

Развертывание в k8s

Контейнерная оркестрация

Оркестраторы организуют расположение и координируют взаимосвязь инструментов в одном проекте, распределяют задачи между ними и контролируют их выполнение.

Оркестраторы удобны для управления микросервисными приложениями, поскольку каждый микросервис в кластере имеет свои данные и свою модель, он автономен с позиции разработки и развертывания и поэтому чаще всего помещается в отдельный контейнер. При большом числе сервисов увеличивается и число контейнеров, и этим всем нужно управлять.

Программы-оркестраторы управляют жизненными циклами контейнеров микросервисных приложений, помогают автоматизировать различные производственные процессы, поэтому команды DevOps интегрируют их в CI/CD. Посмотрим, какие задачи берут на себя оркестраторы.

- Подготовка инфраструктуры и развертывание
- Конфигурация и планирование
- Распределение ресурсов
- Масштабирование контейнеров на основе рабочих нагрузок
- Балансировка нагрузки
- Маршрутизация трафика
- Мониторинг состояния контейнеров
- Обеспечение безопасного взаимодействия между контейнерами

Kubernetes

Kubernetes — OpenSource-система для управления контейнерными кластерами. Появилась в результате наработок Google при использовании механизма для изоляции процессов в виртуальной среде (Borg). В 2014 г. Google открыла код Kubernetes и стала распространять систему под лицензией Apache 2.0.

Kubernetes выстраивает эффективную **систему распределения контейнеров по узлам кластера в ЗАВИСИМОСТИ** от текущей **нагрузки и имеющихся потребностей** в работе сервисов.

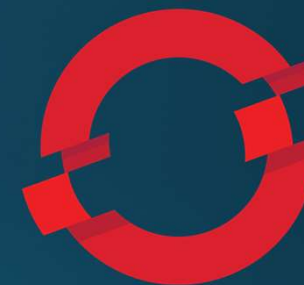
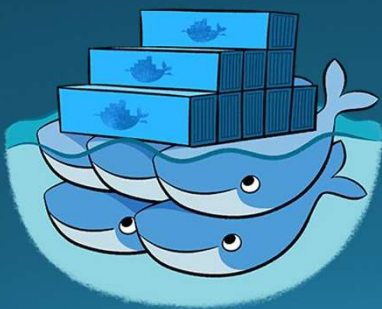
Эта платформа оркестрации способна обслуживать огромное количество хостов, запускать на них многочисленные контейнеры, отслеживать их состояние, контролировать совместную работу, проводить балансировку нагрузки и многое другое.



<https://kubernetes.io/>

Альтернативы Kubernetes'у

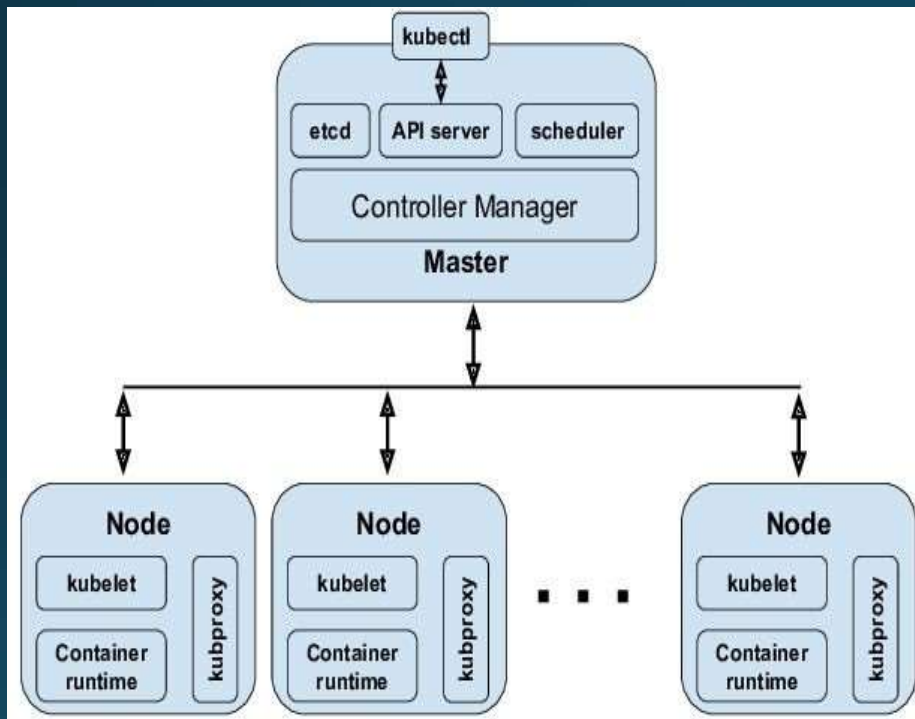
- OpenShift
- Docker Swarm
- Nomad
- Rancher
- SLURM
- ...



OPENSIFT



K8S: архитектура



Мастер-ноды:

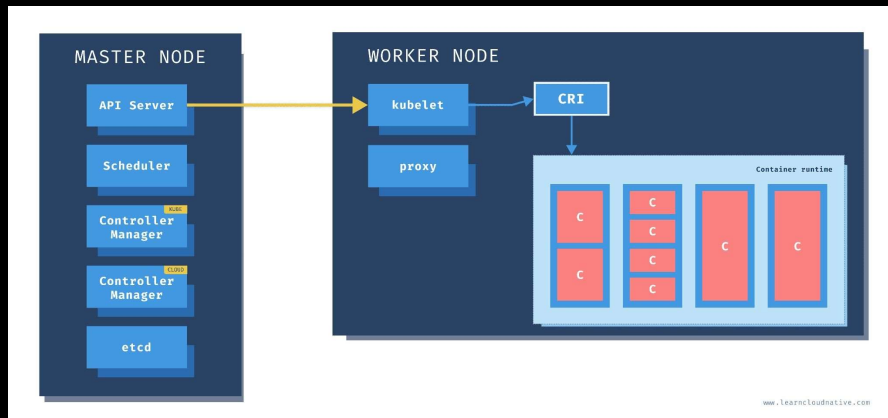
1. API server
2. Scheduler
3. Controller manager
4. etcd

Worker-ноды:

1. Container runtime
2. kubelet
3. kube-proxy

The differences between Docker, containerd, CRI-O and runc <https://www.tutorialworks.com/difference-docker-containerd-runc-crio-oci/>

Kubernetes flavors



	k0s	MicroK8s	kind	k3s + k3d	Minikube
Управление созданием/удалением узлов	X	X	✓	✓	✓
Система управления узлами	X	X	Docker	Docker	virtualbox, vmwarefusion, kvm2, vmware, none, docker, podman, ssh
Система запуска контейнеров	containerd	containerd	containerd, CRI-O	CRI-O	Docker, CRI-O, containerd
CNI по умолчанию	Calico	Calico	kindnet	Flannel	bridge
Возможность подключения ФС родительской ОС	HostPath	HostPath	HostPath + docker mount	HostPath + docker mount	HostPath + ... (зависит от системы виртуализации)
Наличие дополнений	X	✓	X	X	✓
Возможность создания кластера непривилегированным пользователем	X	X	✓	✓	✓
Vanilla Kubernetes	X	✓	✓	X	✓

Minikube

Сперва установить Docker и kubectl
(<https://kubernetes.io/ru/docs/tasks/tools/install-kubectl/#%D1%83%D1%81%D1%82%D0%B0%D0%BD%D0%BE%D0%B2%D0%BA%D0%B0-kubectl-%D0%B2-linux>), ПОТОМ

<https://kubernetes.io/ru/docs/tasks/tools/install-minikube/>



Стартуем!

```
minikube start --driver=none --kubernetes-  
version=1.22.12
```

Также может понадобится:

```
sudo apt install conntrack socat
```

```
minikube addons enable ingress
```

Компоненты K8S

Основные объекты Kubernetes
включают в себя:

- Pod
- Service
- Volume
- Namespace

... и их еще очень много: **Node, Ingress, ConfigMap, Secret, ServiceAccount, RoleBinding, ...**

`kubectl api-resources`

Абстракции более высокого уровня,
которые опираются на Controllers:

- Deployment
- DaemonSet
- StatefulSet
- ReplicaSet
- Job

API reference:

<https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.22/>

Что еще поставить?

- **Helm** - пакетный менеджер для k8s (<https://helm.sh/>)
- **k9s** - UI в командной строке (<https://github.com/derailed/k9s>)
- **kubens + kubectl** - удобное переключение контекста/namespace (<https://github.com/ahmetb/kubectx>)
- **NGINX Ingress Controller** (<https://www.nginx.com/products/nginx-ingress-controller/>)
- **cert-manager** - выпускаем и управляем сертификатами в k8s (<https://cert-manager.io/docs/>)
- **kube-prometheus-stack** - Prometheus + Grafana (с комплектом дашбордов) + всякое (<https://github.com/prometheus-community/helm-charts/tree/main/charts/kube-prometheus-stack>)



```
Context: minikube
Cluster: minikube
User: minikube
K9s Rev: v0.26.3
K8s Rev: v1.22.12
CPU: n/a
MEM: n/a
```

Pods(all)[16]

NAMESPACE↑	NAME	P
ingress-nginx	ingress-nginx-admission-patch-9bwtr	•
ingress-nginx	ingress-nginx-controller-f5476f687-sqlkp	•
kube-system	coredns-78fcd69978-wggtf	•
kube-system	etcd-ycloud	•
kube-system	kube-apiserver-ycloud	•
kube-system	kube-controller-manager-ycloud	•
kube-system	kube-proxy-pmpnf	•
kube-system	kube-scheduler-ycloud	•
kube-system	storage-provisioner	•
prometheus	alertmanager-prometheus-kube-prometheus-alertmanager-0	•
prometheus	prometheus-grafana-768876648c-hr4qw	•
prometheus	prometheus-kube-prometheus-operator-7499997946-mwfc8	•

<pod>

Kubernetes

Kubernetes (aka k8s) - открытая система для автоматического развертывания, масштабирования и управления контейнеризированными приложениями



K8s as a Service

K8s предоставляется как сервис многими облачными провайдерами:

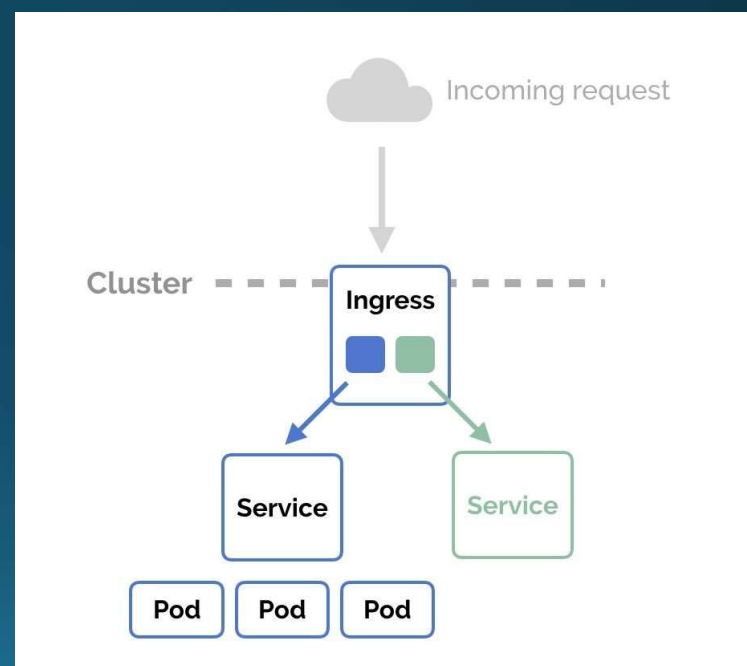
- Amazon Elastic Kubernetes Service (EKS)
- Azure Kubernetes Service (AKS)
- Google Kubernetes Engine (GKE)
- Yandex Manager Service for Kubernetes
- DigitalOcean managed Kubernetes
- Sber Cloud Manager Kubernetes
- Kubernetes в облаке Mail.ru

Основные преимущества

- Infrastructure as Code (IaC)
- кто-то уже настроил кластер за вас
- незаметное изменение состава узлов кластера
- простое масштабирование экземпляров приложения
- множество готовых для k8s приложений (helm)

Основные сущности

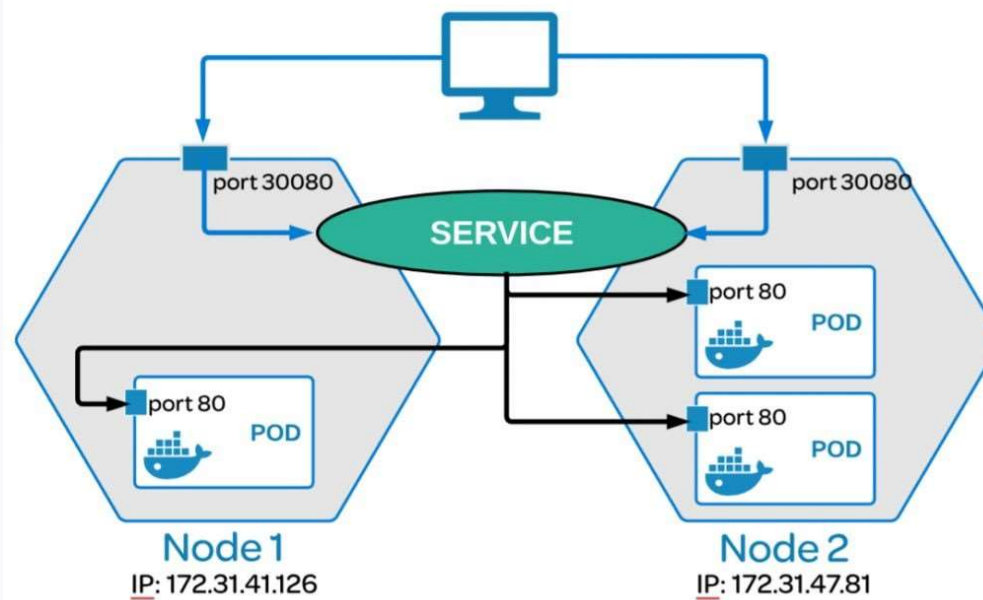
- Node - узел кластера
- Pod - контейнер¹
- ReplicaSet - набор идентичных Pods
- Deployment - управление Pods и ReplicaSets
- Service - публикация Pods как сетевого сервиса
- Ingress - маршрутизация внешних запросов к Service. Обычно HTTP



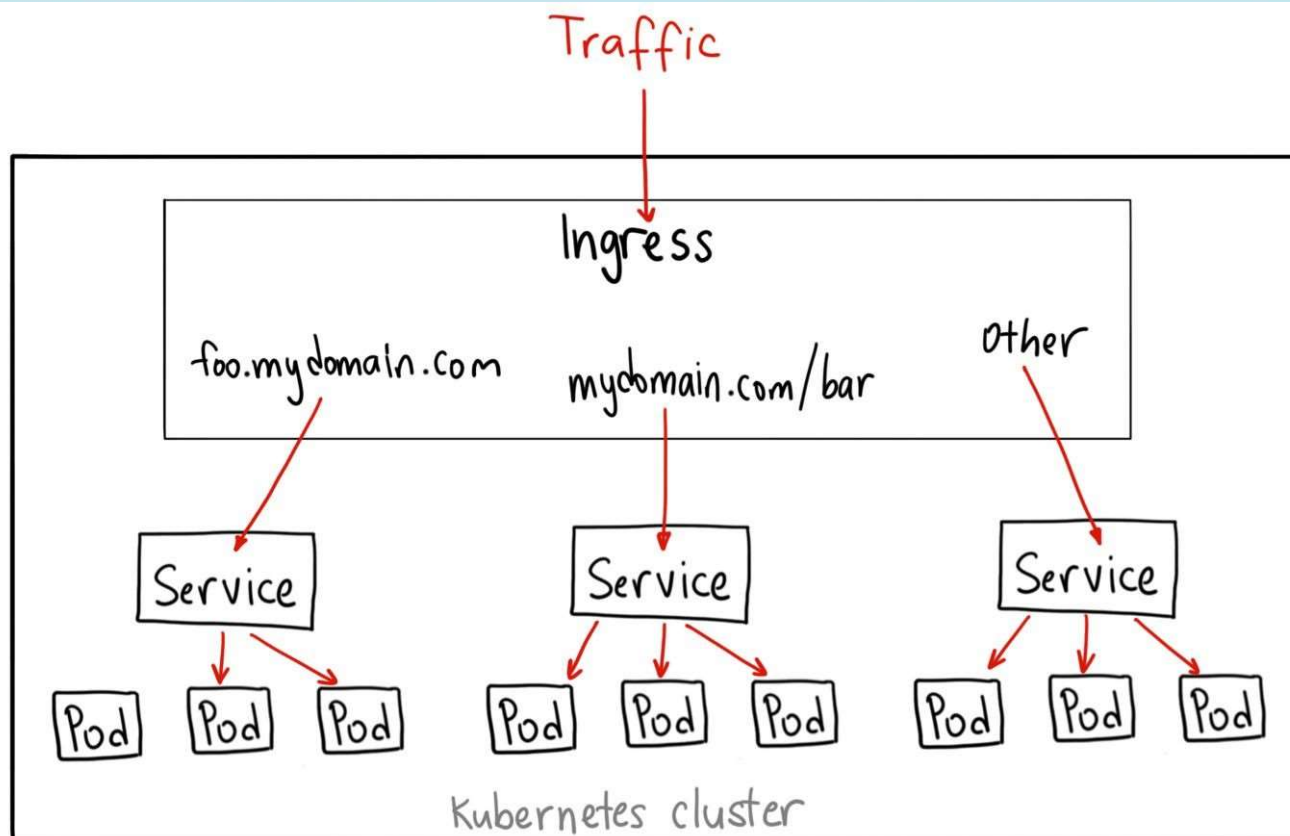
Service

Kubernetes Service

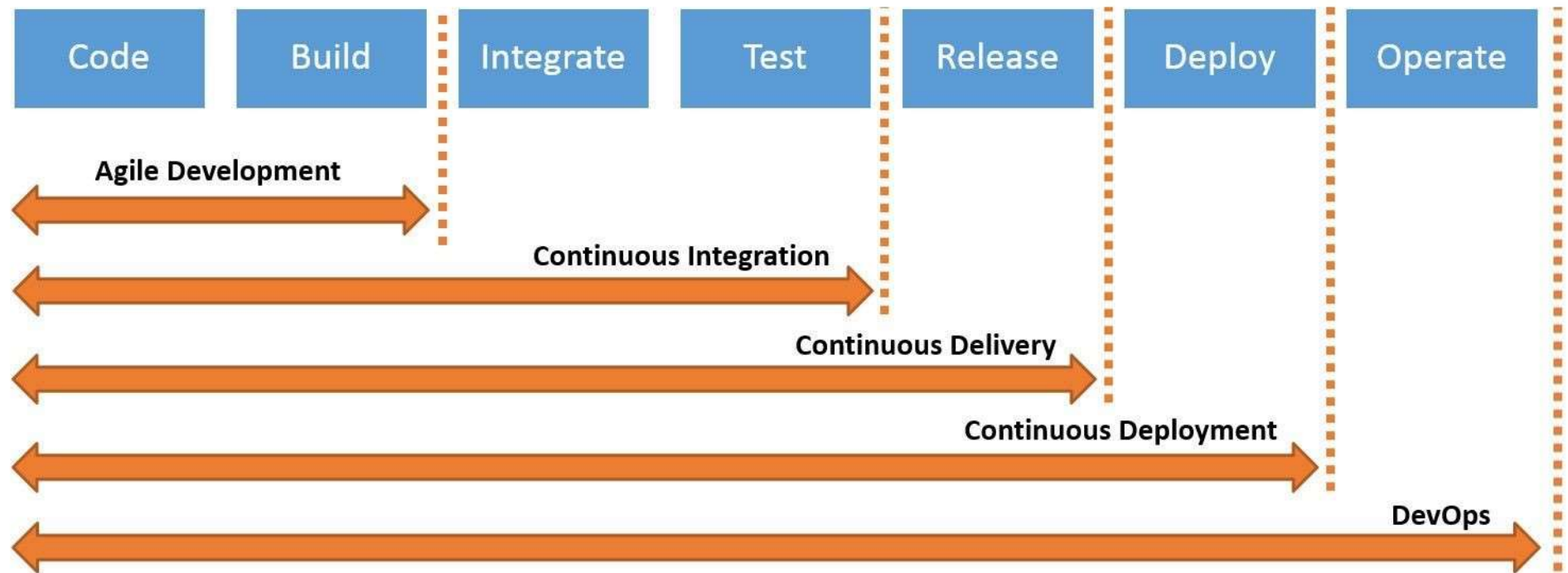
A service allows you to dynamically access a group of replica pods.



Ingress



Continuous Deployment



Стратегии приемочного тестирования

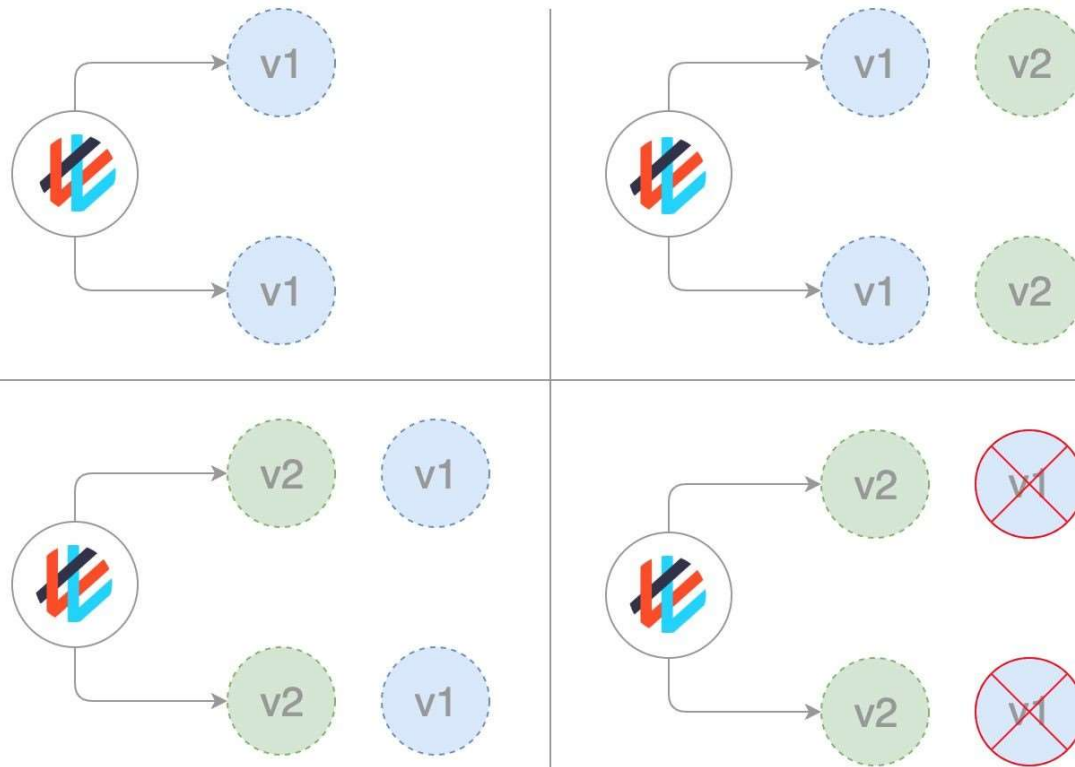
- Приемочное тестирование

- тестируем сразу в prod :)
- тестируем в prod под feature flags
- stage и prod окружения
 - в одном k8s кластера в разных namespaces
 - в разных k8s кластерах

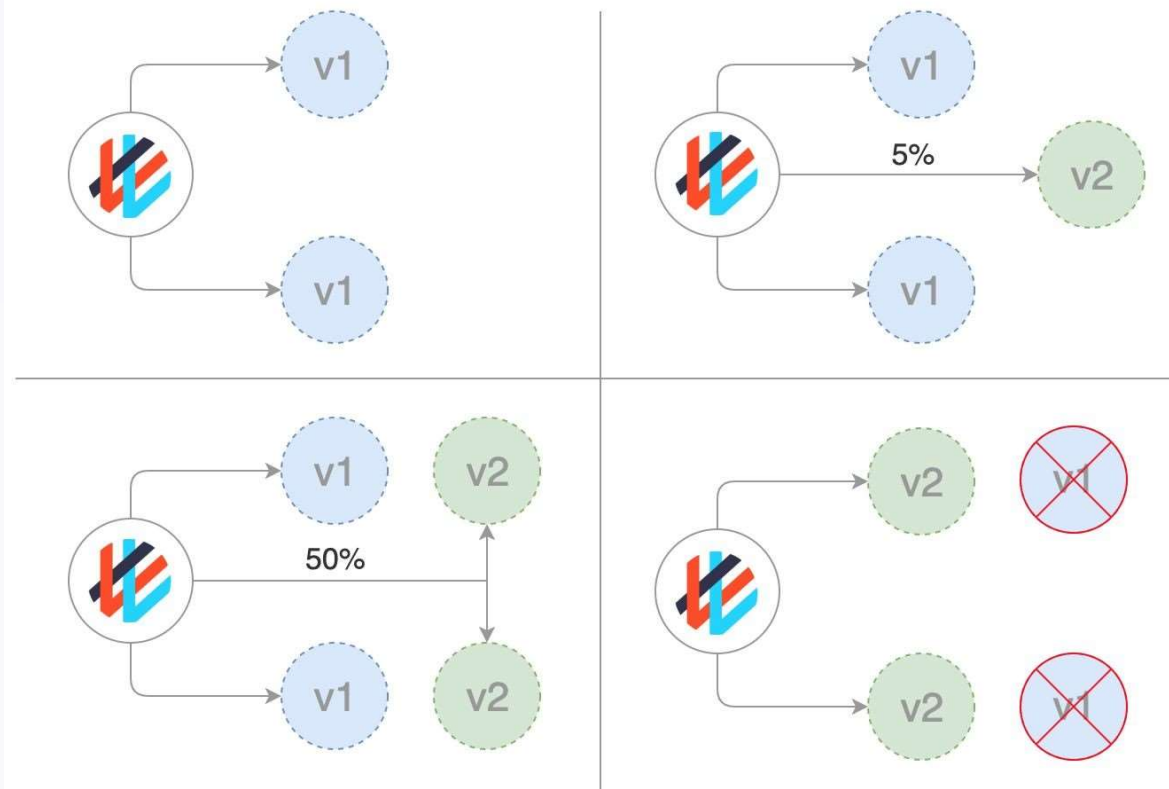
Стратегии развертывания

- Rolling - стандартная стратегия k8s
- Recreate - старые pods убиваются разом и создаются новые
- Blue/green - сразу работают старая и новая версии
- Canary - прогрессивный поэтапный подход

Blue green deployment



Canary deployment

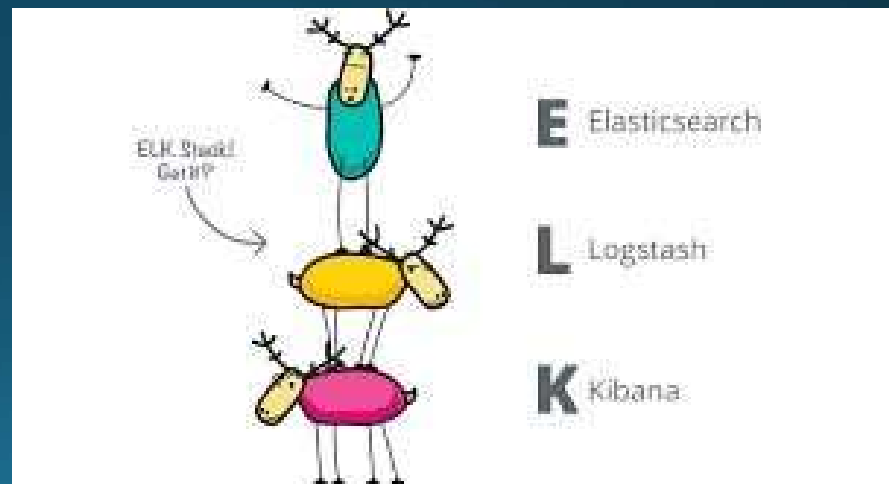


Логирование

- Главный способ поиска ошибок на prod
- Использовать инструменты языка. Например, `python.logging`
- Разделить уровни логирования и управлять ими
 - CRITICAL
 - ERROR
 - WARNING
 - INFO
 - DEBUG
- Для приложений в `docker` писать логи в `STDOUT` или `STDERR`
- Для обычных приложений писать в файл и предусмотреть ротацию

ELK Stack - Elastic Logstash Kibana

- Elasticsearch - СУБД с богатой функциональностью полнотекстового поиска
- Logstash - ETL для логов
- Kibana - web UI для Elasticsearch

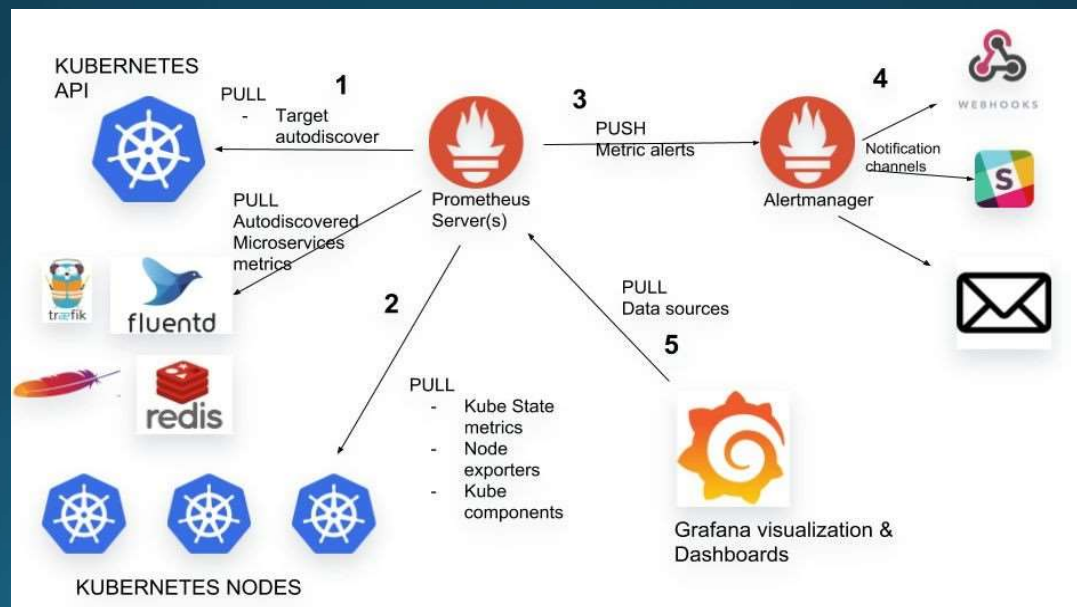


Мониторинг

- использование ресурсов (CPU, RAM, Disks, Network)
- метрики k8s кластера
- метрики приложений
 - доступны через HTTP. Например /metrics
 - позволяют заранее узнавать о проблемах
 - большое число ошибок и исключений
 - data drift
 - concept drift

Prometheus + Grafana + AlertManager

- Prometheus - сервер для сбора и хранения метрик
- Grafana - UI для визуализации метрик
- AlertManager - отправка оповещений при срабатывании триггеров по метрикам



Production код

- Код в системе контроля версий
- Есть документация по запуску и внесению изменений
- Общие части кода выделены в библиотеки
- Имеются модульные тесты
- Настроен CI и процесс Code Review
- Есть артефакты поставки
- Настроен CD для артефактов
- Описана стратегия приемочного тестирования
- Описана стратегия развертывания
- Настроен CD для развертывания
- На prod настроены логирование, мониторинг и алертинг
- Назначены дежурные

Менеджер пакетов для k8s

- объединяет запуск различных k8s компонентов
- репозиторий готовых charts

