Федеральное государственное бюджетное образовательное учреждение высшего образования
«Сибирский государственный университете телекоммуникаций и информатики»

Лабораторная работа №5

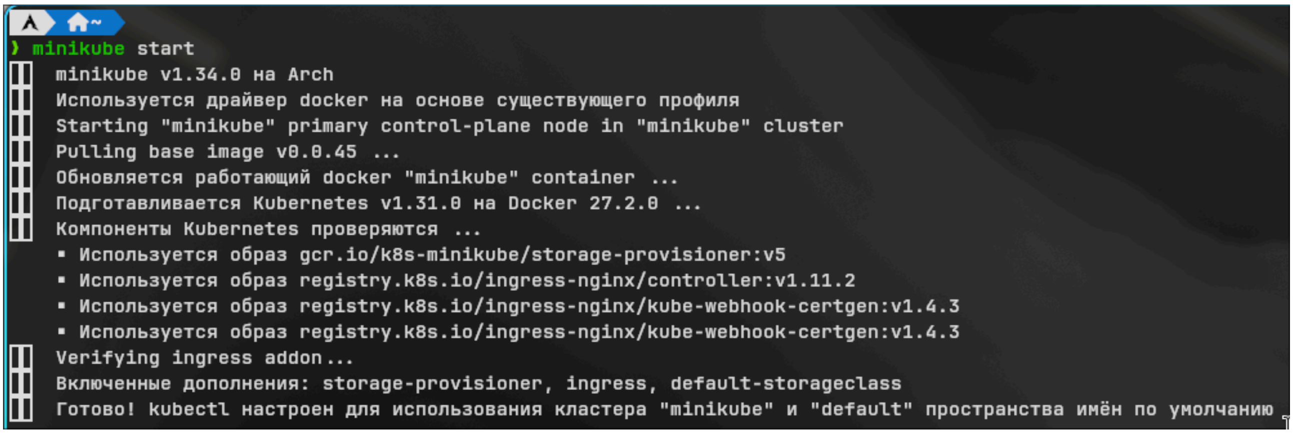
Архитектура вычислительных систем

Выполнил студент группы :
ИП-216
Андрущенко Филипп

Новосибирск
2024

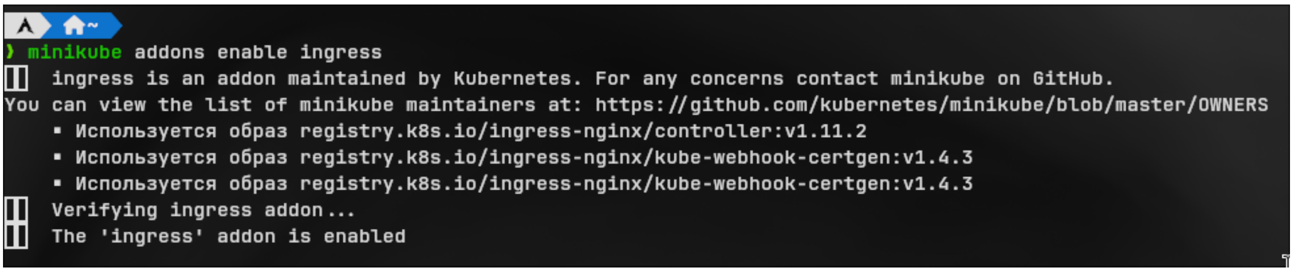
DOD на 3 :

Запуск Minikube



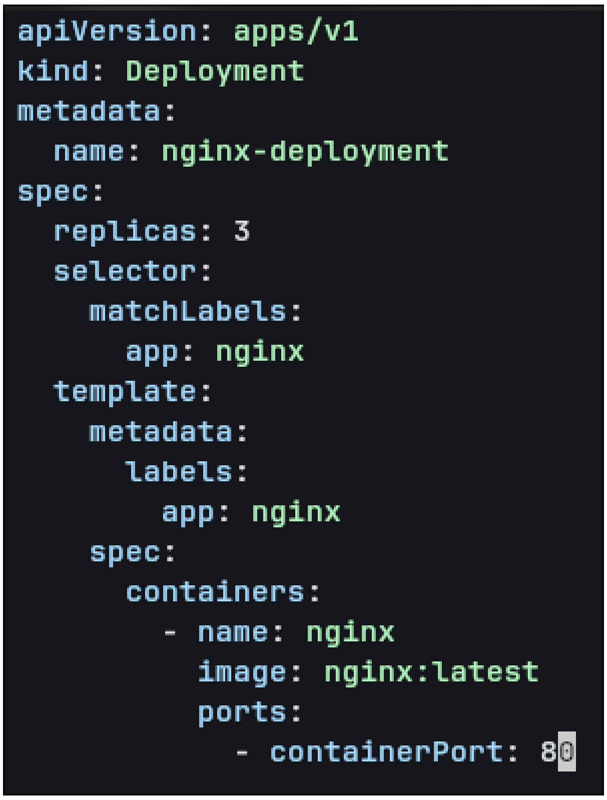
```
~
❯ minikube start
   minikube v1.34.0 на Arch
   Используется драйвер docker на основе существующего профиля
   Starting "minikube" primary control-plane node in "minikube" cluster
   Pulling base image v0.0.45 ...
   Обновляется работающий docker "minikube" container ...
   Подготавливается Kubernetes v1.31.0 на Docker 27.2.0 ...
   Компоненты Kubernetes проверяются ...
    ▪ Используется образ gcr.io/k8s-minikube/storage-provisioner:v5
    ▪ Используется образ registry.k8s.io/ingress-nginx/controller:v1.11.2
    ▪ Используется образ registry.k8s.io/ingress-nginx/kube-webhook-certgen:v1.4.3
    ▪ Используется образ registry.k8s.io/ingress-nginx/kube-webhook-certgen:v1.4.3
   Verifying ingress addon...
   Включенные дополнения: storage-provisioner, ingress, default-storageclass
   Готово! kubectl настроен для использования кластера "minikube" и "default" пространства имён по умолчанию
```

Включаем ingress, который позволяет управлять внешним доступом к сервисам внутри кластера



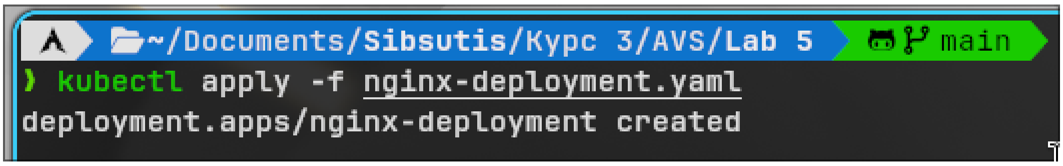
```
~
❯ minikube addons enable ingress
   ingress is an addon maintained by Kubernetes. For any concerns contact minikube on GitHub.
You can view the list of minikube maintainers at: https://github.com/kubernetes/minikube/blob/master/OWNERS
    ▪ Используется образ registry.k8s.io/ingress-nginx/controller:v1.11.2
    ▪ Используется образ registry.k8s.io/ingress-nginx/kube-webhook-certgen:v1.4.3
    ▪ Используется образ registry.k8s.io/ingress-nginx/kube-webhook-certgen:v1.4.3
   Verifying ingress addon...
   The 'ingress' addon is enabled
```

Создаём Deployment, который разворачивает nginx



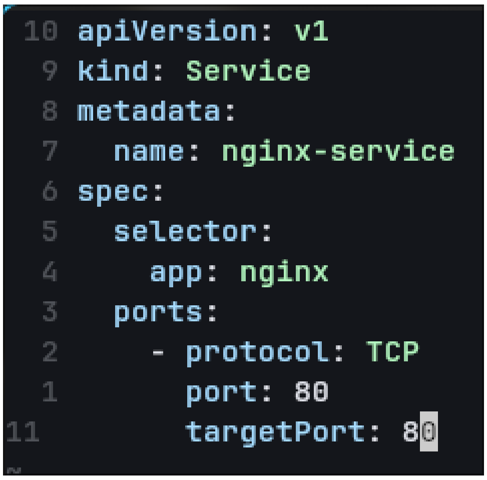
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:latest
          ports:
            - containerPort: 80
```

Применяем манифест



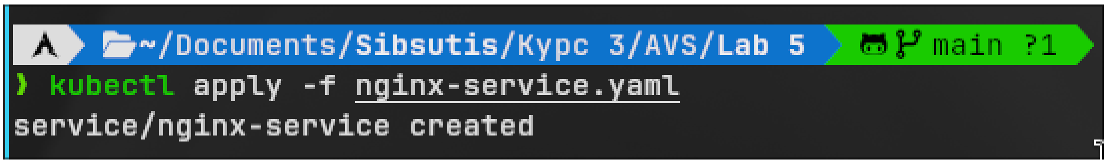
```
~/Documents/Sibsutis/Курс 3/AVS/Lab 5   main
❯ kubectl apply -f nginx-deployment.yaml
deployment.apps/nginx-deployment created
```

Создаём Service, чтобы был сетевой доступ к контейнерам



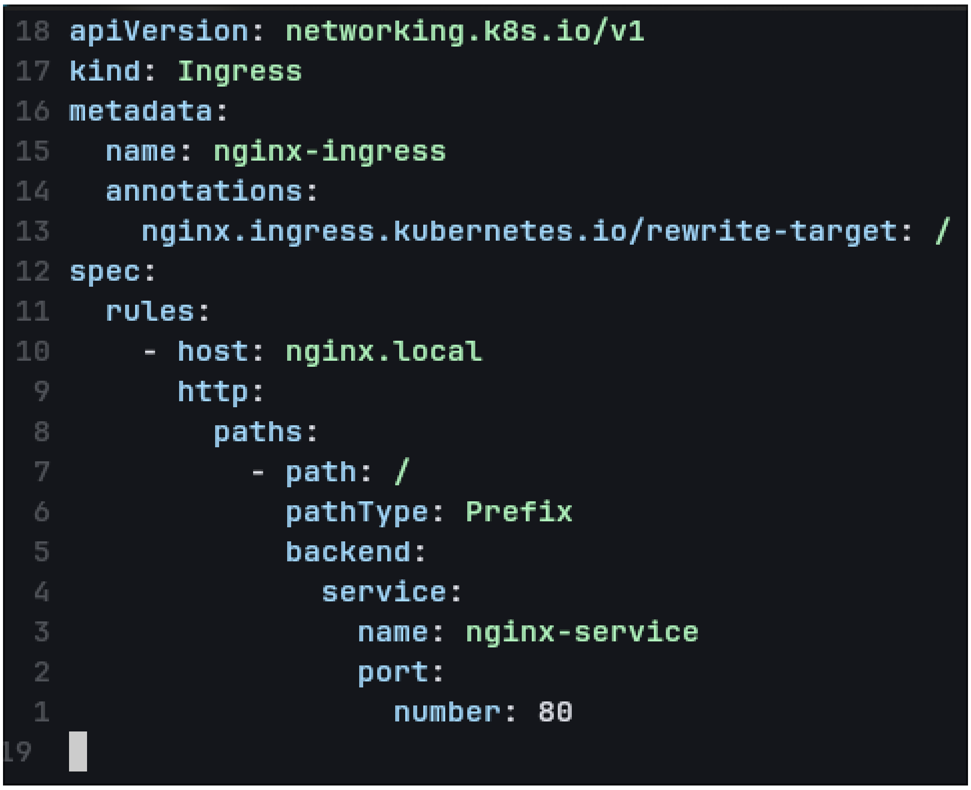
```yaml
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
spec:
  selector:
    app: nginx
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
```

Применяем манифест



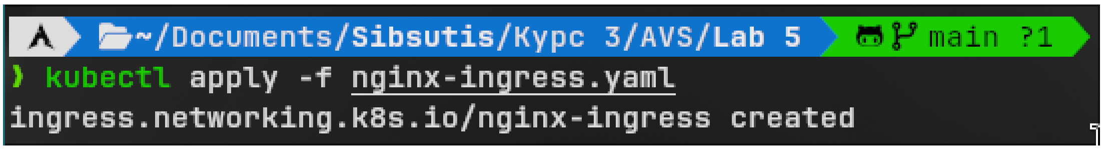
```
~/Documents/Sibsutis/Курс 3/AVS/Lab 5   main ?1
❯ kubectl apply -f nginx-service.yaml
service/nginx-service created
```

Настраиваем Ingress для маршрутизации трафика



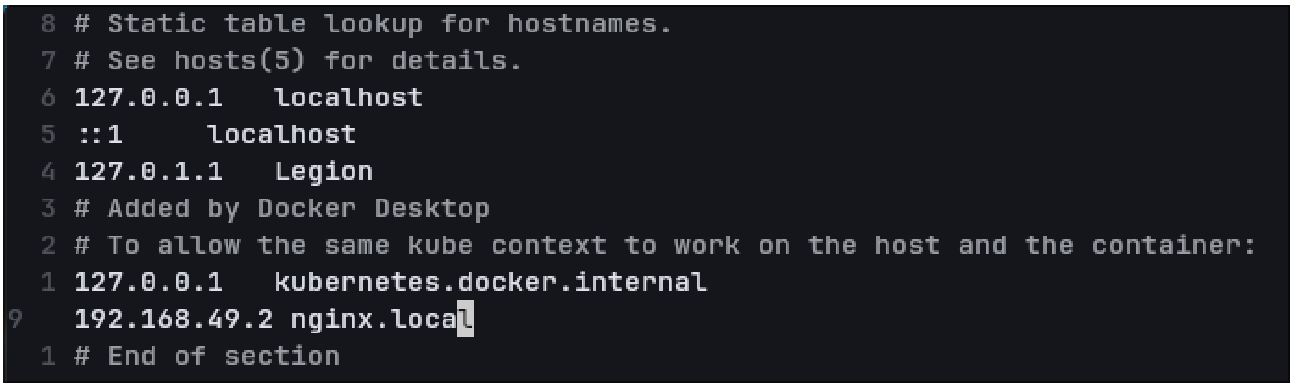
```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: nginx-ingress
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /
spec:
  rules:
    - host: nginx.local
      http:
        paths:
          - path: /
            pathType: Prefix
            backend:
              service:
                name: nginx-service
                port:
                  number: 80
```

Применяем манифест



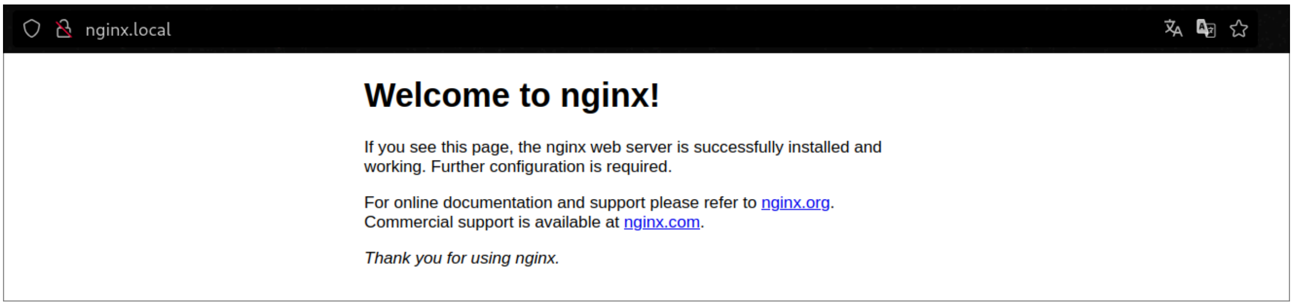
```
~/Documents/Sibsutis/Курс 3/AVS/Lab 5   main ?1
❯ kubectl apply -f nginx-ingress.yaml
ingress.networking.k8s.io/nginx-ingress created
```

Добавляем домен в /etc/hosts



```
# Static table lookup for hostnames.
# See hosts(5) for details.
127.0.0.1   localhost
::1         localhost
127.0.1.1   Legion
# Added by Docker Desktop
# To allow the same kube context to work on the host and the container:
127.0.0.1   kubernetes.docker.internal
192.168.49.2 nginx.local
# End of section
```

Проверка



nginx.local

Welcome to nginx!

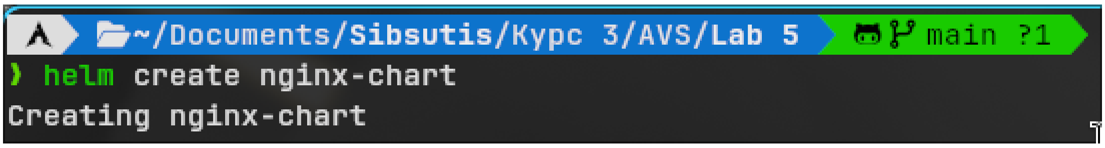
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

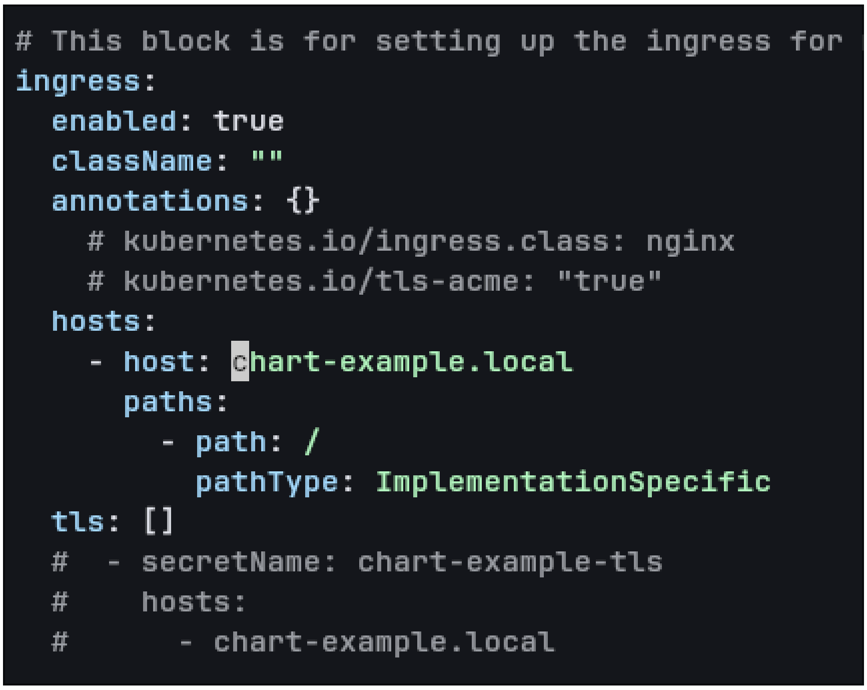
DOD на 4 :

Создаём helm-чарт

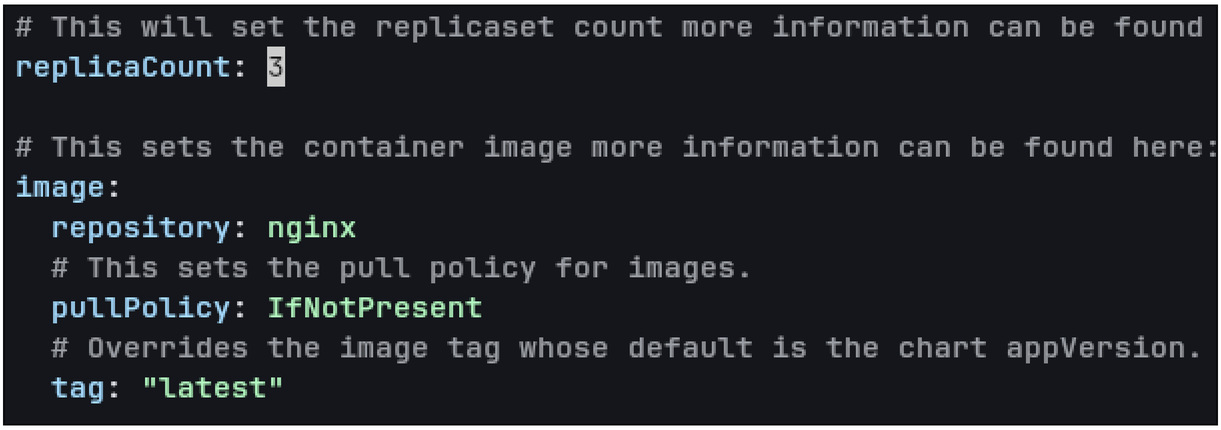


```
~/Documents/Sibsutis/Курс 3/AVS/Lab 5   main ?1
❯ helm create nginx-chart
Creating nginx-chart
```

Задаём настройки

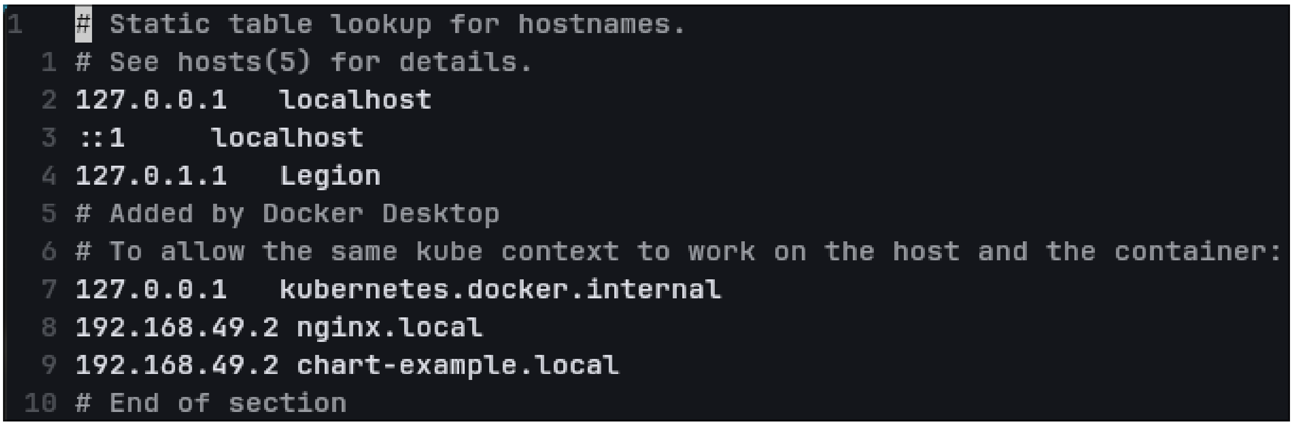


```
# This block is for setting up the ingress for
ingress:
  enabled: true
  className: ""
  annotations: {}
    # kubernetes.io/ingress.class: nginx
    # kubernetes.io/tls-acme: "true"
  hosts:
    - host: chart-example.local
      paths:
        - path: /
          pathType: ImplementationSpecific
  tls: []
  #  - secretName: chart-example-tls
  #    hosts:
  #      - chart-example.local
```



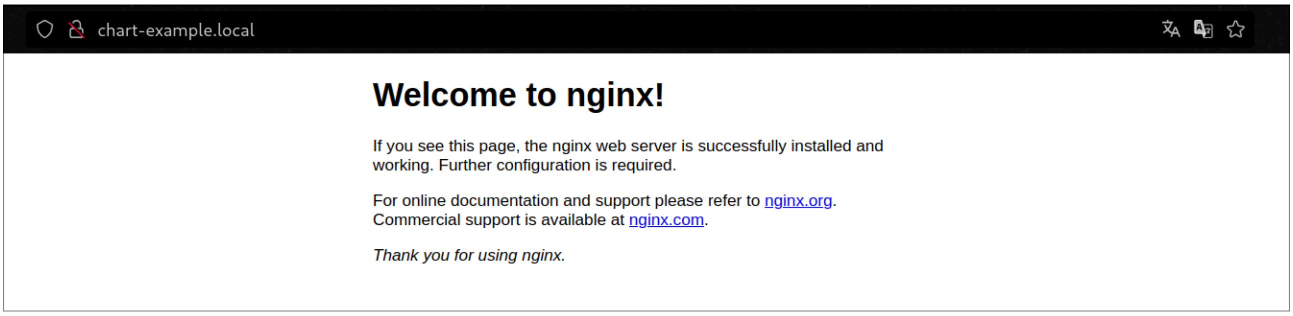
```
# This will set the replicaset count more information can be found
replicaCount: 3

# This sets the container image more information can be found here:
image:
  repository: nginx
  # This sets the pull policy for images.
  pullPolicy: IfNotPresent
  # Overrides the image tag whose default is the chart appVersion.
  tag: "latest"
```

Добавляем домен в /etc/hosts



```
# Static table lookup for hostnames.
# See hosts(5) for details.
127.0.0.1    localhost
::1       localhost
127.0.1.1    Legion
# Added by Docker Desktop
# To allow the same kube context to work on the host and the container:
127.0.0.1    kubernetes.docker.internal
192.168.49.2 nginx.local
192.168.49.2 chart-example.local
# End of section
```

Проверка



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.