Конфигурирование манифестов модуля

immediate

1 марта 2025 г.

Среди компонентов модуля должны быть как минимум следующие:

1. *box* для хранения данных
2. *api* для вызова расчётов через API
3. *mlcmp* для обработки расчётов
4. *files* для файлового API

Для более сложного взаимодействия и для модулей, состоящих из нескольких функций, может быть подготовлено больше компонентов, а некоторые компоненты могут повторяться. Далее рассмотрен простейший вариант.

Результат: Подготовлены и добавлены в репозиторий файлы манифестов.

1 Компонент *box* для хранения данных

```
apiVersion: "unified-platform.cs.hse.ru/v1"
kind: DataBox
metadata:
  name: mymodule-data-box
  namespace: pu-username-pa-bm99
spec:
  s3DefaultStorage: {}
---

apiVersion: "unified-platform.cs.hse.ru/v1"
kind: DataBox
metadata:
  name: users
  namespace: pu-username-pa-bm99
spec:
  s3DefaultStorage: {}
```

По умолчанию выделяется 1ГБ места под DataBox. Если нужно больше места, можно добавить следующее:

```
spec:
  s3DefaultStorage:
    capacity: 10G
```

2 Компонент *api* для вызова расчётов через API

```
apiVersion: "unified-platform.cs.hse.ru/v1"
kind: APIComponent
metadata:
  name: somename-api
  namespace: pu-username-pa-bm99
spec:
  published: true
  mlComponent:
    name: somename-mlcmp
  restfulApi:
    path: <some-endpoint>
    auth:
      basic:
        credentials: appname-apis-cred
      identityPassThrough: true
```

Здесь,

- *spec/mlComponent* - название ML-компонента, доступ к которому предоставляет API-компонент.
- *spec/restfulAPI/path* - имя API при вызове. Например в следующей ссылке вызова ML-компонента этот пункт манифеста отвечает за часть ссылки *endpoint*. Можно выбрать произвольное имя.

  ```
  https://platform-dev-cs-hse.objectoriented.ru/pu-username-pa-bm99/
  endpoint/predict
  ```

- *spec/restfulAPI/auth* - компонент доступа пользователей к вызову API. Доступ настраивается с помощью компонента *appname-apis-cred*. Для настройки доступа нужно в приложении определить компонент *appname-apis*, что описано в пунктах 7.2 и 7.6.

3 Компонент *mlcmp* для обработки расчётов

```
apiVersion: "unified-platform.cs.hse.ru/v1"
kind: MLComponent
metadata:
  name: somename-mlcmp
  namespace: pu-username-pa-bm99
spec:
  image:
    existingImageName: registry-platform-dev-cs-hse.objectoriented.ru/
    lab-name/bm99-module-container-name:12ab345
  resourceLimits:
    cpu: 500m
    memory: 256M
  env:
    - name: CUSTOM_PARAMETER_1
      value: "3.14"
    - name: ANOTHER_PARAMETER
      value: "some_value"
  mlService:
    packageRegistryName: python-package-registry
    inference:
      fileExchange:
        fileBox: user-box
        inferenceFilesPath: /home/appname/users/tmp
      model:
        modelBox: model-box
        modelPath: /home/path/to/model.joblib
      entryPoint:
        pythonPath: .
        pythonFunction: modulename.predict.inference
  connectedBoxes:
    - name: model-box
      path: /home/path/to
      mountS3Box:
        subPath: users/developer/file_groups/models
        s3BoxName: mymodule-data-box
    - name: user-box
      copyS3Box:
        s3BoxName: users
```

Здесь,

- *metadata/name* - название ML-компонента. Оно используется в API-компоненте и может использоваться в других обращениях к компоненту через kubernetes.
- *spec/image/existingImageName* - название образа, из которого собирается ML-компонент. Этот образ был подготовлен в пункте 5.
- *spec/image/resourceLimits* - ограничения по ресурсам для ML-компонента.
  - Ресурсы процессора *cpu* измеряются в ядрах. *500m* - 500 миллиядер, то есть половина ядра процессора занята развёрнутым ML-компонентом.
  - Ресурсы оперативной памяти *memory* могут измеряться в мегабайтах, гигабайтах и других метриках памяти
  - Подробнее - `https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/`
- *spec/env* - переменные окружения, передаваемые в контейнер. Вы можете вынести настройки контейнера в переменные окружения и задавать их в ML-компоненте, чтобы не нужно было изменять код и пересоздавать образ Docker с новыми значениями.
- *spec/mlService*
  - *packageRegistryName* - название репозитория с пакетами Python, должно соответствовать «внешнему» названию из компонента приложения в пункте 7.6. Название может быть другим, главное - соответствие названий друг другу.
  - *inference/fileExchange* - соединение файловой системы контейнера (сервиса) и ящика S3 для обмена файлами.
    - *fileBox* - название ящика S3 для файлов пользователей. Должно соответствовать названию из пункта *connectedBoxes* ниже в этом файле, а не названию *metadata/name* из компонента *box*.
    - *inferenceFilesPath* - путь, куда будут записываться создаваемые файлы внутри контейнера. К этому пути должны быть права доступа у пользователя контейнера. С этой папкой фреймворк взаимодействует автоматически, то есть лучше сделать уникальный путь, который нигде больше не используется.
  - *inference/model* - соединение файловой системы контейнера (сервиса) и ящика S3 для подключения модели расчётов.
    - *modelBox* - название ящика S3, где размещена модель. Должно соответствовать названию из пункта *connectedBoxes* ниже в этом файле, а не названию *metadata/name* из компонента *box*.
    - *modelPath* - путь, по которому модель будет доступна внутри контейнера. Этот путь передаётся в функцию *inference*. Может быть папкой или файлом.
  - *inference/entryPoint*

    * *pythonPath* - относительный путь от корня репозитория к папке, которая будет добавлена в *PYTHONPATH*.
    * *pythonFunction* - имя функции-адаптера из пункта 3, включая имена промежуточных модулей.

- *spec/connectedBoxes* - подключенные компоненты *box* (ящики)
  - Первый ящик в примере используется для доступа к модели.
    * *name* - имя, по которому к ящику можно обращаться выше в пункте *spec/mlService/inference/model/m*
    * *path* - путь внутри контейнера, к которому монтируется ящик.
    * *mountS3Box/s3BoxName* - название компонента *box* из пункта 6.1.
  * *mountS3Box/subPath* - путь внутри ящика, который монтируется к пути *path*. Здесь используется путь *users/developer/file*$_g$*roups/models*$S3.APIFilesUSER$
  - Второй ящик в примере используеся для взаимодействия с пользователями
    * *name* - имя, по которому к ящику можно обращаться выше в пункте *spec/mlService/inference/fileExcha*
    * *copyS3Box/s3BoxName* - назание компонента *box* из пункта **6.1**.

4 Компонент *files* для файлового API

```
apiVersion: "unified-platform.cs.hse.ru/v1"
kind: APIComponent
metadata:
  name: files-api
  namespace: pu-username-pa-bm99
spec:
  published: true
  files:
    enabled: true
  restfulApi:
    auth:
      basic:
        credentials: bm99-apis-cred
      identityPassThrough: true
```

Здесь,

- *metadata/name* - название компонента, не используется
- *spec/restfulApi/auth/basic/credentials* - аналог реквизитов доступа из компонента API для вызова расчётов.

Остальные пункты, кроме названия компонента, не настраиваются.

Endpoint файлового API - *files*. Вызывать его можно по ссылкам вида:

```
https://platform-dev-cs-hse.objectoriented.ru/pu-username-pa-bm99/
files/box-name/path/in/box
```