

Использование Kubernetes для оркестрации

Kubernetes — это мощная платформа для оркестрации контейнеров, которая автоматизирует развертывание, управление и масштабирование контейнеризованных приложений. В этой документации мы рассмотрим основные концепции Kubernetes, процесс установки Kubernetes на НАЙС ОС, создание и управление Kubernetes-кластерами, основные команды `kubectl`, а также мониторинг и логирование в Kubernetes. В примерах будет использоваться пакетный менеджер [tdnf](#), который является стандартным для НАЙС ОС.

Основные концепции Kubernetes: поды, сервисы, деплойменты

Прежде чем углубиться в детали установки и использования Kubernetes, важно понять основные концепции этой платформы.

Поды (Pods)

Под — это наименьшая и самая простая единица в Kubernetes, которая представляет собой один или несколько контейнеров, работающих в общей изолированной среде. Контейнеры в поде разделяют сеть и хранилище, что позволяет им легко взаимодействовать друг с другом. Поды обычно используются для запуска одного экземпляра приложения или службы.

Сервисы (Services)

Сервис предоставляет стабильный IP-адрес и DNS-имя для набора подов, что позволяет легко обращаться к подам, независимо от их расположения в кластере. Сервисы обеспечивают балансировку нагрузки и автоматическое обнаружение подов.

Деплойменты (Deployments)

Деплоймент управляет созданием и обновлением подов и обеспечивает их желаемое состояние. С помощью деплойментов можно описать, сколько экземпляров подов должно быть запущено, как обновлять поды и как восстанавливать их в случае сбоя.

Установка Kubernetes: Minikube, kubectl, облачные решения

Установка Kubernetes может выполняться различными способами в зависимости от требований и окружения. Рассмотрим установку Kubernetes с использованием Minikube и kubectl на НАЙС ОС.

Установка Minikube

Minikube — это инструмент для запуска локального Kubernetes-кластера на одном узле. Он идеально подходит для разработки и тестирования.

Для установки Minikube на НАЙС ОС выполните следующие шаги:

Установка Minikube и зависимостей

Установите необходимые зависимости:

```
sudo dnf install -y curl conntrack
```

Загрузите и установите Minikube:

```
curl -Lo minikube  
https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64  
chmod +x minikube  
sudo mv minikube /usr/local/bin/
```

Запуск Minikube

Запустите Minikube:

```
minikube start --driver=none
```

Эта команда запустит локальный Kubernetes-кластер.

Установка kubeadm

kubeadm — это инструмент для быстрой настройки Kubernetes-кластера. Он автоматизирует большинство ручных шагов, необходимых для создания кластера.

Для установки Kubernetes с помощью kubeadm на НАЙС ОС выполните следующие шаги:

Установка kubeadm, kubelet и kubectl

Добавьте репозиторий Kubernetes и установите необходимые пакеты:

```
sudo dnf install -y kubectl kubeadm kubelet  
sudo systemctl enable --now kubelet
```

Инициализация кластера

Инициализируйте кластер с помощью kubeadm:

```
sudo kubeadm init --pod-network-cidr=10.244.0.0/16
```

После инициализации выполните команды для настройки kubectl:

```
mkdir -p $HOME/.kube  
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config  
sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

Установка сетевого плагина

Установите сетевой плагин, например, Flannel:

```
kubectl apply -f
https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel
.yml
```

Облачные решения

Для производственных сред и масштабируемых приложений часто используются облачные решения Kubernetes, такие как Google Kubernetes Engine (GKE), Amazon Elastic Kubernetes Service (EKS) и Azure Kubernetes Service (AKS). Эти сервисы предоставляют управляемые Kubernetes-кластеры, упрощая управление и масштабирование.

Создание и управление Kubernetes-кластерами

После установки Kubernetes важно уметь создавать и управлять кластерами. Рассмотрим основные команды и процедуры для этого.

Создание подов

Для создания пода используйте команду `kubectl run` или создайте манифест YAML.

Создание пода с помощью команды `kubectl run`:

```
kubectl run mypod --image=nginx --restart=Never
```

Пример манифеста YAML для пода

Создайте файл `pod.yaml` с содержимым:

```
apiVersion: v1
kind: Pod
metadata:
  name: mypod
spec:
  containers:
  - name: nginx
    image: nginx
```

Создайте под с помощью манифеста:

```
kubectl apply -f pod.yaml
```

Создание сервисов

Сервисы используются для обеспечения постоянного IP-адреса и DNS-имени для набора подов.

Пример манифеста YAML для сервиса

Создайте файл `service.yaml` с содержимым:

```
apiVersion: v1
kind: Service
metadata:
  name: myservice
spec:
  selector:
    app: myapp
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
  type: ClusterIP
```

Создайте сервис с помощью манифеста:

```
kubectl apply -f service.yaml
```

Создание деплоиментов

Деплоименты управляют созданием и обновлением подов и обеспечивают их желаемое состояние.

Пример манифеста YAML для деплоимента

Создайте файл `deployment.yaml` с содержимым:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: mydeployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: myapp
  template:
    metadata:
      labels:
        app: myapp
    spec:
      containers:
        - name: nginx
```

```
image: nginx
ports:
- containerPort: 80
```

Создайте деплоймент с помощью манифеста:

```
kubectl apply -f deployment.yaml
```

Основные команды kubectl

kubectl — это командная строка для управления Kubernetes-кластерами. Рассмотрим основные команды kubectl, используемые в повседневной работе.

Работа с подами

- `kubectl get pods`: отображает список подов.
- `kubectl describe pod`: отображает подробную информацию о поде.
- `kubectl delete pod`: удаляет под.
- `kubectl logs`: отображает логи контейнера в поде.
- `kubectl exec`: выполняет команду внутри контейнера в поде.

Работа с сервисами

- `kubectl get services`: отображает список сервисов.
- `kubectl describe service`: отображает подробную информацию о сервисе.
- `kubectl delete service`: удаляет сервис.

Работа с деплоями

- `kubectl get deployments`: отображает список деплоев.
- `kubectl describe deployment`: отображает подробную информацию о деплое.
- `kubectl delete deployment`: удаляет деплой.
- `kubectl rollout status deployment`: отображает статус развертывания деплоя.
- `kubectl scale deployment`: изменяет количество реплик в деплое.
- `kubectl rollout undo deployment`: откатывает деплой к предыдущему состоянию.

Мониторинг и логирование в Kubernetes

Эффективный мониторинг и логирование являются ключевыми аспектами для поддержания стабильности и производительности приложений в Kubernetes-кластере.

Мониторинг

Для мониторинга Kubernetes-кластера часто используются такие инструменты, как Prometheus и Grafana.

Установка Prometheus и Grafana

Установите Prometheus и Grafana с помощью манифестов Kubernetes:

```
kubectl apply -f
https://raw.githubusercontent.com/prometheus-operator/prometheus-
operator/master/bundle.yaml
kubectl apply -f
https://raw.githubusercontent.com/grafana-operator/grafana-operator/master/deploy/
manifests/00-crds.yaml
kubectl apply -f
https://raw.githubusercontent.com/grafana-operator/grafana-operator/master/deploy/
manifests/01-namespace.yaml
kubectl apply -f
https://raw.githubusercontent.com/grafana-operator/grafana-operator/master/deploy/
manifests/03-grafana-deployment.yaml
```

Настройка мониторинга

Создайте сервис для доступа к Grafana:

```
apiVersion: v1
kind: Service
metadata:
  name: grafana
spec:
  type: NodePort
  ports:
    - port: 3000
      targetPort: 3000
  selector:
    app: grafana
```

Примените манифест:

```
kubectl apply -f grafana-service.yaml
```

Теперь вы можете получить доступ к Grafana через NodePort.

Логирование

Для логирования в Kubernetes часто используются такие инструменты, как Fluentd, Elasticsearch и Kibana (EFK Stack).

Установка EFK Stack

Установите Fluentd, Elasticsearch и Kibana с помощью манифестов Kubernetes:

```
kubectl apply -f
https://raw.githubusercontent.com/kubernetes/kubernetes/master/cluster/addons/flue
ntd-elasticsearch/fluentd-es-configmap.yaml
kubectl apply -f
https://raw.githubusercontent.com/kubernetes/kubernetes/master/cluster/addons/flue
ntd-elasticsearch/fluentd-es-ds.yaml
```

```
kubectl apply -f
https://raw.githubusercontent.com/kubernetes/kubernetes/master/cluster/addons/flue
ntd-elasticsearch/es-statefulset.yaml
kubectl apply -f
https://raw.githubusercontent.com/kubernetes/kubernetes/master/cluster/addons/flue
ntd-elasticsearch/kibana-deployment.yaml
```

Настройка логирования

Создайте сервис для доступа к Kibana:

```
apiVersion: v1
kind: Service
metadata:
  name: kibana
spec:
  type: NodePort
  ports:
    - port: 5601
      targetPort: 5601
  selector:
    app: kibana
```

Примените манифест:

```
kubectl apply -f kibana-service.yaml
```

Теперь вы можете получить доступ к Kibana через NodePort для просмотра и анализа логов.

Kubernetes предоставляет мощные инструменты для оркестрации контейнеров, позволяя автоматизировать развертывание, управление и масштабирование приложений. Понимание основных концепций Kubernetes, установка и настройка кластера на НАЙС ОС, а также использование основных команд `kubectl` и инструментов для мониторинга и логирования помогут вам эффективно использовать эту платформу в вашей ИТ-инфраструктуре.