

Создаем Ingress

1. Перейдите в директорию: `1.9.ingress`

```
cd ~/slurm/practice/1.kube-basics-lecture/1.9.ingress/
```

2. Замените `<номер своего логина>` на Ваш номер логина в файле `ingress.yaml`

3. Создайте `deployment`, `service`, `ingress` и `configmap` из файлов `deployment.yaml`, `service.yaml`, `ingress.yaml` и `configmap.yaml`

4. Проверьте что появились два запущенных `pod'a`, `service` и `ingress`.

5. С помощью команды `curl my-app.s<номер своего логина>.edu.slurm.io` проверьте доступность приложения снаружи кластера.

7. Скопируйте ответ команды и отправьте в качестве решения для данного шага.

8. Удалите созданные объекты после выполнения задания.

Самостоятельная работа

1. Создайте и запустите `deployment` с именем `my-app-1` из образа `nginx 1.12` и `deployment` из образа `nginx:1.13` с именем `my-app-2`. В оба `deployment'a` смонтируйте конфигурацию из `configmap'ов` по аналогии с предыдущими примерами. В одной конфигурации `nginx` должен отдавать `"I am 1.12"`, в другой `"I am 1.13"`

2. Создайте `servic'ы` для обоих `deployment'ов` с именами `my-app-1` и `my-app-2` соответственно.

3. Создайте `ingress` для обоих сервисов с именем хоста `my-app.s<номер своего логина>.edu.slurm.io` таким образом, чтобы запросы на `/1` приходили на `my-app-1`, а запросы на `/2` приходили на `my-app-2`.

Для указания пути в ingress используется поле `path` на одном уровне с `backend`. Например

```
paths:  
- path: /testpath  
  backend:  
    serviceName: my-service  
    servicePort: 80
```

4. Проверьте с помощью `curl` что правила в ingress'e работают.
5. Отправьте получившийся у вас манифест ingress'a в формате YAML в качестве ответа на этом шаге.
6. После выполнения задания удалите все созданные объекты.