
apiVersion: image.openshift.io/v1
metadata: ...
spec:
  lookupPolicy:
    local: false
status:
  dockerImageRepository: 'image-registry.openshift-image-
registry.svc:5000/default/reddit'
  tags:
    - tag: latest
      items:
        - created: '2019-10-23T21:28:43Z'
          dockerImageReference: >-
            image-registry.openshift-image-
registry.svc:5000/default/reddit@sha256:ce2f2f34ebb9855e48171a7aeeebcdb87602b8a83c9148b7
c91cd9632ad6dcda
          image: >-
            sha256:ce2f2f34ebb9855e48171a7aeeebcdb87602b8a83c9148b7c91cd9632ad6dcda
          generation: 1
```

BuildConfig

- Отвечает за:
  - На что стриггериться
  - Откуда взять исходный код
  - Как его собрать
  - Куда положить
  - Кого оповестить
- Отлично работает с ImageStream
- По умолчанию работает с внутренним реестром

```
kind: BuildConfig
apiVersion: build.openshift.io/v1
metadata: ...
spec:
  runPolicy: "Serial"
  triggers:
    - type: "GitHub"
      github:
        secret: "secret101"
    - type: "Generic"
      generic:
        secret: "secret101"
    - type: "ImageChange"
  source:
    git:
      uri: "https://github.com/openshift/ruby-hello-world"
  strategy:
    sourceStrategy:
      from:
        kind: "ImageStreamTag"
        name: "ruby-20-centos7:latest"
  output:
    to:
      kind: "ImageStreamTag"
      name: "origin-ruby-sample:latest"
  postCommit:
      script: "bundle exec rake test"
```

DeploymentConfig

- DeploymentConfig реализованы отдельным оператором который следит обеспечивает всю подготовку и выкатку приложения
- Зачем?
  - Длительные запросы надо аккуратно завершить
  - Миграции базы данных должны быть синхронизированы с обновлением приложения
  - Внешние зависимости тоже надо обрабатывать
- Через стандартные InitContainers сложно управлять выкаткой множества подов

Стратегии

- Rolling strategy
  - Отрабатываем `pre` хуки. Создаем новые поды до размера `maxSurge`, опускаем старые поды до размера `maxUnavailable`. Отрабатывам `post` хуки
- Recreate strategy
  - Отрабатываем `pre` хуки. Удаляем все старые поды. Отрабатывам `mid` хуки. Запускаем новые поды. Отрабатывам `post` хуки
- Custom strategy
  - Указываем образ который реализует всю логику выкатки

```yaml
kind: DeploymentConfig
apiVersion: apps.openshift.io/v1
metadata: ...
spec:
  strategy:
    type: Rolling
    rollingParams:
      updatePeriodSeconds: 1
      intervalSeconds: 1
      timeoutSeconds: 600
      maxUnavailable: 25%
      maxSurge: 25%
    resources: {}
    activeDeadlineSeconds: 21600
  triggers: ...
  selector: ...
  template: ...
```

Routes vs Ingress

- Ingress Controller. Openshift поддерживает Routes и Ingress
- Routes реализованы на HAProxy и могут быть заменены хардварными F5 BIG-IP
- Routes менее гибкие и функциональные чем Ingress
- Routes возможно более зрелые, за счет интеграций и узкой специализации
- Routes можно заставить работать на Инфра нодах
- `oc expose` - наше все

```yaml
kind: Route
apiVersion: route.openshift.io/v1
metadata: ...
spec:
  host: reddit-default.apps.os-e42.openshift.express42.io
  subdomain: ''
  to:
    kind: Service
    name: reddit
    weight: 100
  port:
    targetPort: 8080-tcp
  wildcardPolicy: None
```

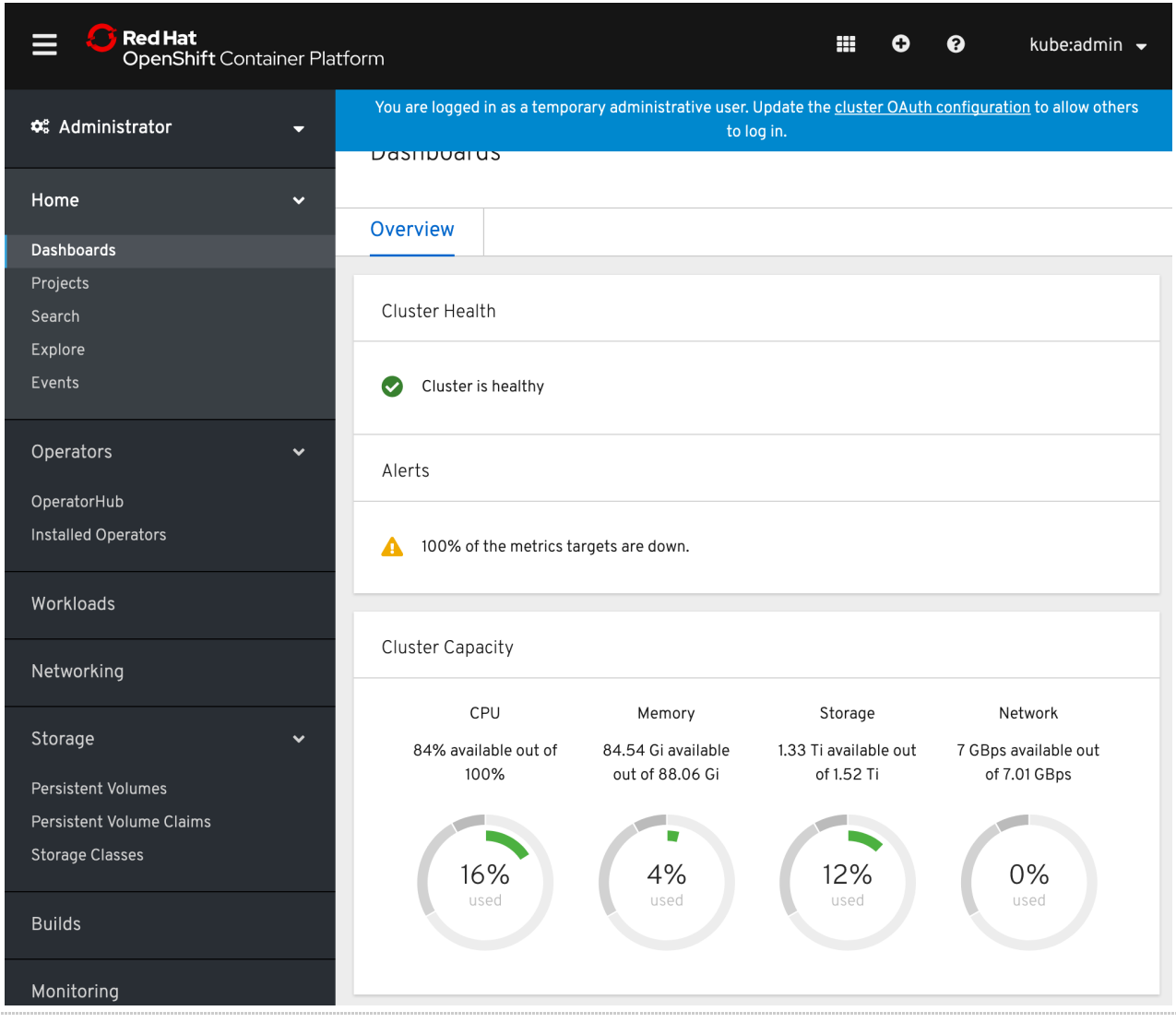
Application

- Не абстракция (в полном смысле слова). Реализована на template
- oc new–app или через WebUI
- Создать приложение можно из Git, Dockerfile, Docker образа, каталога, манифестами.

Документация

Demo - Openshift UI

Administrator



OperatorHub

Мониторинг

Логирование

Developer

- Topology
- New app

End Demo - Openshift UI

Openshift дополонительно

Операторы

- Оператор - first class citizen
- Operator SDK
- Operator Lifecycle Manager

Внутренний репо

- Умеет кешировать
- Используется по умолчанию
- Подходит для временного хранения

Service Mesh

- Нужен предустановленный Istio
- Визуализация через Kiali
- Трейсинг через Jaeger

Openshift 4.3

- Релиз в четвертом квартале 2019
- Metering for Services
- Windows containers

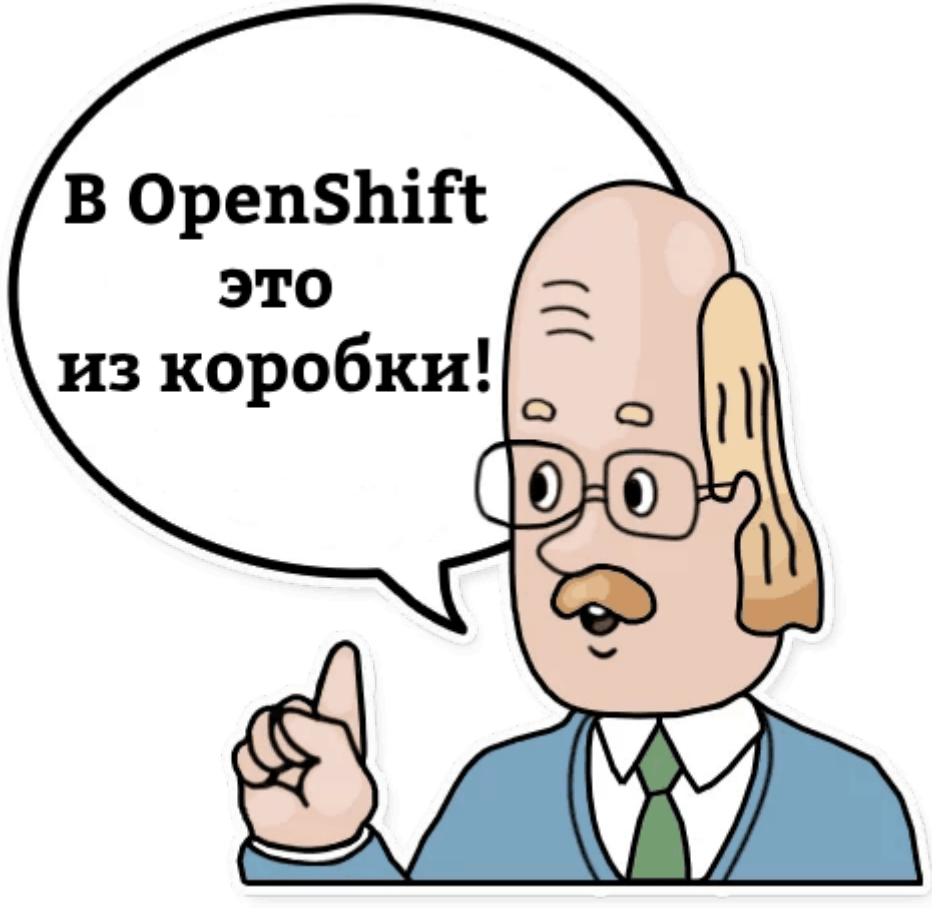
OpenShift Pipelines Operator

- Основан на Tekton pipelines
- Устанавливается через OperatorHub
- Планируется GA в Openshift 4.3

Serverless Operator

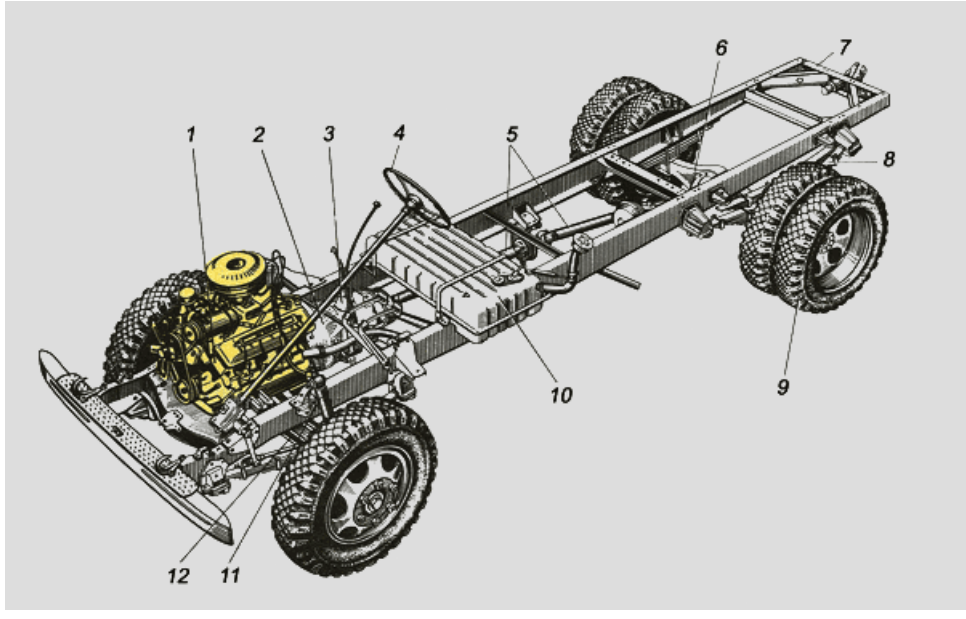
- На базе Knative
- Требует Istio
- Устанавливается через OperatorHub
- Планируется GA в Openshift 4.3

Openshift vs Kubernetes


В OpenShift
это
из коробки!

Сквозь времена

- В 1999 Linux vs Windows
- В 2009 Android vs iOS
- В 2019 Kubernetes vs Openshift



VS

Муки выбора

Если захотите попробовать

- Openshift online
- GCP, но следите за ресурсами, оно там дорого

Конец

- Оставляйте обратную связь в слайке и на сайте Отуса
- О том как прошла лекция, как вам материал, что можно сделать лучше