

# Ingress и Ingress Controller

# Слайд 1. Цель урока

## Что изучим:

- Различия между Ingress и NodePort.
- Роль Ingress Controller.
- Популярные контроллеры: Nginx, Traefik.
- Практическая настройка маршрутизации HTTP(S) по домену.

## Почему важно:

- Позволяет централизованно управлять внешним доступом к сервисам Kubernetes.
- Упрощает работу с SSL и виртуальными хостами.
- Избегает необходимости открывать каждый NodePort отдельно.

## Слайд 2. NodePort vs Ingress

Фича	NodePort	Ingress
Доступ к сервису	Через <code>&lt;NodeIP&gt;:&lt;Port&gt;</code>	Через HTTP/HTTPS домены и пути
Количество открытых портов	Каждый сервис открывает свой порт	Один порт для всех маршрутов (обычно 80/443)
SSL/TLS	Настраивается вручную для каждого сервиса	Настраивается централизованно на контроллере
Маршрутизация	Нет поддержки виртуальных хостов	Поддержка маршрутизации по хосту и пути
Масштабируемость	Сложно для большого числа сервисов	Удобно для десятков и сотен сервисов

**Вывод:** Ingress удобен для управления внешним HTTP(S) доступом, особенно в production.

## Слайд 3. Что такое Ingress

- **Ingress** — объект Kubernetes, который описывает правила маршрутизации HTTP(S) трафика к сервисам внутри кластера.
- Позволяет:
  - Пропускать запросы на разные сервисы по домену или пути.
  - Настраивать SSL/TLS для одного или нескольких доменов.
  - Включать правила переписывания URL и редиректы.

## Пример структуры:

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: example-ingress
spec:
  rules:
  - host: example.com
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: web-service
            port:
              number: 80
```

## Слайд 4. Ingress Controller

- **Ingress Controller** — это компонент, который слушает объекты Ingress и создаёт маршрутизацию в кластере.
- Без контроллера Ingress ничего не работает.
- Популярные варианты:
  - **Nginx Ingress Controller**
    - Широко используется, стабильный, простая настройка.
  - **Traefik**
    - Поддержка динамических конфигураций, встроенный LetsEncrypt.
  - Есть и другие: HAProxy, Istio Gateway, Contour.

### Принцип работы:

1. Создаётся объект Ingress с правилами.
2. Контроллер читает Ingress.
3. Контроллер конфигурирует прокси (Nginx/Traefik) для маршрутизации.

# Слайд 5. Установка Nginx Ingress Controller

1. Добавляем репозиторий Helm:

```
helm repo add ingress-nginx https://kubernetes.github.io/ingress-nginx  
helm repo update
```

2. Устанавливаем контроллер:

```
helm install nginx-ingress ingress-nginx/ingress-nginx \\\n  --namespace ingress-nginx --create-namespace
```

3. Проверяем Pods:

```
kubectl get pods -n ingress-nginx
```

Что устанавливается:

- Pod с Nginx.
- Service типа LoadBalancer/NodePort.
- CRDs (если нужны для расширенных функций)

## Слайд 6. Настройка маршрутизации по домену

Ingress для конкретного домена:

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: my-app-ingress
  namespace: default
spec:
  rules:
  - host: myapp.local
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: my-app-service
            port:
              number: 80
```



## Слайд 7. SSL/TLS с Ingress

- Можно подключить сертификат вручную:

```
tls:  
- hosts:  
  - myapp.example.com  
secretName: myapp-tls
```

- Контроллер Nginx или Traefik умеет автоматически управлять сертификатами через LetsEncrypt.

## Слайд 8. Практические советы

- Всегда проверяйте порядок правил: Ingress выбирает первый совпадающий путь.
- Для тестирования используйте простой сервис `nginx` или `httpbin`.
- Следите за логами контроллера:

```
kubectl logs -n ingress-nginx deploy/nginx-ingress-controller
```

- Ingress упрощает работу с внешним доступом, но не заменяет полноценный LB при высоких нагрузках.