

2. Непосредственно деплой

Шаг 1: Авторизация в Docker Hub

Перед загрузкой образов в Docker Hub (можно загружать и в собственный Docker Registry, но на это не остаётся времени 😞) авторизуемся.

```
docker login
```

Шаг 2: Сборка и тегирование Docker-образов

2.1. Сборка Maven-проекта

Сначала соберём наши Java-модулитс помощью Maven:

```
cd /path/to/multi-module-app mvn clean package
```

* Или же в IntelliJ Idea можно в инструменте Maven прожать 2 вкладки - clean+install.

2.2. Сборка и тегирование образов

2.2.1. Сборка образа для service-a

```
cd service-a  
docker build -t vsevolodshengelay/service-a:latest .
```

2.2.2. Сборка образа для service-b

```
cd ../service-b  
docker build -t vsevolodshengelay/service-b:latest .
```

2.2.3. Сборка образа для api-gateway

```
cd ../api-gateway  
docker build -t vsevolodshengelay/api-gateway:latest .
```

Шаг 3: Загрузка образов в Docker Hub

Теперь, когда образы собраны и тегированы, загрузим их в Docker Hub.

3.1. Загрузка образа service-a

```
docker push vsevolodshengelay/service-a:latest
```

3.2. Загрузка образа service-b

```
docker push vsevolodshengelay/service-b:latest
```

3.3. Загрузка образа api-gateway

```
docker push vsevolodshengelay/api-gateway:latest
```

Шаг 4. Применение манифестов Kubernetes

Теперь мы можем применить манифесты к нашему кластеру.

4.1. Применение манифестов

```
kubectl apply -f k8s/
```

4.2. Проверка состояния подов

Проверим, что хотя бы поды запустились (находятся в состоянии Running) :

```
kubectl get pods
```

```
D:\multi-module-app>kubectl get pods
NAME                                READY   STATUS              RESTARTS   AGE
api-gateway-7c4c9549f9-kmxnz       0/1     ContainerCreating   0          39s
service-a-5bd6cfb55f-99p5l        0/1     ContainerCreating   0          38s
service-a-5bd6cfb55f-mjw6l        0/1     ContainerCreating   0          38s
service-b-76767b9c5f-gmgb1        0/1     ContainerCreating   0          38s
service-b-76767b9c5f-rvglb        0/1     ContainerCreating   0          38s
```

Шаг 5: Проверка работы приложения

5.1. Настройка доступа через Ingress

Сделаю оговорку: деплой я провожу в minikube из-за того, что не имею доступа к машине, на которой настроил k8s-кластер. Кроме необходимости ставить Ingress через Helm, деплой отличаться не будет.

Настроим Ingress:

5.1.1. Получение IP-адреса Minikube

```
minikube ip
```

5.1.1. Получение IP-адреса Minikube

Не забудем поставить Ingress-аддон

```
minikube addons enable ingress
```

6.2. Проверка работы сервисов

6.2.1. Запрос к service-a через api-gateway

Ожидаемый ответ:

```
Hello from Service A. Pod: service-a-xxxxxxxxxx-xxxxx
```

6.2.2. Запрос к service-b через api-gateway

```
bash
```

```
curl http://myapp.local/service-b/hello
```

Ожидаемый ответ:

```
CSS
```

```
Hello from Service B. Pod: service-b-xxxxxxxxxx-xxxxx
```

Дополнительные сведения

1. Обновление образов

Если вы внесете изменения в код и захотите обновить образы в Docker Hub:

1. Пересоберите образы:

```
bash
```

```
# docker build -t yourusername/service-a:latest . docker build -t yourusername/service-b:latest .  
docker build -t yourusername/api-gateway:latest .
```

2. Перезагрузите образы в Docker Hub:

```
bash
```

```
docker push yourusername/service-a:latest docker push yourusername/service-b:latest docker push yourusername/  
/api-gateway:latest
```

3. Обновите деплойменты в Kubernetes:

- Добавьте или обновите поле `imagePullPolicy` в манифестах Deployment:

```
yaml
```

```
imagePullPolicy: Always
```

- Примените манифесты:

```
bash
```

```
kubectl apply -f k8s/
```

- Или перезапустите поды вручную:

```
bash
```

```
kubectl rollout restart deployment/service-a kubectl rollout restart deployment/service-b kubectl  
rollout restart deployment/api-gateway
```

2. Доступ к приватным образам

Если вы решили сделать образы приватными в Docker Hub:

- Создайте секрет Docker Registry:

```
bash
```

```
kubectl create secret docker-registry regcred \ --docker-server=https://index.docker.io/v1/ \ --docker-  
username=yourusername \ --docker-password=yourpassword \ --docker-email=youremail@example.com
```

- Добавьте секрет в деплойменты:

```
yaml
```

```
spec: imagePullSecrets: - name: regcred
```

Преимущества использования Docker Hub

- **Универсальность:** Образы доступны из любого места, где есть доступ к Docker Hub.
- **Упрощение процесса сборки:** Нет необходимости переключаться на Docker-демон Minikube.
- **Совместная работа:** Легко делиться образами с другими членами команды.