K8S 39: Kubernetes. Deployment. Введение

Описание:

Начиная с этого задания, мы рассмотрим пути деплоймента Kubernetes кластера, начиная с локального кластера.
Достаточно просто установить Kubernetes на небольшой тестовый стенд. Вы можете скачать Kubernetes из репозитория на одной или нескольких виртуальных машинах или физических серверах.
Однако запуск Kubernetes в масштабе с производственными рабочими нагрузками требует большей осмысленности и усилий.
Ниже приведены критерии, которые следует учитывать при оценке архитектуры Kubernetes для рабочих нагрузок проекта:

- Высокая доступность — развертывает ли ваше решение Kubernetes кластеры высокой доступности с репликацией базовых метаданных для восстановления после сбоев?
- Обновления — Kubernetes обеспечивает мажорные обновление каждые 3-4 месяца. Какова ваша стратегия обновления Kubernetes? Какие периоды простоя потребуются для обновления и приемлемо ли это для бизнеса?
- Поддержка гибридных систем — поддерживает ли ваше решение Kubernetes узлы, расположенные в частных датацентрах и облаке, на которых должен работать Kubernetes? Предлагает ли он одинаковый или аналогичный уровень SLA и функциональность по ним?
- Поддержка федеративности — поддерживает ли ваше решение Kubernetes развертывание федеративных кластеров, которые могут расти в частных и публичных облаках для обеспечения надежности инфраструктуры и динамической пакетной совместимости?
- Дополнительные возможности — какие дополнительные функции необходимы вашей команде для запуска Kubernetes в масштабе и поддержки большого числа пользователей? Поддерживаются ли они вашим решением Kubernetes? Некоторые примеры включают поддержку единого входа, RBAC, изолированную сеть, постоянное хранилище.

Сразу начнем с самого простого варианта...
Minikube
Вариант использования: песочница разработчика.
Плюсы: удобство, локализованная песочница, тестирование, изучение.
Минусы: не пригоден для промышленных нагрузок, не масштабируется, не может быть распределенным.

Minikube — это рекомендованный официальный метод для изучения работы Kubernetes. Он позволяет установить однонодовый кластер в виртуальной машине практически с любым поставщиком (virtualbox, kvm, hyperv, vmware), а также позволяет запустить и на хосте при помощи Docker. Данный вариант хорошо подходит для запуска кластера с целью ознакомления, тестирования новых версий кластера и инструмента для кластера.
Аналогом данного решения можно рассмотреть Minishift, который представляет собой инструмент для запуска кластера OpenShift в аналогичной манере, как и Minikube. Однако данное решение ограничено одним хостом, а значит, всю гибкость и все возможности Kubernetes вы попросту не сможете увидеть. Более того, не со всеми инструментами получится познакомиться, в чем вы чуть позднее убедитесь.
Если же вы решили, что одной ноды вам уже не хватает и хочется бОльшего, то у вас есть 2 варианта — либо развернуть свой кластер на собственных мощностях (или арендованных) и заниматься полным управлением самим, либо запустить кластер в поставщике конечного решения и не думать ни о чем, кроме мощностей, которые вам требуются (на самом деле, даже это может не требоваться в некоторых случаях).
Хотелось бы еще упомянуть про возможность запуска кластера на SoC (System on Chip) системах, начиная от Raspberry Pi, благодаря проектам типа k3s и MicroK8s от компаний Rancher (к ней мы еще вернемся) и Canonical (создатели дистрибутива Ubuntu) соответственно. Да, звучит это дико, но, как минимум, забавно и в домашних инсталляциях (даже на виртуальных машинах) может быть полезно.
Более сложный вариант...
Kubernetes с использованием контейнеров
Вариант использования: песочница разработчика.
Плюсы: удобство, локализованная песочница, тестирование, изучение.
Минусы: не пригоден для промышленных нагрузок, не масштабируется, не может быть распределенным.
Docker очень легко позволяет быстро опробовать фреймворки в изолированных средах, поскольку установка пакета и его зависимостей в контейнере не приводит к их установке на хосте.
Kubernetes можно легко развернуть как набор контейнеров Docker на одном хосте. Хост может быть физическим сервером, виртуальной машиной или рабочей машиной разработчика.
Компоненты Kubernetes запускаются в виде нескольких контейнеров в этом подходе развертывания:

- etcd: — этот компонент хранит данные конфигурации, которые могут быть доступны Kubernetes Master’s API Server путем простого HTTP или JSON API.
- API Server: — этот компонент является центром управления для главного узла Kubernetes. Это облегчает связь между различными компонентами, тем самым поддерживая работоспособность кластера.
- Controller Manager: — этот компонент обеспечивает соответствие желаемого состояния кластеров текущему состоянию путем масштабирования рабочих нагрузок.

- Scheduler: — этот компонент распределяет рабочую нагрузку на соответствующем узле (лейбл, ресурсы, и т.д.) — в данном случае все рабочие нагрузки будут размещены локально на вашем хосте.

Чтобы начать работу с Kubernetes, используя Docker-контейнеры на локальном хосте, запустите компоненты Kubernetes, как описано ниже:
Версия Kubernetes может быть изменена по желанию. Давайте использовать 1.15.5 для этого упражнения. Установите переменную среды K8S_VERSION в 1.15.5, используя следующую команду:

```
# export K8S_VERSION=v1.5.6
```

Запустите kubelet, используя имадж hypercube из gcr.io. Kubelet загрузит главные компоненты Kubernetes (api-сервер, планировщик, Controller Manager) и другие службы, такие как etcd, в виде отдельных контейнеров Docker. Обратите внимание, что мы запускаем kubelet также как контейнер docker (- containerized).

```
# docker run \
–volume=/:/rootfs:ro \
–volume=/sys:/sys:ro \
–volume=/dev:/dev \
–volume=/var/lib/docker/:/var/lib/docker:rw \
–volume=/var/lib/kubelet/:/var/lib/kubelet:rw \
–volume=/var/run:/var/run:rw \
–net=host \
–pid=host \
–privileged=true \
-d \
gcr.io/google_containers/hyperkube:${K8S_VERSION} \
/hyperkube kubelet –containerized \
–hostname-override=”127.0.0.1″ –address=”0.0.0.0″ \
–api-servers=http://localhost:8080 \
–config=/etc/kubernetes/manifests
```

Запустите service proxy, который используется для балансировки нагрузки входящих запросов на конечные точки:

```
# docker run -d –net=host –privileged \
gcr.io/google_containers/hyperkube:${K8S_VERSION} \
/hyperkube proxy –master=http://127.0.0.1:8080 –v=2
```

Теперь вам доступен запуск kubectl прямо на данном хосте:

```
# ./kubectl get nodes
NAME STATUS AGE
127.0.0.1 Ready 2m
```

Если вы запускаете docker на OS/X, то затем вам нужно будет перенаправить порт 8080. Далее мы с вами будем рассматривать и вариант с собственной инсталляцией кластера, и вариант с облачным поставщиком, но для начала вспомним про варианты локальной установки.

Полезные ссылки:

- Installing Kubernetes with Minikube (official docs)
- Getting started (official docs)
- K3s - Lightweight Kubernetes
- MicroK8s - Zero-ops Kubernetes for workstations and edge / IoT

Задание:

1. Какое из перечисленных решений рекомендуется в официальной документации для изучения Kubernetes?
- Minishift
- Minikube
- MicroK8s
- K3s
1. В каких случаях могут использовать развернутые кластеры на железе?
- Компании требуется защищенный, полностью контролируемый контур.
- Ваша компания - банк.
- Разработчики решили, что обработка персональных данных в Kubernetes - новый тренд.
- Все перечисленное.
1. В чем удобство использования облачных кластеров у облачных поставщиков?
- Не нужно думать о деньгах.
- Решает вопросы закупки и обслуживания оборудования для кластера.
- Не нужно думать о ресурсах.
- Оно работает на магии.
1. Какой из облачных поставщиков НЕ предлагает возможность облачного кластера?
- Google Compute Platform
- Amazon Web Services
- DigitalOcean
- Hetzner Online
- Yandex Cloud
- Mail.ru Cloud Solutions
1. Можно ли считать OpenShift поставляемым кластером Kubernetes?

- Да, там внутренности Kubernetes, так что все верно - можно пользоваться как обычным Kubernetes кластером.
- Нет, OpenShift - это отдельный продукт со своими особенностями.