

Автоскейлинг в Kubernetes

Насколько это реально, как использовать, ограничения

Вадим Исаканов, Cogia.de

<https://github.com/vadimisakanov/>





LISTEN. ANALYZE. ACT.

Мы классные!



Vadim Isakanov 4:29 PM

Привет

Я сегодня выступаю на митапе в Селектел, будет 100-200 человек, разработчики и админы, сделать какоенибудь объявление про нас?)))

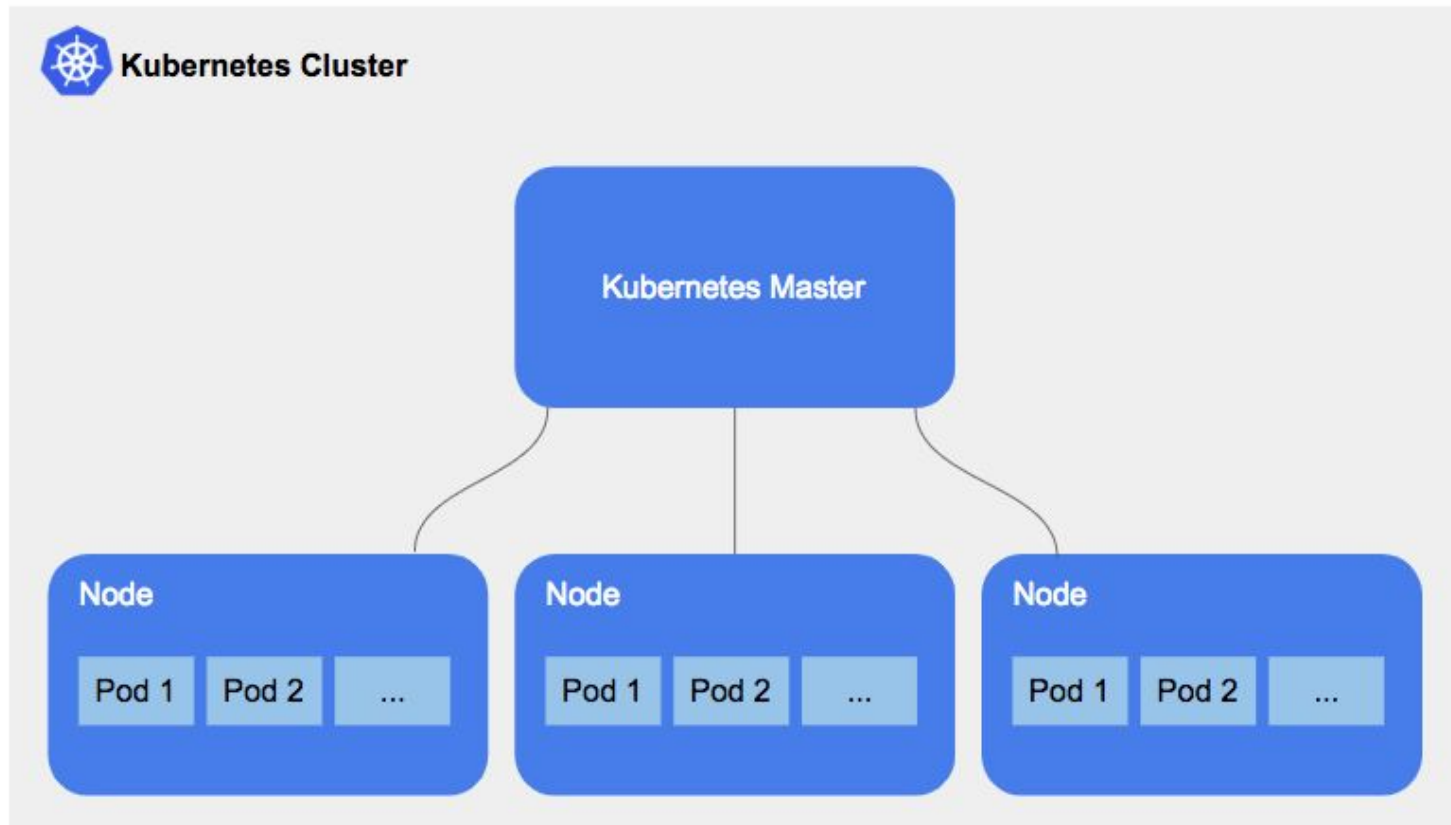
Вакансии и т.п. 😊



Yakov Bezrukov 4:29 PM

Похвастайся какие мы классные 😊

Kubernetes - платформа для инфраструктуры

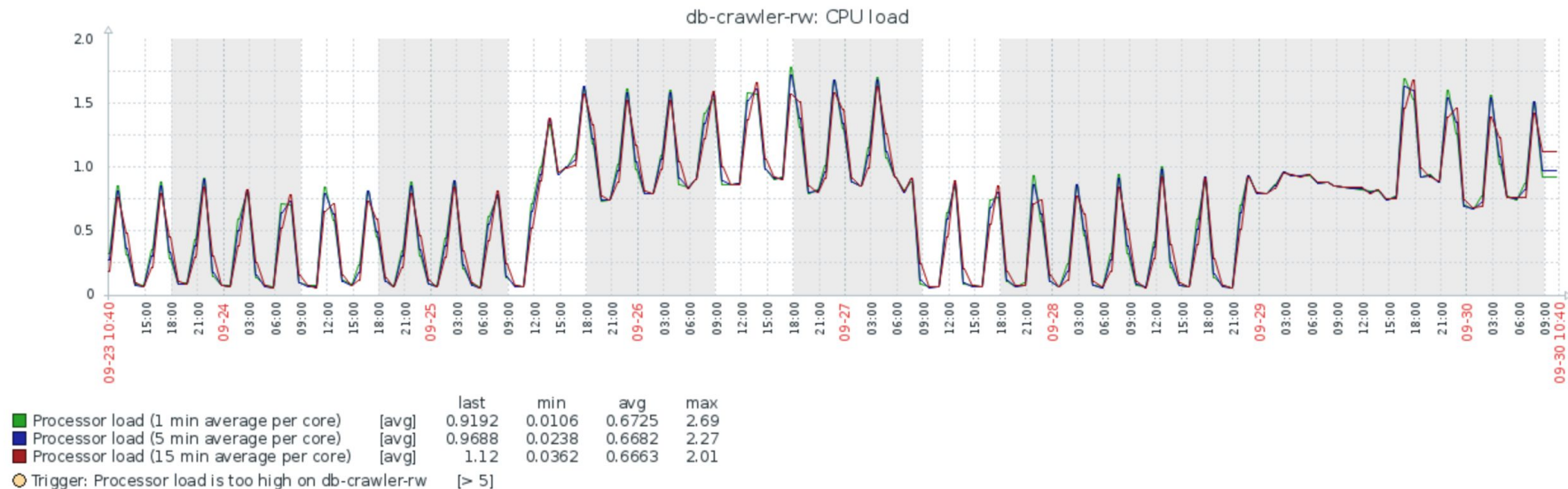


Нужен ли вам автоскейлинг?

Кубернетес заявляет автоскейлинг (автоматическое масштабирование) "из коробки".

Это одна из ключевых возможностей K8S - использовать только необходимые ресурсы, кластеры и приложения масштабируют себя сами.

Да - при неравномерной загрузке



Как это сделать?

Скейлинг подов:

- горизонтальный (horizontal pod autoscaler)
- вертикальный (vertical pod scaling)

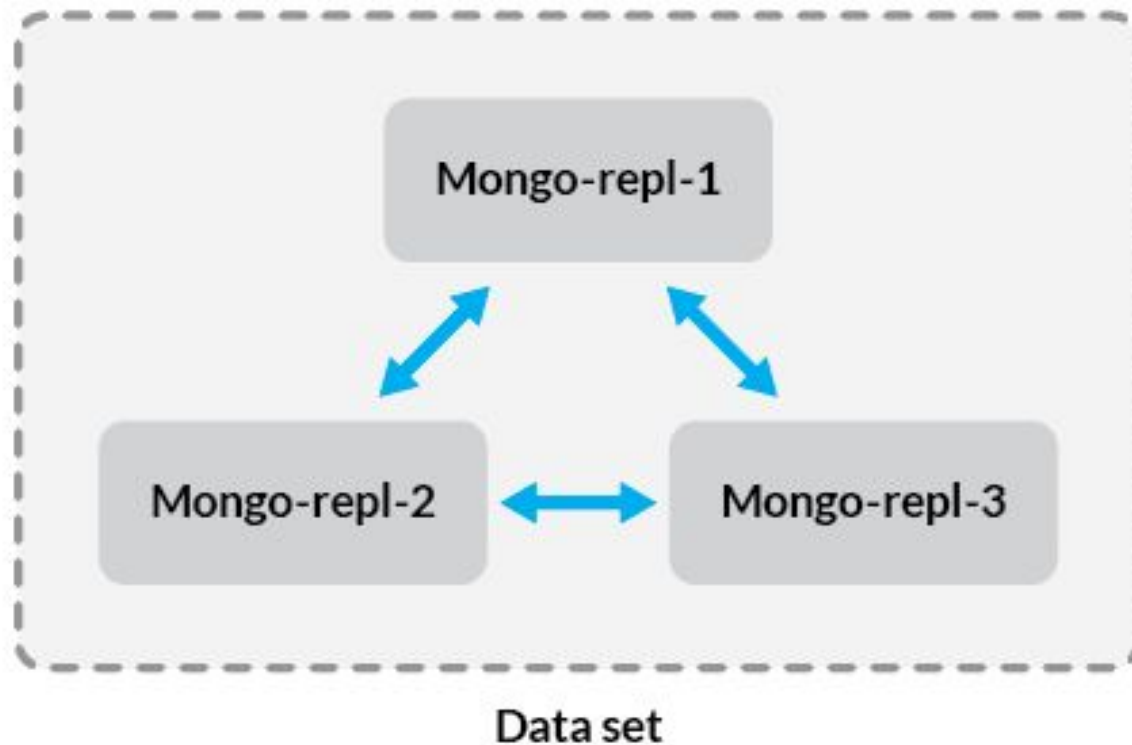
Скейлинг нод:

- Cluster Autoscaler

Horizontal pods autoscaler

- Масштабируется число подов (контейнеров)
- Скейлинг на основе запросов ресурсов
- Не требуется пересоздание подов

Горизонтальное масштабирование



Включаем horizontal pod autoscaler

```
$ helm install stable/metrics-server --name metrics-server
```

```
$ kubectl edit deployment metrics-server
```

```
- --metric-resolution=60s
```

```
- --kubelet-preferred-address-types=InternalIP
```

```
$ kubectl top pod
```

```
$ kubectl autoscale deployment wordpress --max=100  
--cpu-percent=80
```

© <https://do.co/2pAWsuE>



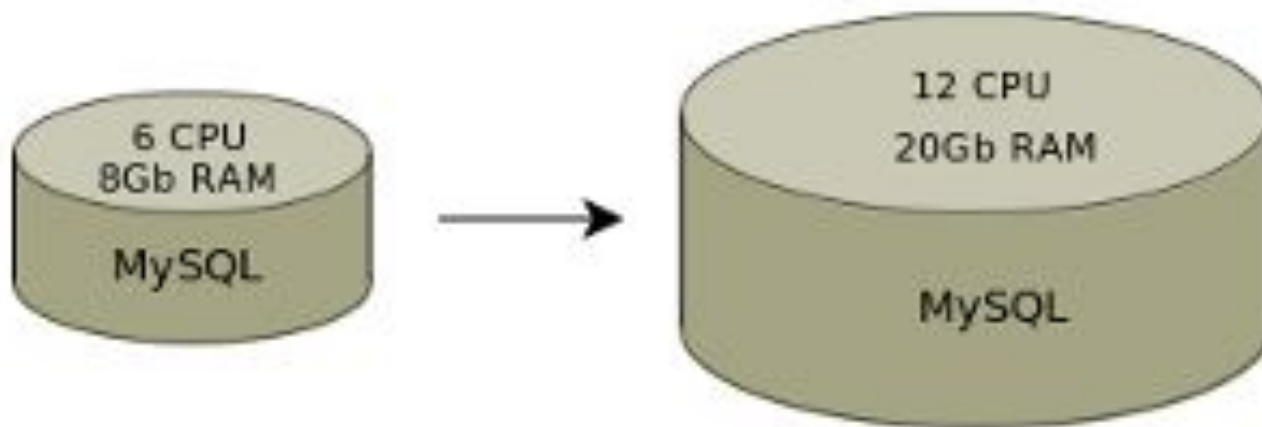
Vertical pods autoscaler

- Ресурсы (“размер”) подов \wedge
- Изменяются доступные ресурсы
- Поды необходимо пересоздать

```
$ kubectl delete pod mysql
```

```
$ kubectl create mysql
```

Вертикальное масштабирование



Включаем vertical pod autoscaler

```
$ git clone https://github.com/kubernetes/autoscaler.git
```

```
$ ./hack/vpa-up.sh # installation
```

```
$ vi mysql-deployment.yaml
```

```
apiVersion: autoscaling.k8s.io/v1beta2
```

```
kind: VerticalPodAutoscaler
```

```
metadata:
```

```
  name: mysql-vpa
```

```
spec:
```

```
  targetRef:
```

```
    apiVersion: "apps/v1"
```

```
    kind: Deployment
```

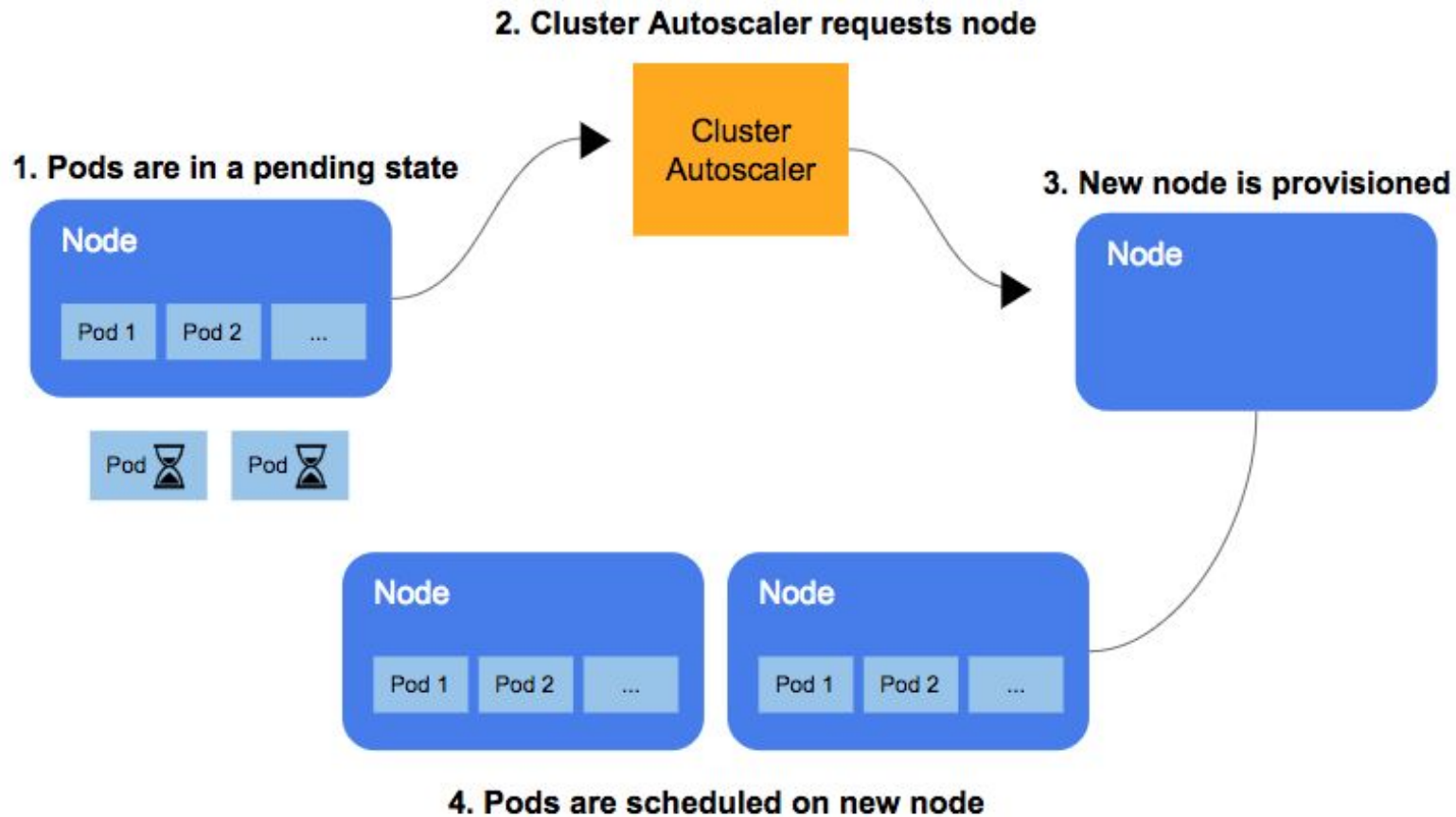
```
    name: mysql
```

© <http://bit.ly/2ol2xFf>



Nodes scaling - Cluster Autoscaler

- Официально доступно в Кубернетес
- Работает после установки Cluster Autoscaler
<https://github.com/kubernetes/autoscaler/tree/master/cluster-autoscaler/>
- Выполняет скейлинг при нехватке или избытке ресурсов



Cluster Autoscaling Providers

- Amazon AWS
- Google Cloud Platform
- Openstack Magnum (планируется для Selectel?)
- MS Azure
- А еще DigitalOcean, AliCloud, BaiduCloud, Packet.net, Kubemark

Устанавливаем Cluster Autoscaler

```
$ kubectl apply -f cluster-autoscaler-secret.yaml
```

```
$ kubectl apply -f cluster-autoscaler-svaccount.yaml
```

```
$ kubectl apply -f cluster-autoscaler-deployment.yaml
```

```
--cluster-name # имя k8s кластера  
--cloud-provider # aws, gce, magnum, etc  
--nodes # min:max:NodeGroupName
```

MAGIC

© <http://bit.ly/2nSj89j>



Масштабирование вверх

Для новых подов недостаточно свободных нод, лимит нод не превышен:

- заказываются новые ноды
- добавляются в кластер
- новые поды разворачиваются на них.

Реальная скорость скейлинга - от нескольких минут

Масштабирование вниз

Pod Resource Requests (запросы на ресурсы) **ниже лимита**, поды с одних нод можно переместить на другие ноды:

- выполняется drain нод, поды останавливаются и запускаются на других нодах
- нода выводится из кластера
- Cluster Autoscaler производит отказ от ноды

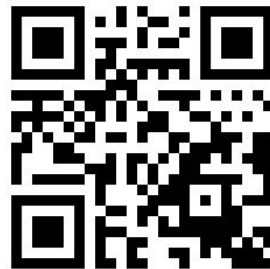
Реальная скорость скейлинга - минуты/часы, много вариантов

Проблемы и особенности

- Агрессивный скейлинг вверх, медленный вниз
- Скейлинг не мгновенный
- Нужно четко определять лимиты ресурсов
- Нельзя привязывать поды к нодам
- Нормально масштабировать можно только stateless приложения

Почитать

- Medium/Kubecost
<http://bit.ly/2nddxKm>
- k8s.io/hpa <http://bit.ly/2pAEjNA>
- k8s.io/cluster-autoscaler
<http://bit.ly/2pzvR0V>



Спасибо!

github.com/vadimisakanov

t.me/vadimisakanov

vadim.isakanov@gmail.com

