Подходы к развертыванию

Не забудь включить запись!

Варианты развертывания

- Managed k8s
- Private/Public Cloud
- Bare metal

Managed

Преимущества:

- Самый быстрый и простой способ получить работающий кластер
- Поддержка от поставщика услуги
- Автомасштабирование из коробки

Недостатки:

- Версии компонентов могут отставать
- Необходимо ждать, пока внедрят новые версии

Private/Public Cloud

Задачу отказоустойчивости машин, сети отдаем команде/провайдеру.

На нас функционирование копонентов кластера

Причины использования:

- Безопасность - живем в приватных облаках, там все и строим
- Нам нужен k8s со своими наворотами

Bare Metal

Причины использования:

- Когда критична производительность
- Когда хотим использовать различные аппаратные ускорители
- Нужна приватная платформа

В этом варианте самостоятельно необходимо обеспечивать отказоустойчивость машин и сети

AWS

- Устанавливаем `eksctl`
- Устанавливаем переменные окружения AWS_DEFAULT_REGION, AWS_ACCESS_KEY, AWS_SECRET_ACCESS_KEY

Создаем кластер:

```
eksctl create cluster
```

Ждем 15-20 минут, получаем рабочий кластер без каких-либо аддонов

AWS Storage Class

Начиная с версии 1.11 EKS автоматически создает Storage Class для провижинга PersistentVolume.

Тома предоставляются через Amazon Elastic Block Store (EBS).

AWS Load Balancer

AWS автоматически интегрирует свой Load Balancer (ELB) в сервисы Kubernetes

Также он создает Security Group и привязывает ее к worker-ноде.

GKE

Проще устанавливать и обновлять.

GKE под капотом использует ресурсы GCP:

- VPC
- Виртуальные машины
- Диски
- Фаервол
- Балансировщик

Создание кластера в GKE

```
gcloud auth init
Create a "project":

gcloud projects create my-gke-project
gcloud config set project my-gke-project

gcloud container clusters create my-gke-cluster --region us-west1 --num-nodes=2
```

При создании первого кластера `gcloud` может вернуть ошибку и ссылку для включения GKE API.

Кластер поставляется с аддонами для мониторинга и логирования. Также можно выбрать Calico в качестве сетевого плагина.

GKE Storage Class

GKE по умолчанию предоставляет Storage Class, создающий диски в GCP

```
kubectl get storageclass
NAME                 TYPE
standard (default)   kubernetes.io/gce-pd
```

```
$ kubectl describe storageclass standard
Name: standard
IsDefaultClass: Yes
Annotations: http://storageclass.beta.kubernetes.io/is-default-
class=true
Provisioner:           kubernetes.io/gce-pd
Parameters:            type=pd-standard
AllowVolumeExpansion: <unset>
MountOptions:          <none>
ReclaimPolicy:         Delete
VolumeBindingMode:     Immediate
Events:                <none>
```

GKE LoadBalancer

При создании сервиса `LoadBalancer` GKE автоматичеки создает L4-балансировщик и правило для фаервола.

```
apiVersion: v1
 kind: Service
metadata:
  name: grafana
spec:
  ports:
    - port: 80
      targetPort: 3000
  type: LoadBalancer
  selector:
    project: devops-with-kubernetes
    app: grafana
```

```
$ kubectl get svc grafana
NAME      TYPE           CLUSTER-IP     EXTERNAL-IP     PORT(S)        AGE
grafana   LoadBalancer   10.59.244.97   35.243.118.88   80:31213/TCP   1m
```

GKE Ingress (L7 LoadBalancer)

GKE дает Ingress Controller, который сам управляет правилами фаервола, L7-балансировщиком и остальными сервисами GCP.

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
    name: nginx-tomcat-ingress
spec:
    rules:
     - http:
         paths:
         - path: /
           backend:
             serviceName: nginx
             servicePort: 80
         - path: /examples
           backend:
             serviceName: tomcat
             servicePort: 8080
         - path: /examples/*
           backend:
             serviceName: tomcat
             servicePort: 8080
```

Другие провайдеры

- Alibaba Cloud
- IBM Cloud
- OVH
- Scaleway (private beta)
- Selectel
- MCS Mail.ru
- Яндекс.Облако

Self-hosted

Какое железо использовать

- Маленькие ноды лучше, чем большие
- Унифицированное управление (ILO, IPMI, желательно CLI)
- Быстрые диски для etcd

Подготовка машин

В рамках подготовки машин, на узлах нам надо:

- Отключить swap
- Отключить фильтрацию трафика
- Включить маршрутизацию (ip_forward)
- Включить максимальную производительность ЦП (scaling_governor)

Установщики

- kubeadm
- kubespray
- kops
- Rancher

kops

- Развертывает Kuberntetes в облачных провайдерах (AWS, GCE, Digital Ocean)
- Развертывает HA master-ноды
- Конфигурация описывается в манифестах

Kops on GitHub

kubespray

- Ansible playbook для установки/обновления Kubernetes
- Поддерживает большинство популярных Linux-дистрибутивов
- Выбор множества сетевых плагинов
- HA режим

Kubespray on GitHub

kubeadm

- Для деплоя руками
- Более полный контроль над управлением кластером
- HA режим

Create cluster with kubeadm

RKE (Rancher Kubernetes Engine)

- С помощью провадеров создает все ноды для кластера
- Может устанавливать уже на существующих машинах
- Поддерживает HA режим
- Графический интерфейс
- Отсутствует Iac

Rancher

Обновление кластера

Общие рекомендации:

- Перед обновлением читать release notes новой версии и официальные руководства по обновлению
- Обновлять теми инструментами, которыми устанавливали
- Использовать IaC и не допускать confuguration drift

Какие компонеты должны быть совместимы

- kubectl
- API server
- kubelet
- controller manager
- etcd
- kube-dns or CoreDNS
- CNI plugin(s)
- Network controller, network policy controller
- Container engine
- Linux kernel

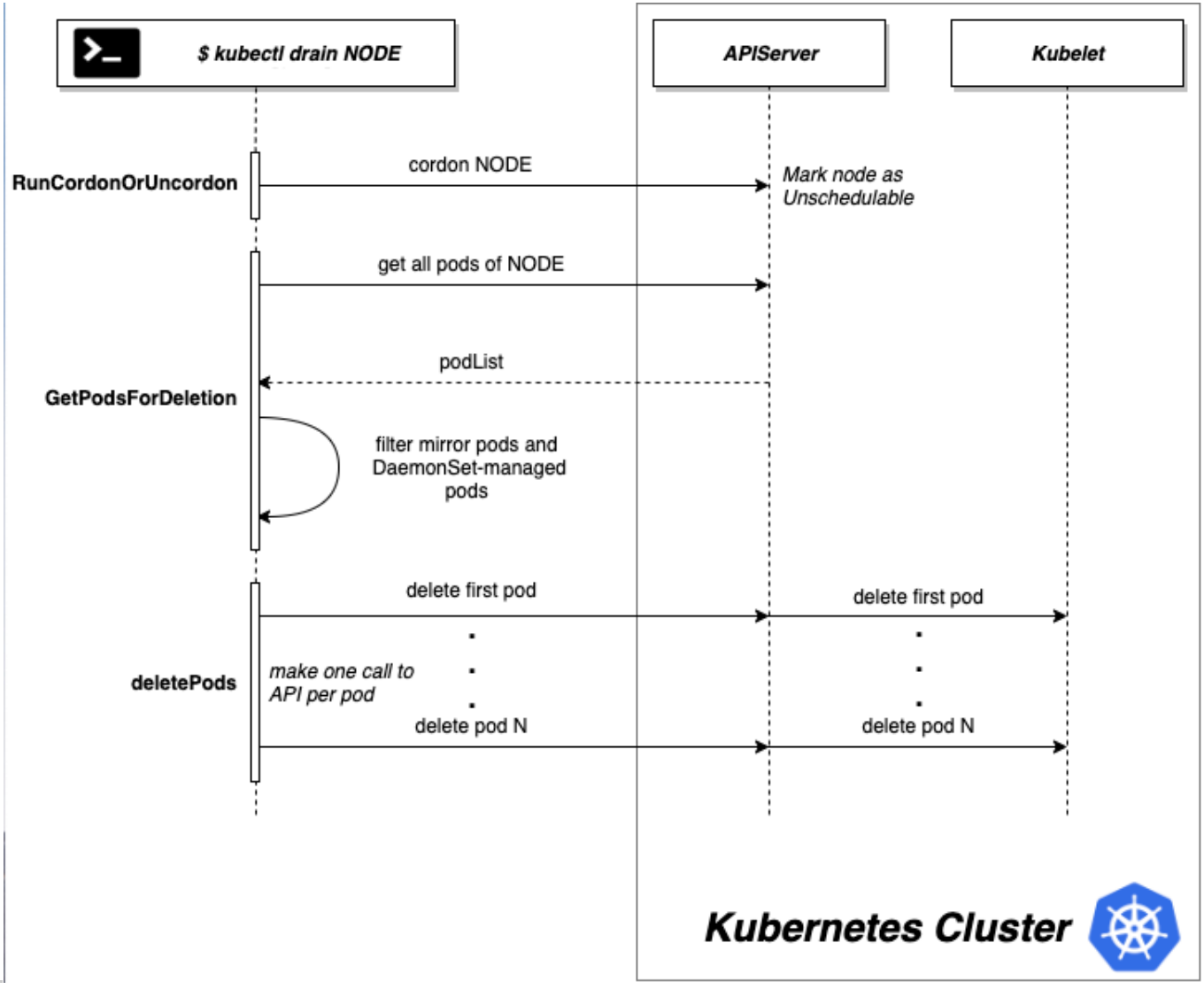
Вывод ноды из кластера

Для вывода подов с ноды и ее удаления из планирования используется команда `kubectl drain`

```
kubectl drain $NODE
```

Документация

kubectl drain



Возвращение ноды в кластер

Для возвращения ноды в кластер используется команда

```
kubectl uncordon $NODE
```

Документация

Обновление с помощью kubespray

```
ansible-playbook upgrade-cluster.yml -b -i inventory/sample/hosts.ini -e
kube_version=v1.6.0
```

Подробности

Резервное копирование кластера

Цели резервного копирования:

- Восстановление после сбоев
- Клонирование кластера

Процесс эксплуатации кластера

- Вся инфраструктура описана кодом и хранится в Git
- Изменения применяются автоматически
- Данные StatefulSet компонент хранятся вне кластера и для них настроено резервное копирование

Что необходимо копировать

- etcd
- State StatefulSet

Инструменты для резервного копирования

- Velero
- Stash - копирование Persistent Volumes
- Снапшоты через Kubernetes API