Lab 3. Настройка базового образ S2I и развертывание мультиконтейнерного приложения.

Упражнение 1. Создание контейнеризованного front end приложения и развертывание его в OpenShift.

Предварительные требования. Создайте репозиторий quay.io/yourname.

Для этого зайдите на quay.io под своей учетной записью RedHat.

Шаг 1. Создайте проект

oc project my-code

Приложение, которое вы развернете в этом кластере OpenShift, представляет собой службу перенаправления URL-адресов. Он имеет front end React для управления ссылками и подключается к API Node.js. Затем этот сервер подключается к базе данных Mongo для хранения и извлечения URL-адресов.

Другое приложение - микросервис «redirector», выполняет поиск в базе данных и перенаправляет трафик на соответствующий URL-адрес. Этот сервис написан на PHP.

Шаг 2. Клонируйте репозиторий с примером исходного кода

Перейдите в директорию со своими учебными проектами. Например

cd sample

и выполните команду

git clone https://github.com/nodeshift-blog-examples/urlshortener

Шаг 3. Создайте front end приложение.

Чтобы развернуть это приложение, нельзя повторно использовать тот же контейнер, который использовали при разработке. Для производственной среды необходимо упаковать приложение React и развернуть его на сервере Nginx. Образ должен будет храниться в реестре.

Скопируйте файлы Dockerfile, nginx.conf и start-nginx.sh из указаного репозитория Git:

cd front

curl https://raw.githubusercontent.com/nodeshift-blog-examples/frontend-containers/main/react-project/Dockerfile.rootless -o Dockerfile

curl https://raw.githubusercontent.com/nodeshift-blog-examples/frontend-containers/main/react-project/start-nginx.sh -o start-nginx.sh

curl https://raw.githubusercontent.com/nodeshift-blog-examples/frontend-containers/main/react-project/nginx.conf -o nginx.conf

Прим. Чтобы избежать копирования всей папки node_modules в контейнер, можно создать файл .dockerignore:

echo "node_modules" > .dockerignore

Шаг 4. Создание образа, для запуска из него pod в OpenShift кластере.

docker build -t quay.io/yourname/urlshortener-front .

Шаг 5. Отправьте образ в реестр

Теперь, когда создан образ, готовый к развертыванию, можно использовать команду docker push для отправки этого образа в реестр. Вы можете использовать любой реестр для этого шага.

Проверьте наличие созданного образа:

docker images

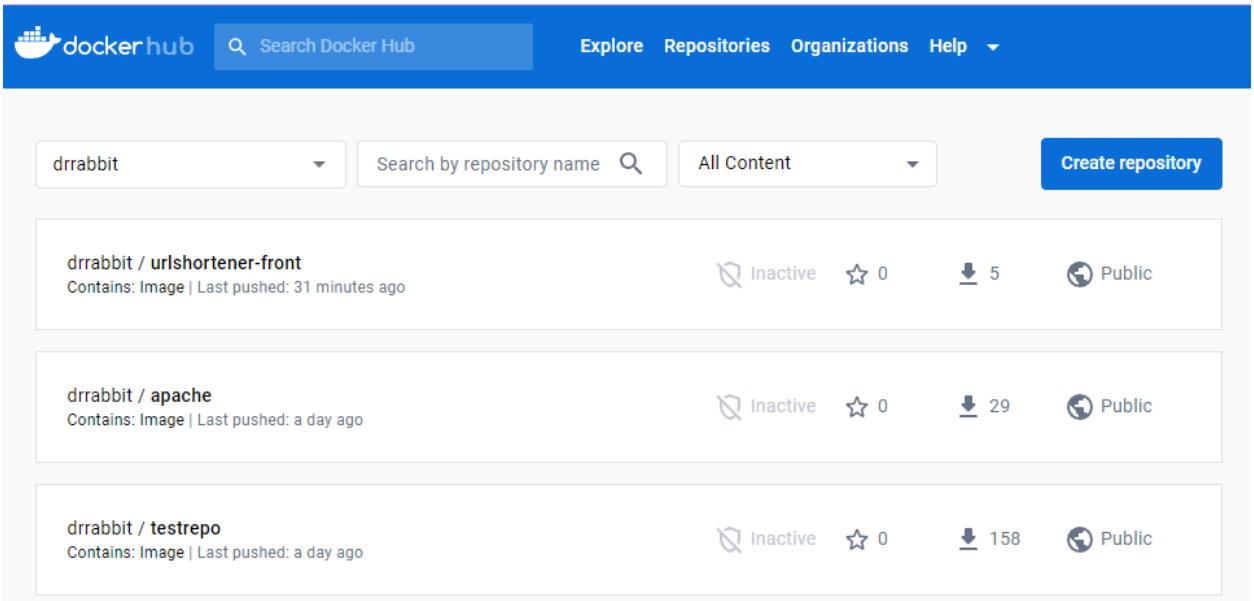
Назначьте тег образу для передачи в целевой репозиторий

docker image tag quay.io/mjsokolov/urlshortener-front:latest yourname/urlshortener-front:latest

Отправьте образ в реестр:

docker push yourname/urlshortener-front:latest

Зайдите на https://hub.docker.com/ и проверьте, что образ появился в реестре:



Упражнение 2. Развертывание front end приложения.

Шаг 1. Разверните образ в кластере OpenShift.

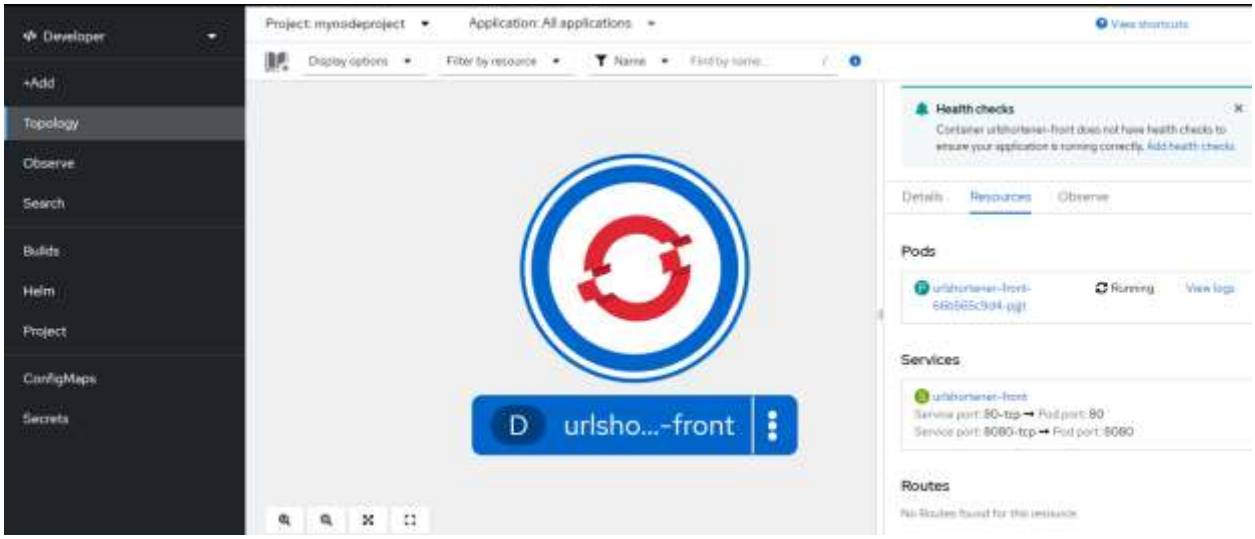
Используя команду oc, инициируйте развертывание:

oc new-app yourname/urlshortener-front:latest

Проверьте стстус:

oc status

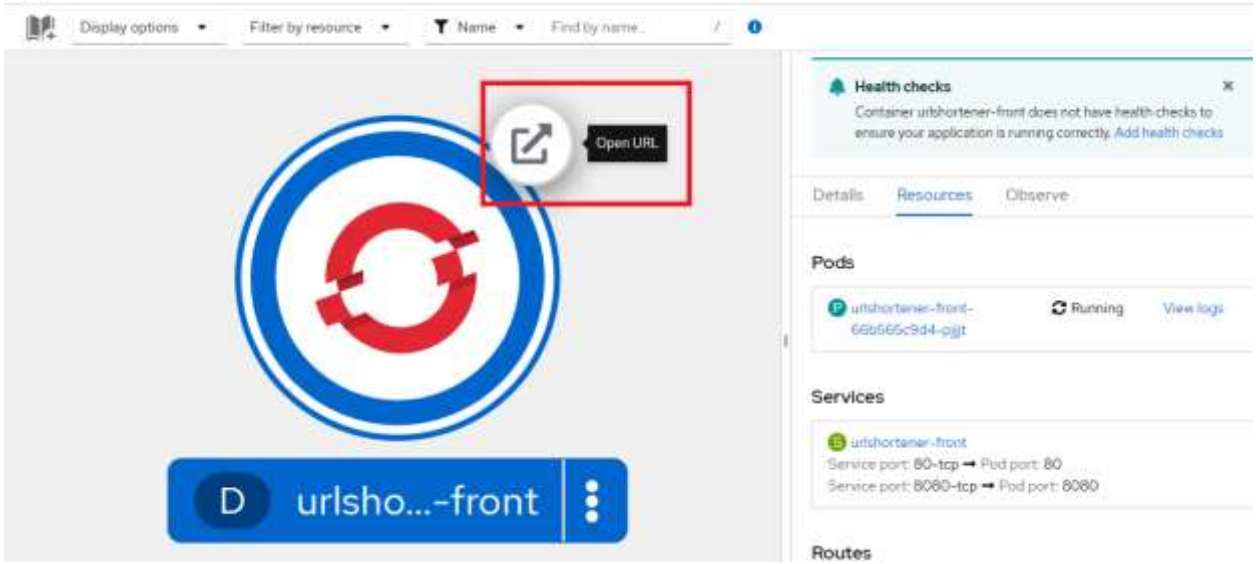
Шаг 2. В веб-консоли перейдите в перспективу разработчика в раздел Topology



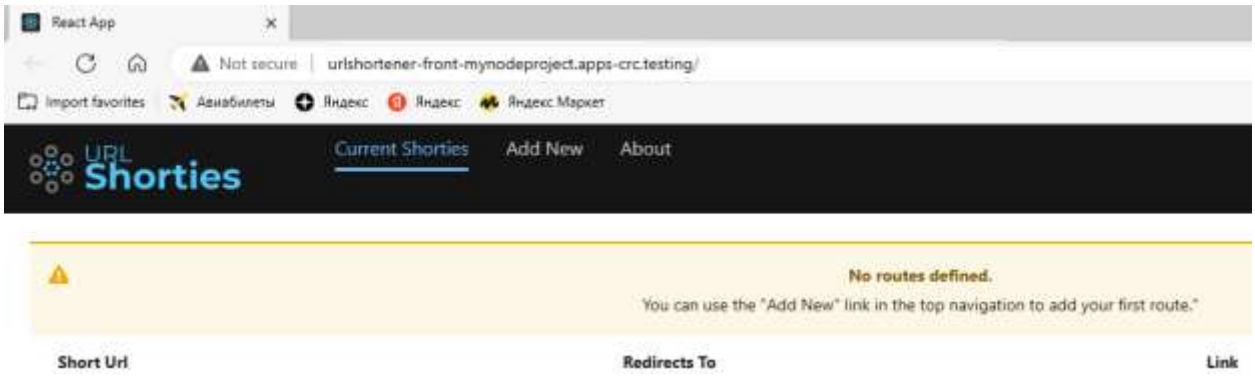
Предоставьте доступ к приложению из-за пределов кластера. Для этого создайте маршрут командой:

oc expose svc/urlshortener-front --port=8080

Теперь нажмите на Open URL:



Проверьте доступность приложения:



Lab 2. Развертывание back end части и подключение ее к front end

Упражнение 1. Развертывание back end

Шаг 1. Сборка Node.js back end с использованием S2I

Выйдите из директории front в родительскую

cd ..

Source-to-Image — это набор инструментов для создания контейнеров непосредственно из исходного кода.

Чтобы собрать серверную часть Node.js из исходного кода на GitHub, используйте команду oc new-app:

oc new-app https://github.com/nodeshift-blog-examples/urlshortener --context-dir=back

Прим. Параметр --context-dir указывает, что исходный код находится в папке /back

Перейдете к представлению «**Topology**» в консоли OpenShift и проверьте, что приложение отображается.

Шаг 2. Теперь, когда вы развернули back end часть приложения, вы можете открыть ее с помощью команды:

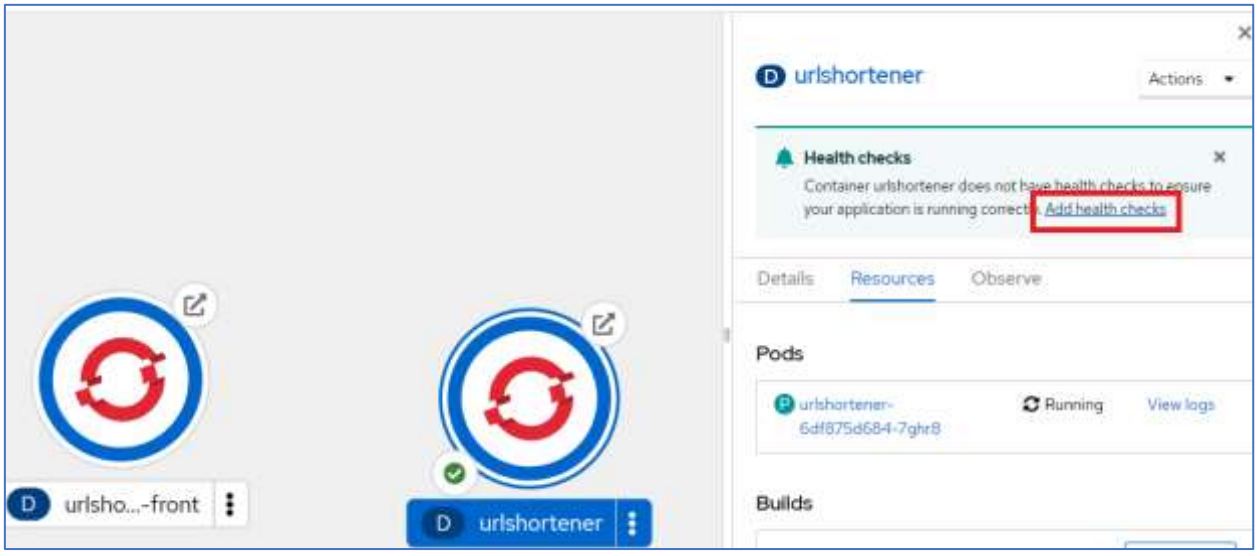
oc expose svc/urlshortener

Если нажмете ссылку «Open URL» , вы должны увидеть ответ от сервера:

{ "msg": "Hello" }

Шаг 2. Добавление health check

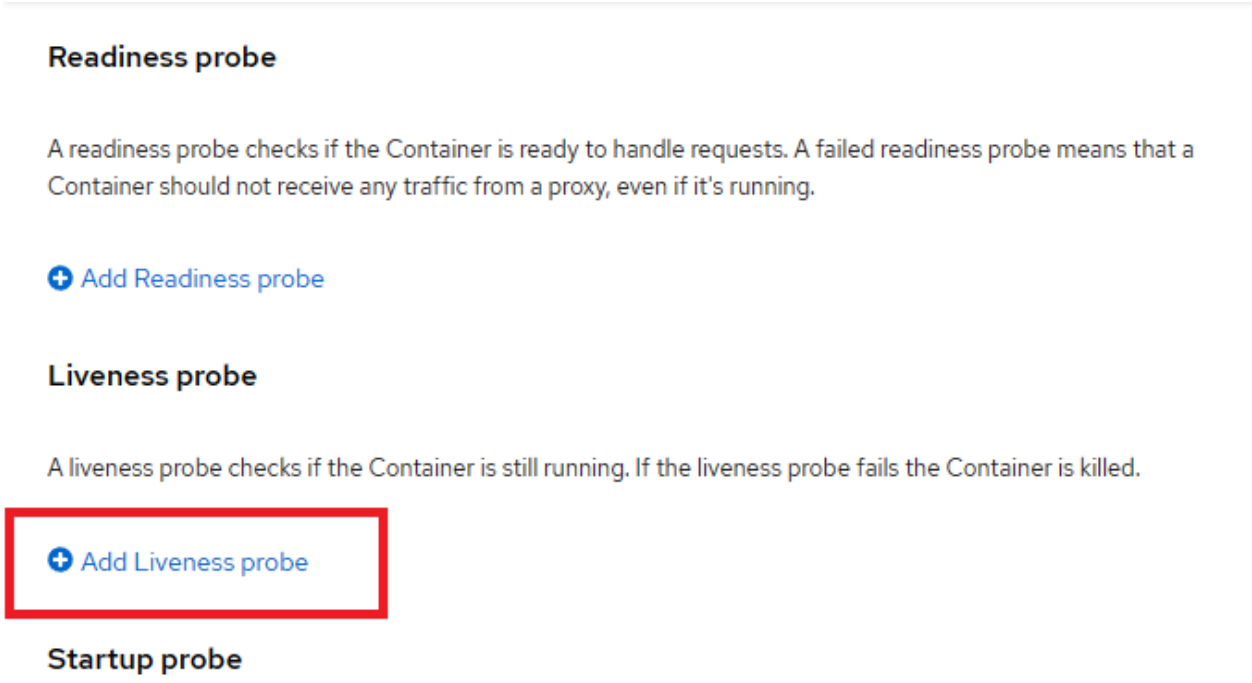
OpenShift может периодически проверять ваш модуль, чтобы узнать, работает ли он. Этот процесс называется проверкой работоспособности . На боковой панели, когда вы щелкнули развертывание urlshortener , вы могли заметить сообщение с рекомендацией добавить проверки работоспособности. Идем дальше и нажимаем **Add Health Checks**



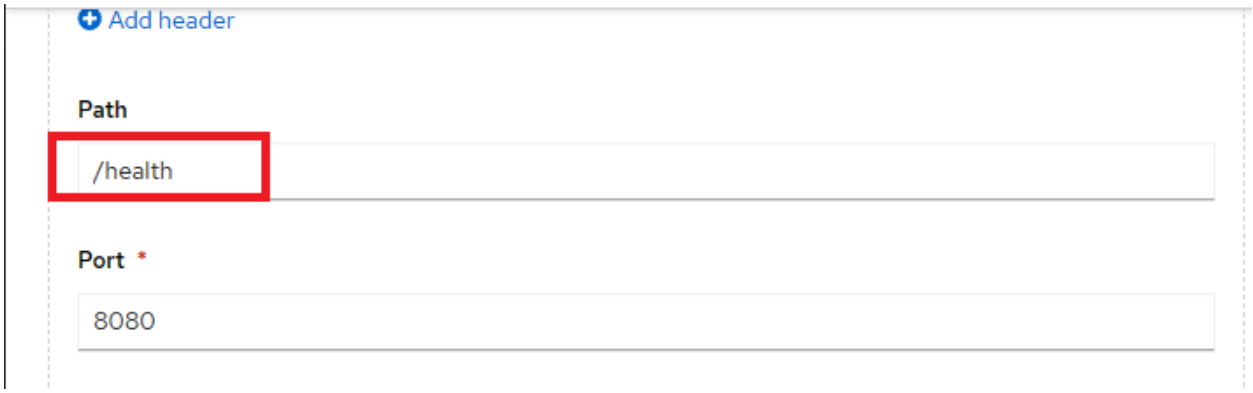
На следующем экране вы можете добавить liveness probe — проверку работоспособности, которая отслеживает ваше приложение, выполняя периодические вызовы по указанному

маршруту. Пока маршрут возвращает HTTP-код 200, OpenShift предполагает, что приложение все еще работает.

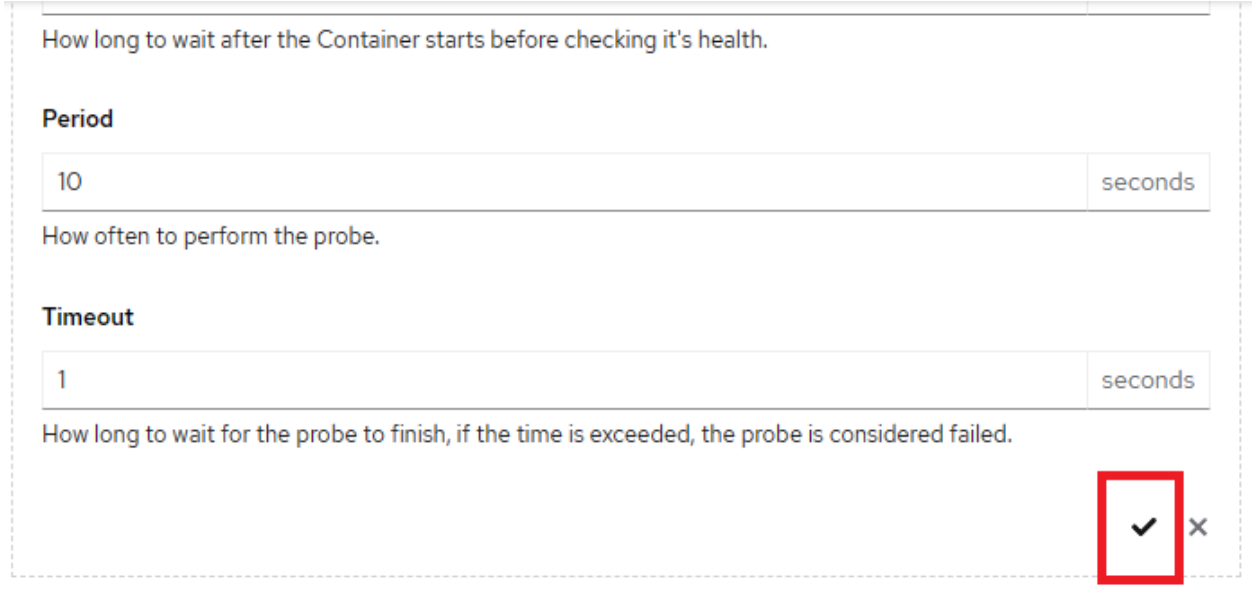
Чтобы добавить эту проверку работоспособности, нажмите «Добавить пробу Liveness» .



Измените путь /health и сохраните все остальные значения по умолчанию.

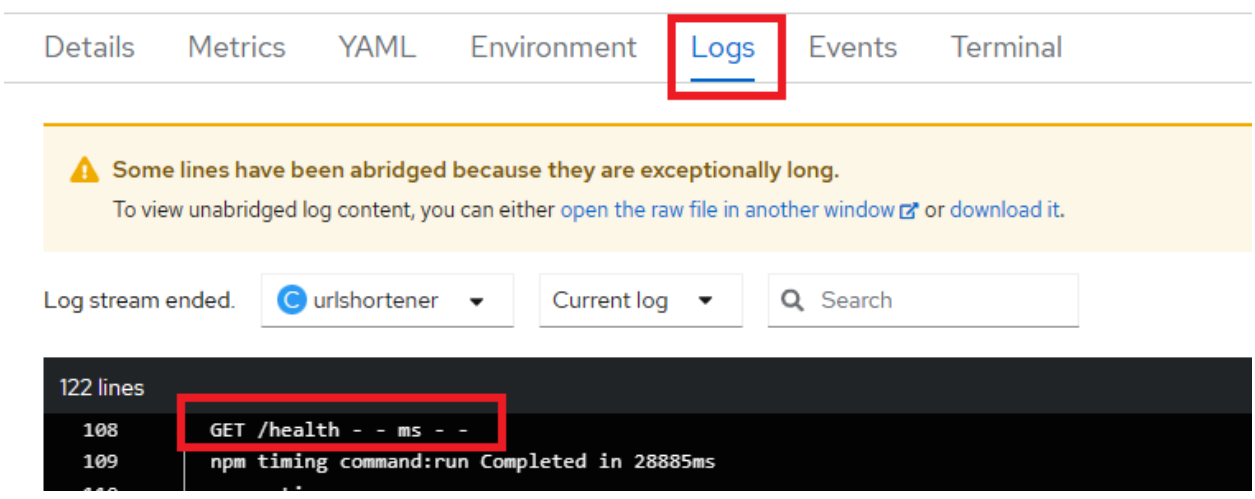


Щелкните галочку в нижней части пунктирной области,



а затем нажмите синюю кнопку «**Add**» , чтобы сохранить проверку работоспособности.

Если вы хотите убедиться, что probe работает, перейдите на вкладку «Ресурсы» на боковой панели развертывания и нажмите « Просмотреть журналы» рядом с именем модуля. На этом экране показаны журналы модуля, и вы должны видеть запрос /health каждые 10 секунд.



Упражнение 2. Подключение front end к back end

Шаг 1. Добавьте переменную среды BASE_URL.

Добавить переменные среды можно с помощью команды oc но прежде чем это сделаеь, нужно узнать к чему должен подключаться front end. Это URL-адрес, который является маршрутом к back end части. Вы можете найти этот маршрут, выполнив команду:

oc get route urlshortener

Теперь, когда вы знаете, как получить URL-адрес, вы можете установить его в качестве переменной среды для развертывания urlshortener-front с помощью команды oc set env.

Bash

oc set env deployment/urlshortener-front BASE_URL=http://$(oc get route urlshortener | awk 'NR>1 {print $2}')

deployment.apps/urlshortener-front updated

PowerShell

oc set env deployment/urlshortener-front BASE_URL=http://$((oc get route urlshortener -o json | ConvertFrom-Json).spec.host)

Теперь, когда вы настроили BASE_URL переменную среды, вы можете вернуться к приложению для сокращения URL-адресов и снова проверить его страницу «О программе»