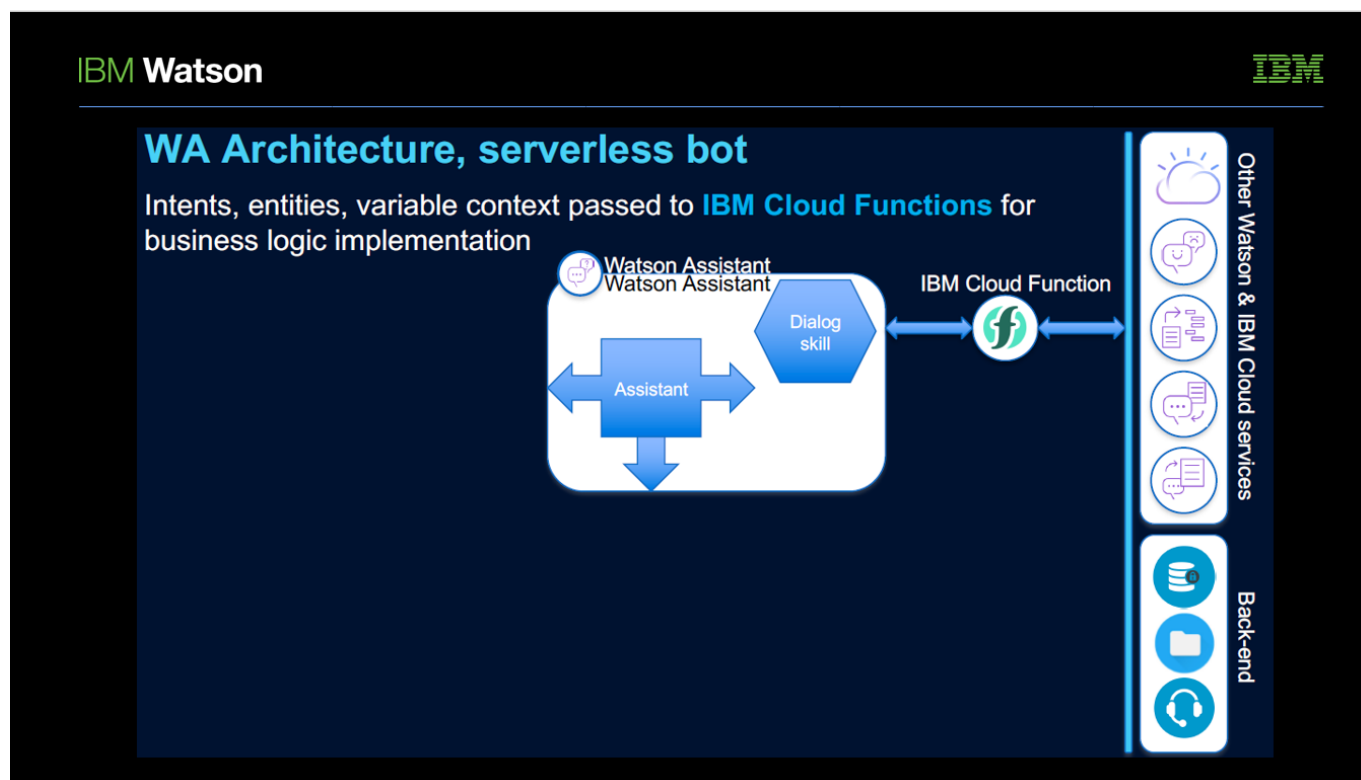


LAB-03. IBM Cloud Functions. Построение DEVOPS процесса разработки, тестирования, deployment

- 1. Цель работы
- 2. Инструменты, необходимые для выполнения работы
- 3. Краткое описание процесса
- 4. Выполнение работы.
- 4.1. Подготовительные работы
- 4.2. Ознакомиться с каталогами и файлами пакета bankapi-customer
- 4.3. Тестирование работоспособности функций локально
- 4.4. Создание toolchain в IBM Cloud.

Цель работы

Интеграция WatsonAssistant с различными сервисами проще всего реализовать через IBM Cloud functions.



pic-0

После написания первых двух функций у меня возникло желание построить DEVOPS процесс разработки и deployment функций в облако. Целью этой работы является построение DEVOPS процесса deployment IBM Cloud Function в облако.

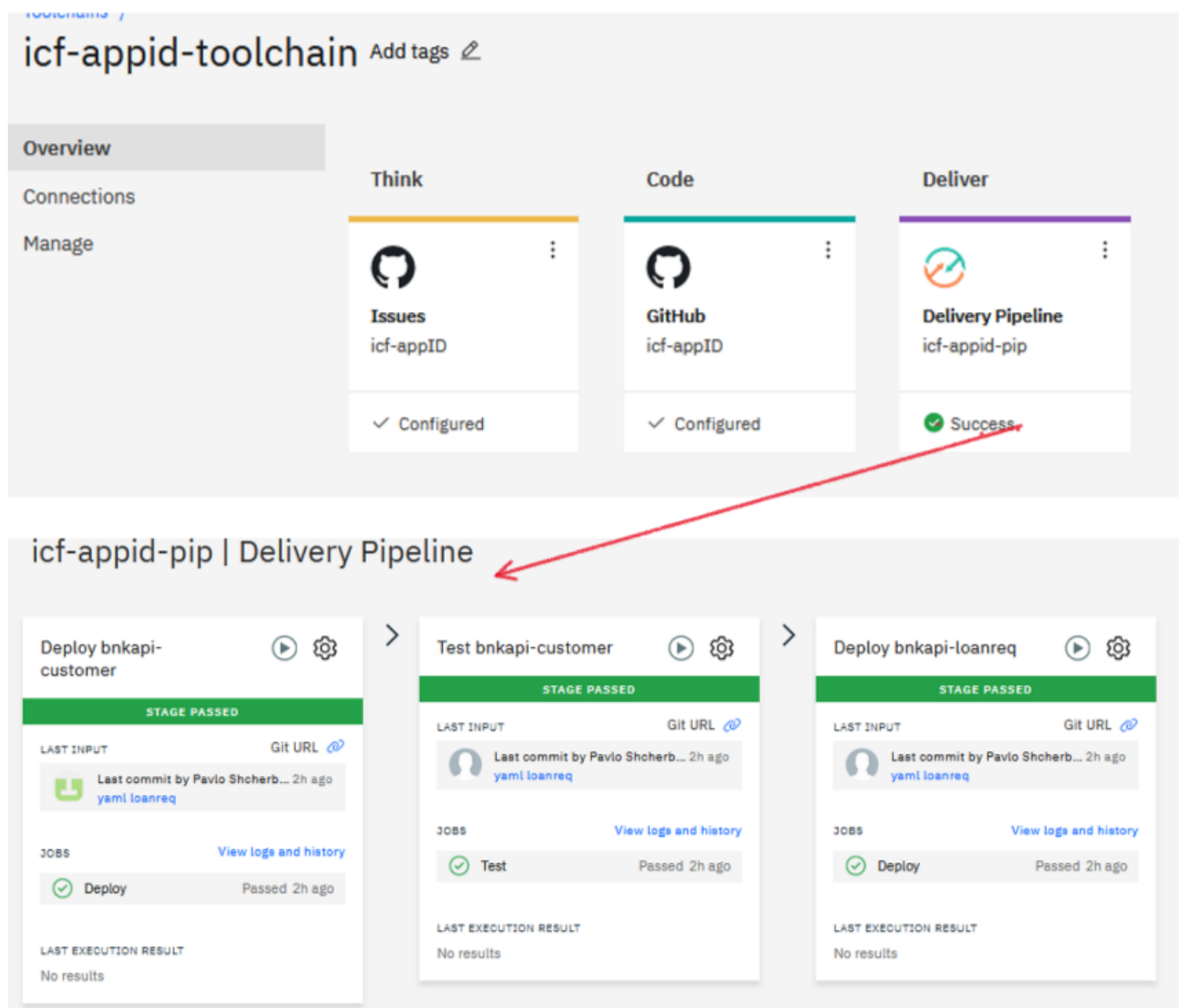
В результате выполнения работы будет построен DEVOPS процесс по deployment IBM Cloud functions в IBM Cloud из github репозитория.

Инструменты, необходимые для выполнения работы

- Нужно иметь открытый Account в [IBM Cloud](#)
- Account на [github](#)
- Установленную Node.js-10 для отладки и запуска приложений локально
- Установленный популярный редактор исходного кода [Visual Studio Code](#) с плагинами для Node.js

Краткое описание процесса

Работа с IBM Cloud Functions выполняется через [специальный CLI](#). Команды CLI позволяют выполнить deployment, запуск функций на выполнение, модификацию параметров запуска, настройка вызова функций через Web API и еще много чего. Но очень хотелось построить процесс так, чтобы разработать и отладить функции локально. Потом сгруппировать функции в пакет. Отправить все содержимое пакета в GitHub и по commit получить развертывание всех функций пакета в IBM Cloud, как на pic-1.



pic-1

Еще на этапе разработки можно создать конфигурационный YAML - файл, который позволит в нем описать контекст вашего пакета, отправить все это вместе с исходным кодом в github и потом использовать для deployment

На этапе разработки процесса приняты такие соглашения:

- Весь проект server less app разбит на каталоги с названиями отображающими названия пакетов.
- в корне каждого пакета создается package.json с перечнем пакетов Node.js, которые обеспечивают test-driven development и прикладные пакеты, необходимы для работы функций пакета

```
{
  "name": "bnkapi-customer",
  "version": "1.0.0",
  "description": "ibm-cloud functions sources",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "dependencies": {
    "axios": "^0.19.2",
    "chai": "^4.2.0",
    "mocha": "^8.0.1",
    "needle": "^2.5.0",
    "querystring": "^0.2.0",
    "supertest": "^4.0.2"
  }
}
```

- В корне пакет создается uml-файл с названием пакета, в котором описывается состав пакет:

```
packages:
  bankapi-customer:
    version: 1.0
    license: Apache-2.0
    actions:
      getCustomers:
        function: src/getCustomers.js
        runtime: nodejs
        inputs:
          custname: string
          custid: string
          custtin: integer
          height: float
        outputs:
          greeting: string
          details: string
        web-export: false
      getCustomerById:
        function: src/getCustomerByID.js
```

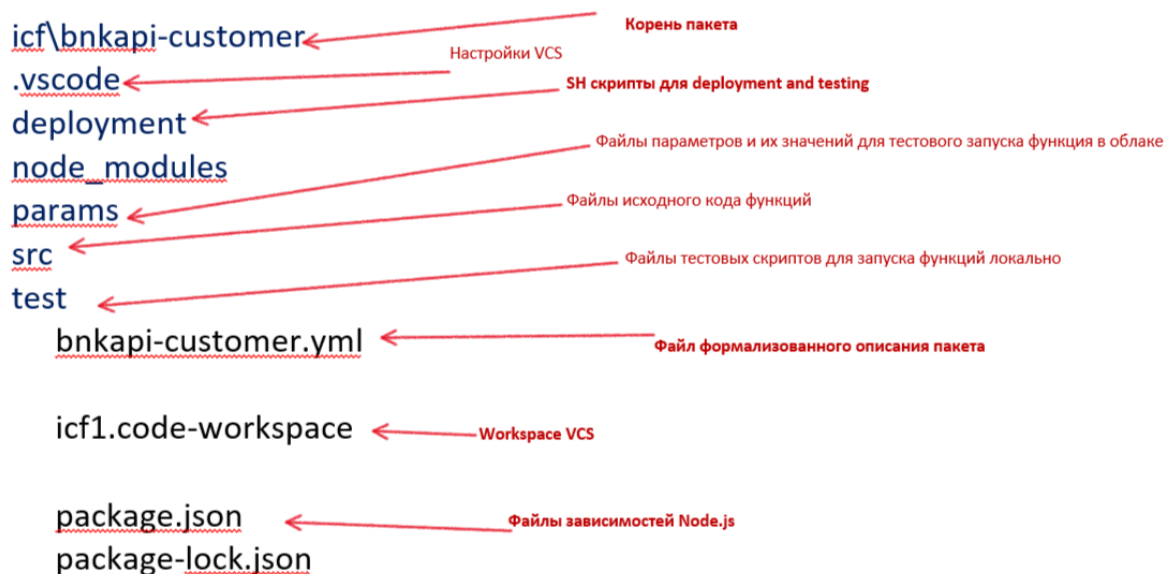
```

runtime: nodejs
inputs:
  cust_id: number
  web-export: false
createCustomer:
  function: src/createCustomer.js
  runtime: nodejs
  inputs:
    first_nm: string
    last_nm: string
    email: string
    tin: string
    phone: string
  outputs:
    idcust: integer
    web-export: false

```

Назначение каталога показано на рисунке pic-2

Структура каталогов пакета: **bnkapi-customer**



pic-2

Выполнение работы.

Подготовительные работы

Выполнить fork github репозиторий в ваш github account. Потом клонировать его на локальную станцию.

перейти в каталог `./bankapi-customer` и выполнить установку пакетов Node.js путем выполнения команды

```
npm install
```

У вас создается каталог `./node_modules` и создадутся пакеты зависимостей Node.js

Ознакомиться с каталогами и файлами пакета `bnkapi-customer/`

В пакете, в каталоге `./src` создано 3 функции, по работе с клиентами:

```
createCustomer.js - Создать клтента
getCustomerByID.js - получить клиента по его id
getCustomers.js - получить список всех клиентов
```

Эти функции по API вызываю облачный API Gateway.

В корне каталога в `yaml`-файле выполнено описание (связывание) `action` пакета с исходным кодом функций.

В каталоге `./test` описаны тест кейсы для тестирования и отладки функций локально.

Тестирование работаспособности функций локально

С помощью VCS запустить в отладчике файл `./test/test-getCustomer.js`

В результате выполнения получим:

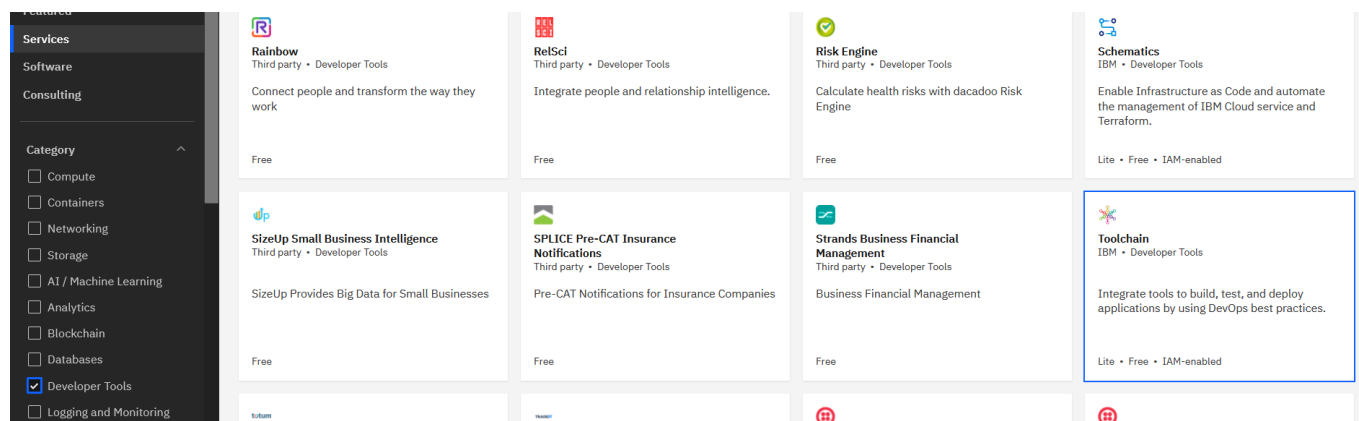
```
Debugger attached.
```

```
Test customer API
bnkapi-customer/getCustomers: Старт
bnkapi-customer/getCustomers: Вызываю сервис: [https://a7275984.eu-
gb.apigw.appdomain.cloud/bnkapi-cust/customer]
bnkapi-customer/getCustomers: Сервис вернул успешный статус: [200]
bnkapi-customer/getCustomers: Возвращаю ответ
  ✓ GetCustomers: Expect customers list (594ms)
Waiting for the debugger to disconnect...
```

То есть локально функция работает. Функцию можно протестировать и отладить. Удобно разработчику.

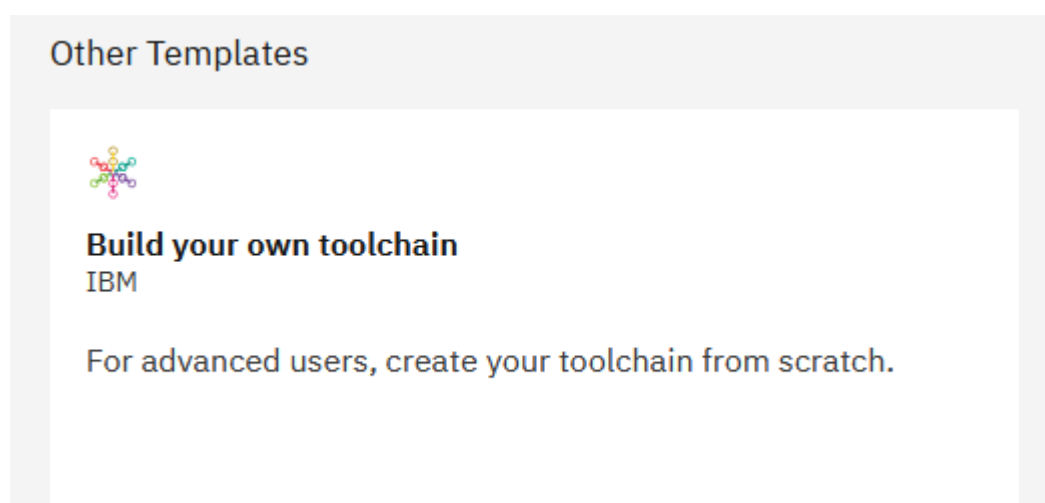
Создание toolchain в IBM Cloud.

- Подключить сервис TollChain из каталога сервисов IBM



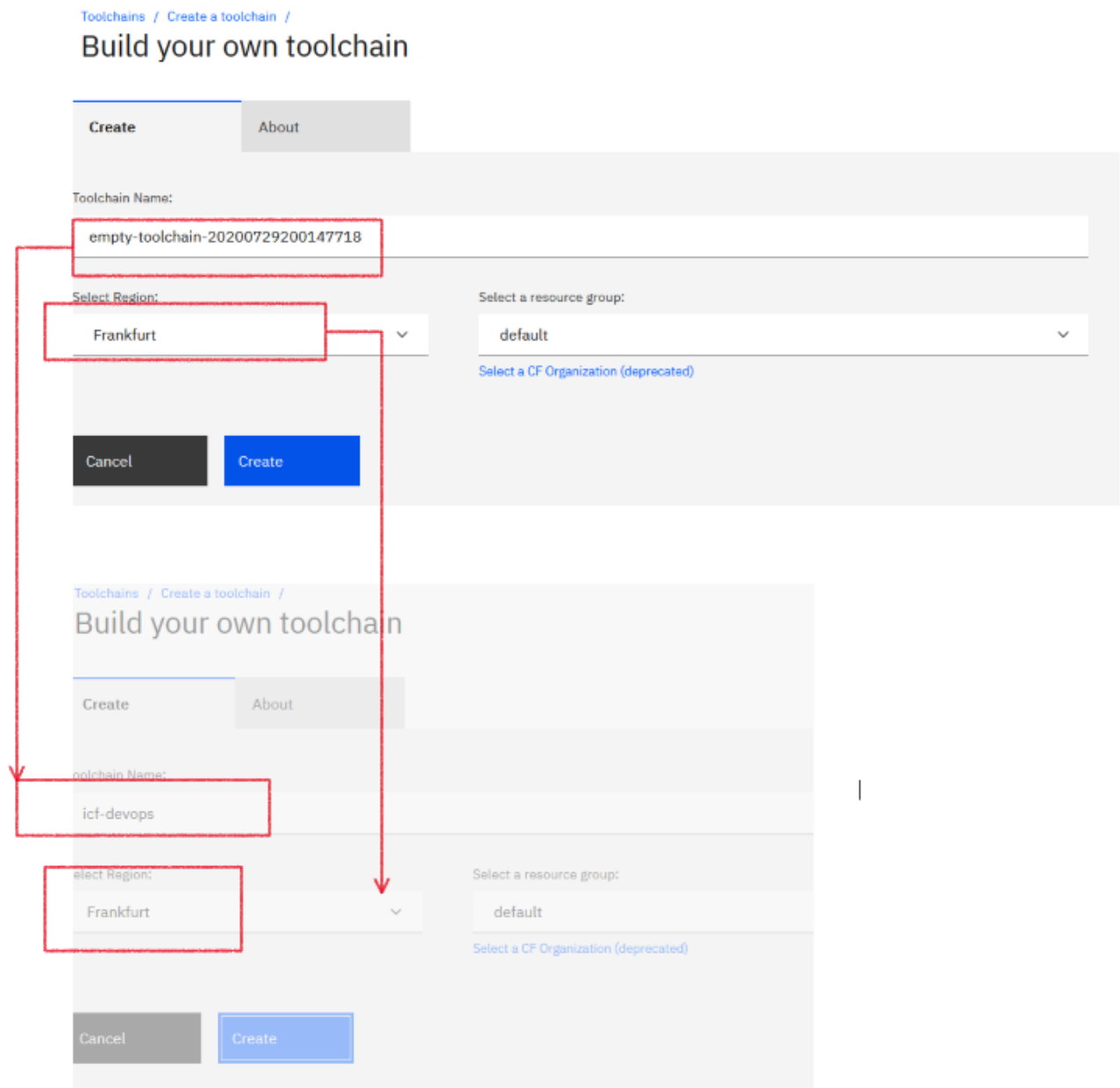
pic-3

- Выбрать вариант создания "Your own toolchain"



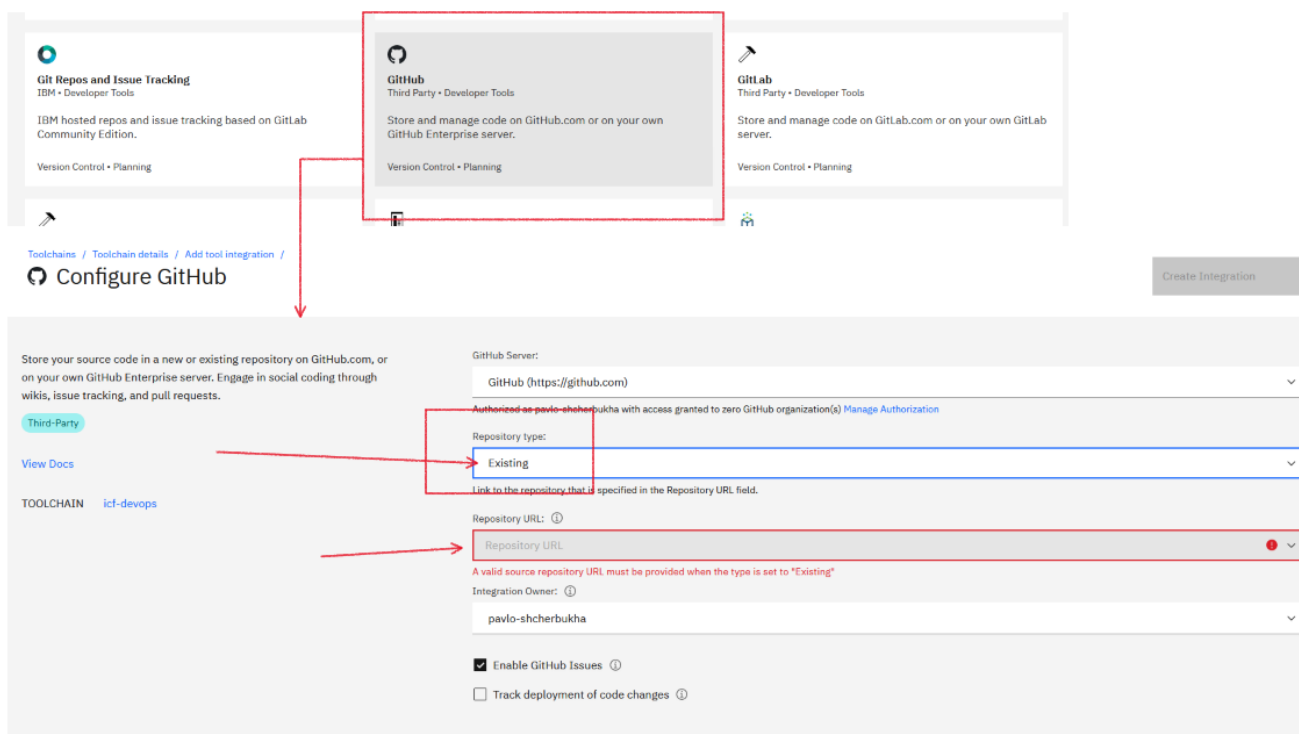
pic-4

Присвоить toolchain имя и выбрать регион: pic-5



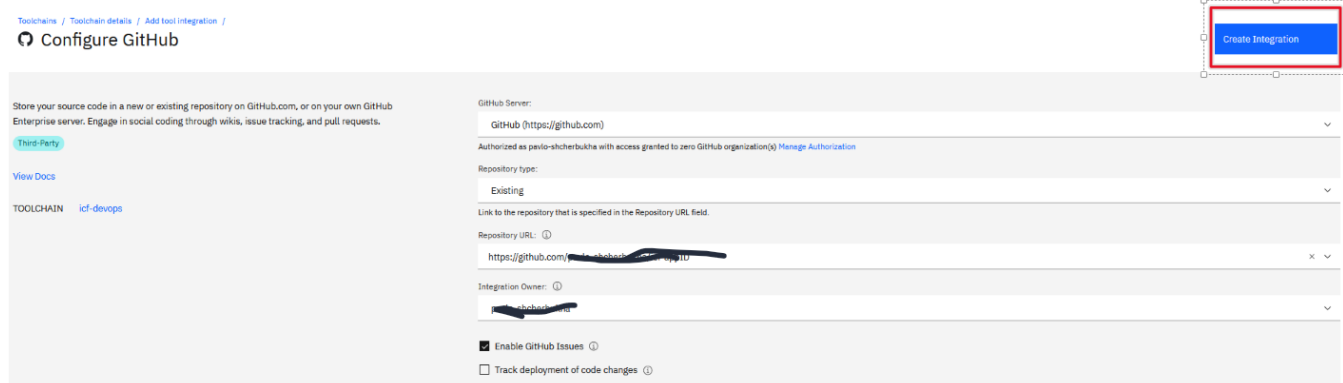
pic-5

- Создать интеграцию с GitHub



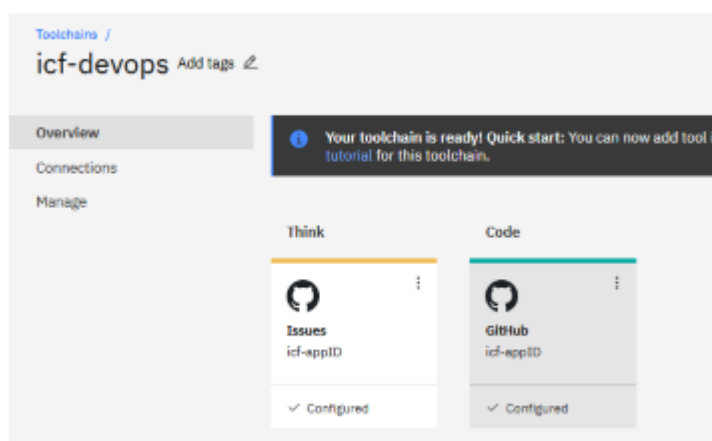
pic-6

Заполнить параметры подключения к github и нажать "create"



pic-7

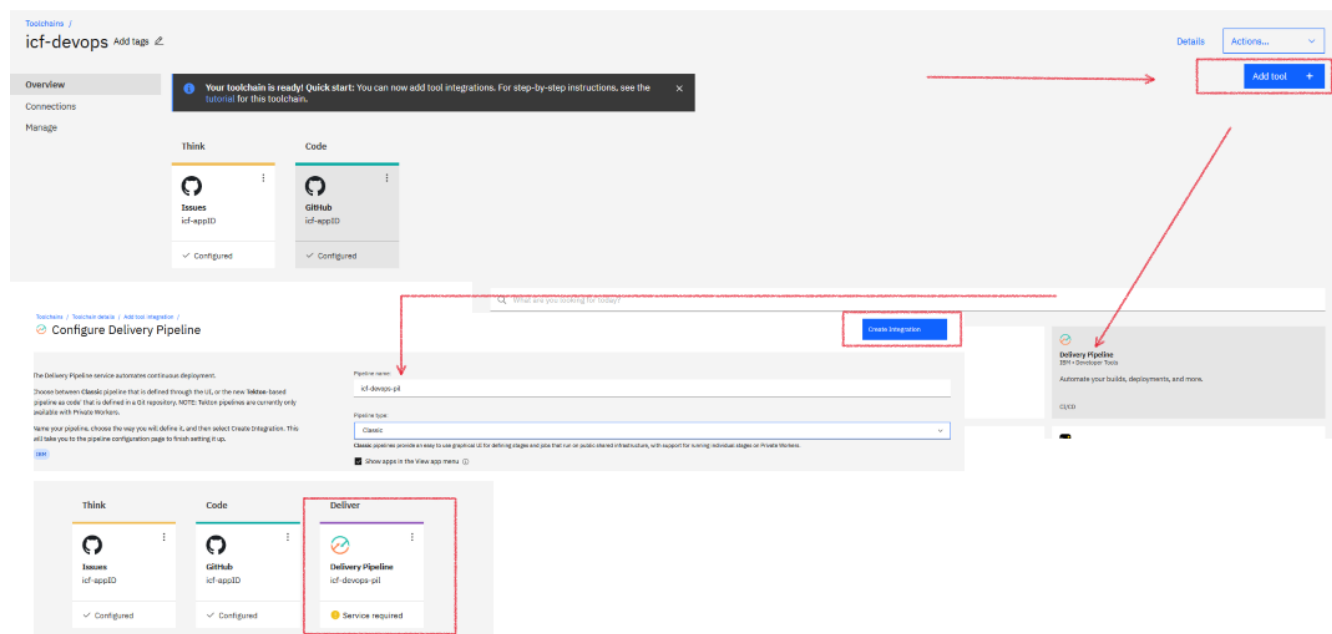
В результате получим первую часть toolchain: pic-8.



pic-8

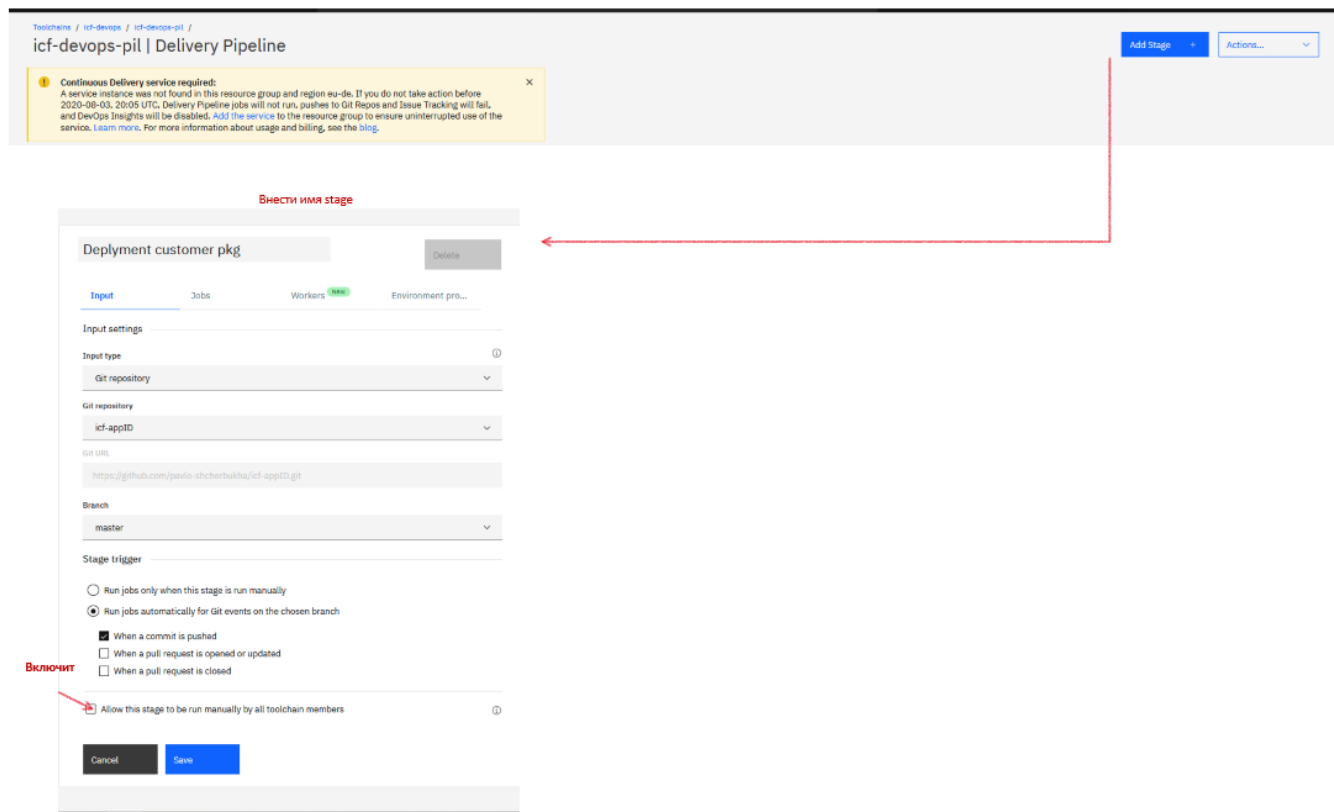
- Подключить инструмент: DeliveryPipeline

На pic-9 показаны все шаги по выполнению подключения и получению результата.



pic-9

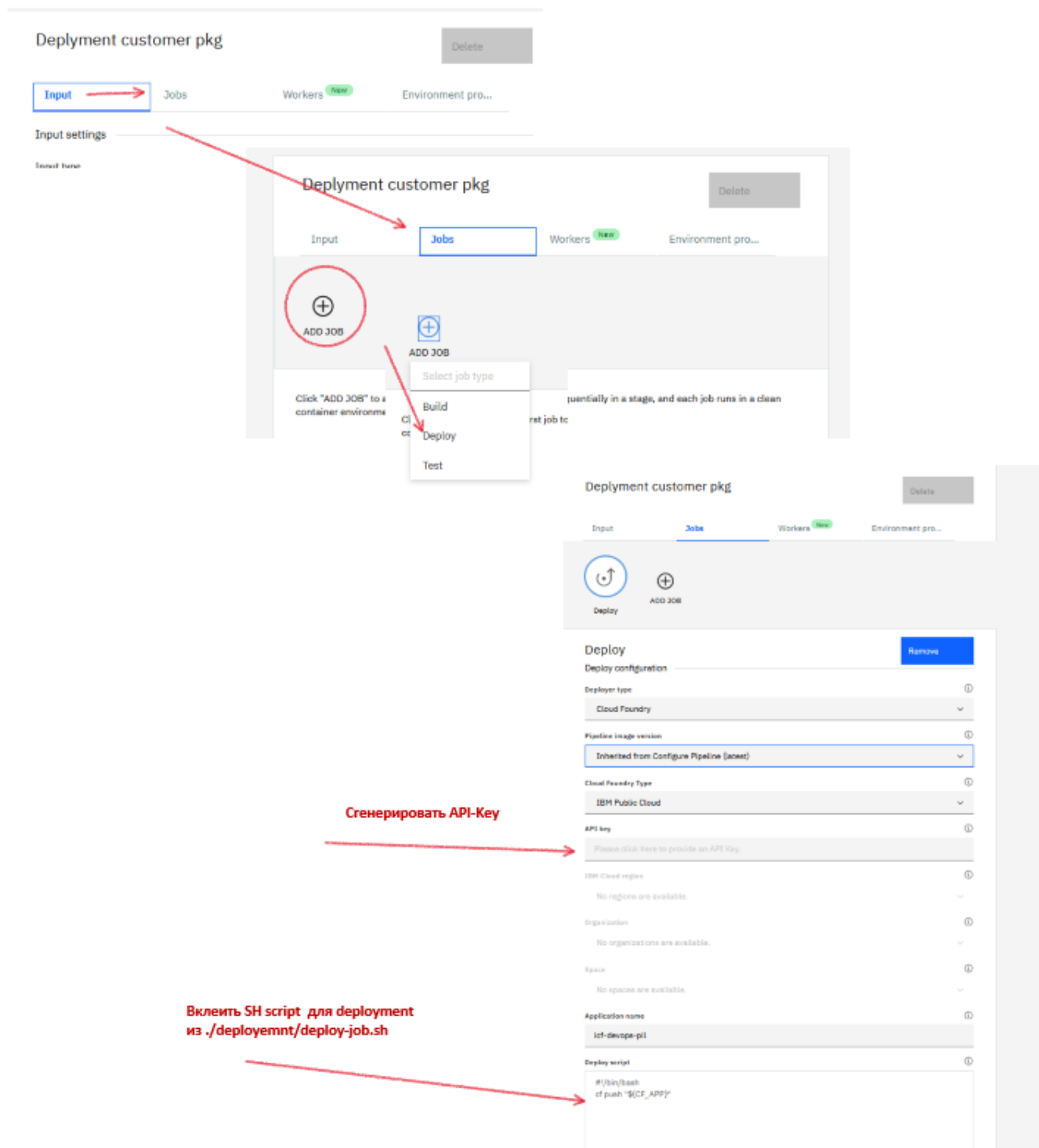
- Сконфигурировать Delivery Pipeline Для этого кликнуть на иконке pipeline и потом на кнопке "Add Stage": pic-10



pic-10

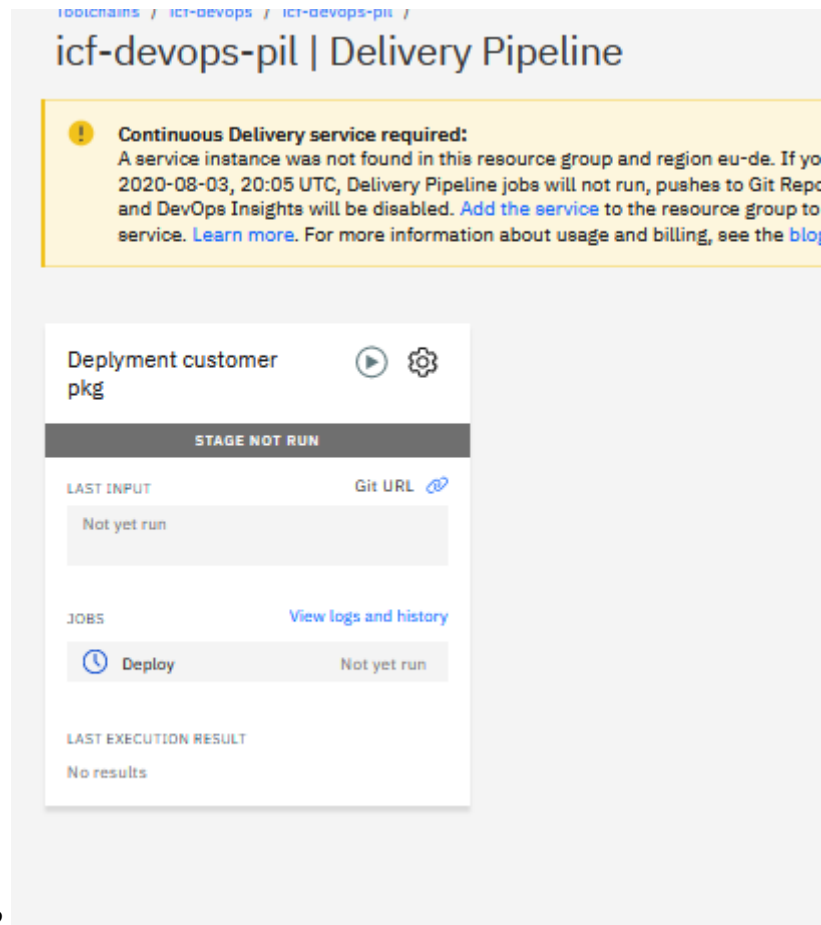
- Сконфигурировать deployment-stage

Для этого необходимо переключиться на вкладку jobs и добавить deployment job: pic-11.



pic-11

Сгенерировать api-key проверить правильное указание региона и других параметров. В поле sh-скрипта вклеить deployment скрипт из файла: ./deployment/deploy-job.sh и сохранить.



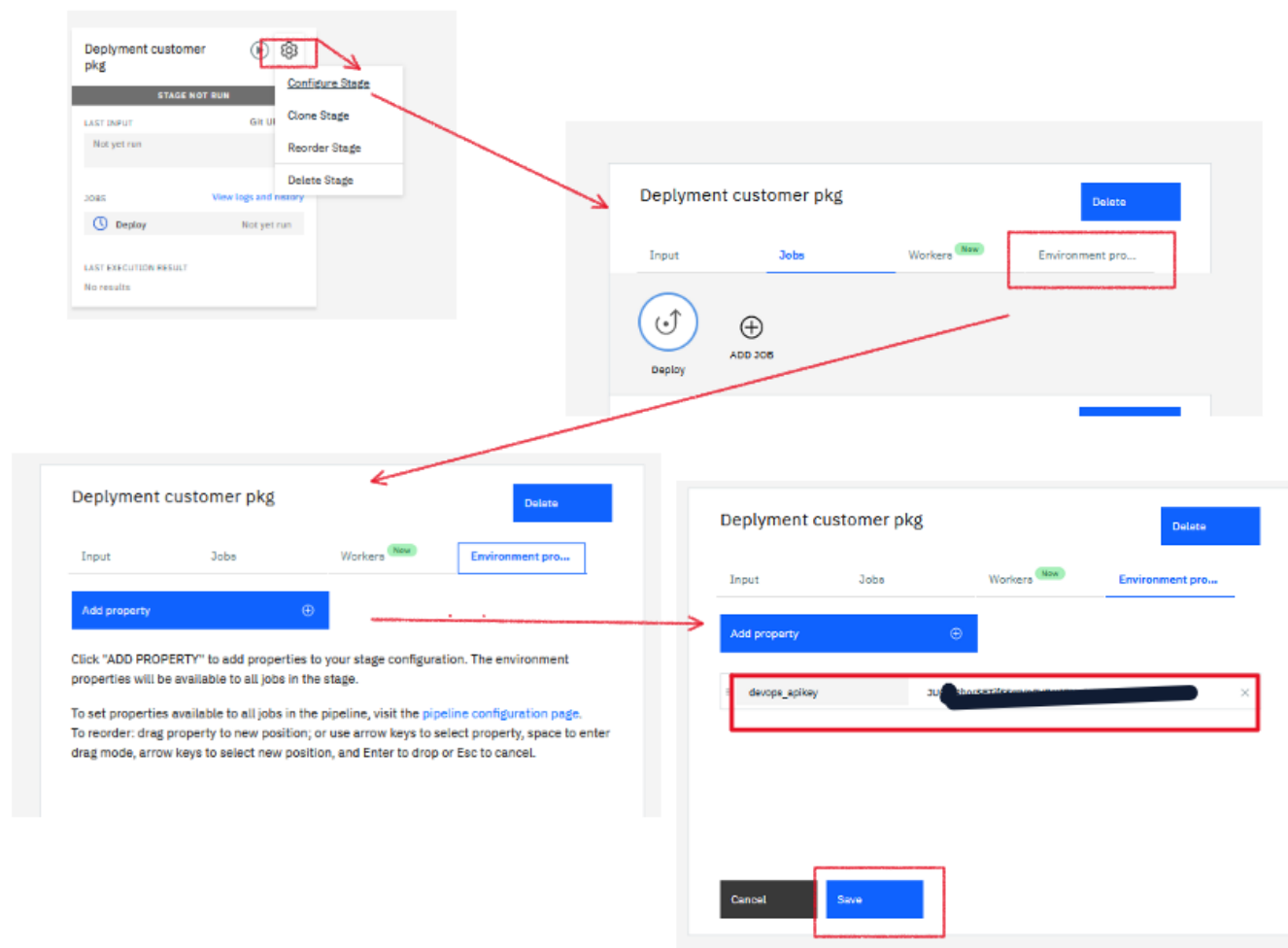
Результат показан на pic-12

pic-12

В деплоймент script в строке **ibmcloud login** указана environment переменная **--apikey \$devops_apikey**,

```
ibmcloud login -a cloud.ibm.com --apikey $devops_apikey -r eu-gb -g default -o
panama -s shdev
cd ./bnkapi-customer
ibmcloud fn deploy --manifest bnkapi-customer.yml
ibmcloud fn package list
ibmcloud fn action list
```

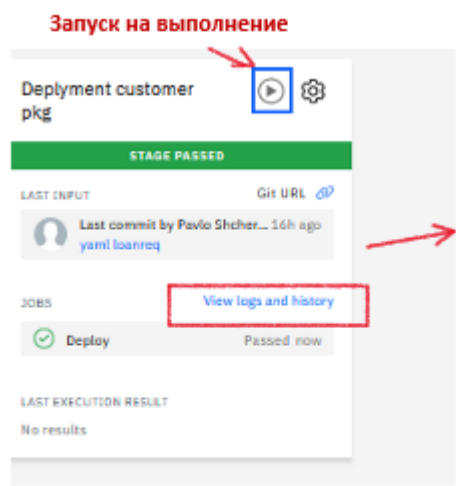
Для внесения env-переменной нужно выполнить операции показанных на pic-13.



pic-13

Запускаем вручную процесс первый раз для проверки pic-14:

Запуск на выполнение



Deploy customer pkg

STAGE PASSED

LAST INPUT Git URL [🔗](#)

Last commit by Pavlo Shcher... 16h ago
yami loanreq

JOBS

Deploy Passed now

LAST EXECUTION RESULT

No results

[View logs and history](#)

```

API endpoint: https://cloud.ibm.com
Region: eu-gb
User: [REDACTED]
Account: [REDACTED]
Resource group: default
CF API endpoint: https://api.eu-gb.cf.cloud.ibm.com (API version: 2.148.0)
Org: [REDACTED]
Space: [REDACTED]
Success: Deployment completed successfully.
packages
/panama_shdev/bankapi-customer private
/panama_shdev/bankapi-loanreq private
/panama_shdev/get-http-resource1 private
/panama_shdev/get-http-resource private
/panama_shdev/appid-be private
/panama_shdev/appbe private
/panama_shdev/Hsgs private
/panama_shdev/Lab01-1 private
/panama_shdev/assistant-v1 private
actions
/panama_shdev/bankapi-customer/getCustomers private nodejs:10
/panama_shdev/bankapi-customer/createCustomer private nodejs:10
/panama_shdev/bankapi-customer/getCustomerById private nodejs:10
/panama_shdev/bankapi-loanreq/createLoanreq private nodejs:10
/panama_shdev/bankapi-loanreq/getLoanreqById private nodejs:10
/panama_shdev/bankapi-loanreq/getLoanreqs private nodejs:10
/panama_shdev/appid-be/getUserProfile private blackbox
/panama_shdev/get-http-resource/location private nodejs:10
/panama_shdev/get-http-resource/location private nodejs:10
/panama_shdev/Hsgs/SendSMS private nodejs:10

```

pic-14