

LAB-02. IBM Cloud Functions. Розробка та deployment функцій з використанням інтерфейсу командного рядка

- 1 [Вступ](#)
- 2 [Знайомство с IBM CLI](#)
- 3 [Розробка функцій, які використовують інші http-сервіси](#)
- 4 [Використання CLI](#)

Вступ

Ця робота призначена для організації роботи розробника з використанням командного рядка IBM CLI. Для роботи будуть потрібні наступні інструменти:

- Visual Studio Code (можна інший редактор)
- IBM Cloud CLI
- plugin cloud-function к IBM Cloud CLI

Інструкція по встановленню IBM Cloud CLI и plugin cloud-function доступна по лінку: [IBM Cloud function setup CLI](#).

Знайомство з IBM Cloud CLI

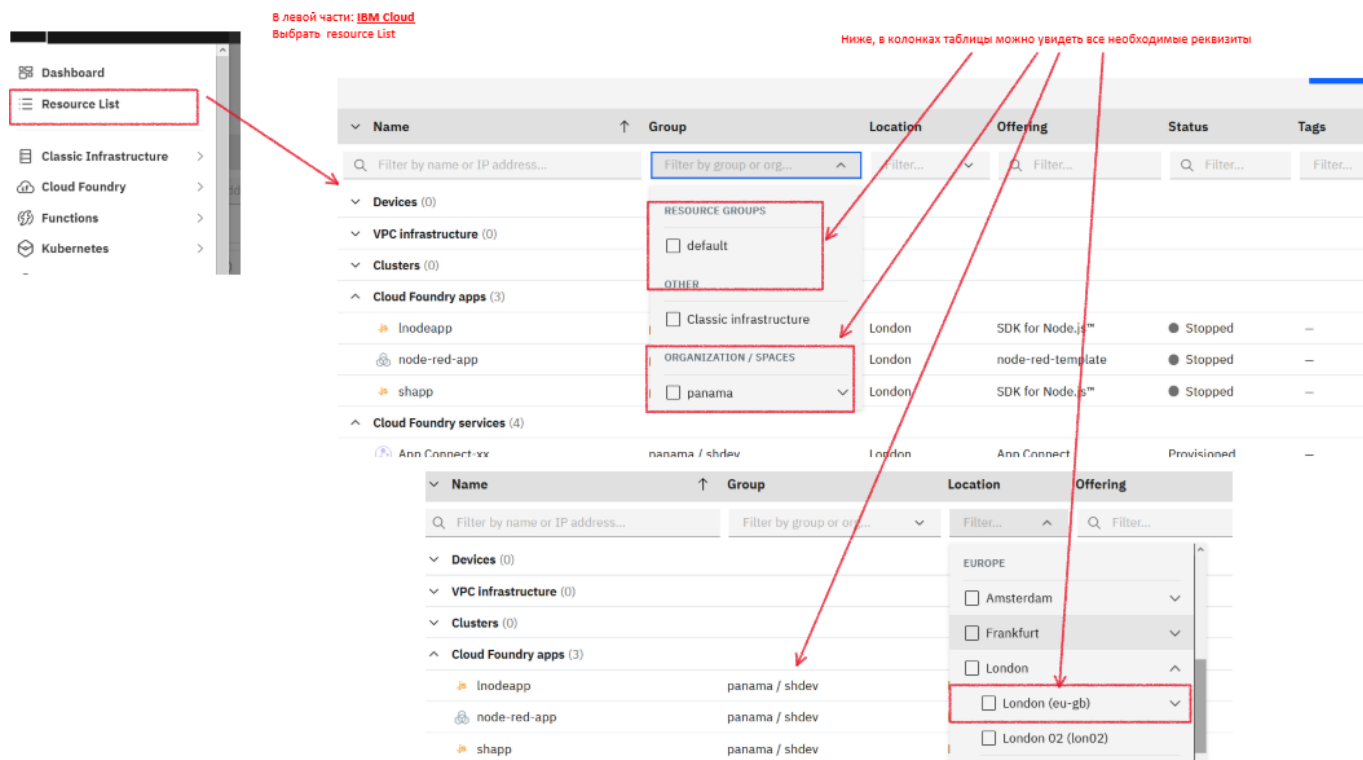
Перше, що потрібно зробити - це виконати логін з допомогою CLI в IBM Cloud

Для цього використовуємо команду: **ibmcloud login**

```
ibmcloud login -a <API endpoint> -u <user mail> -p <user password> -r <region> -g  
<resource group> -o <cloudfoundry organisation> -s <space>
```

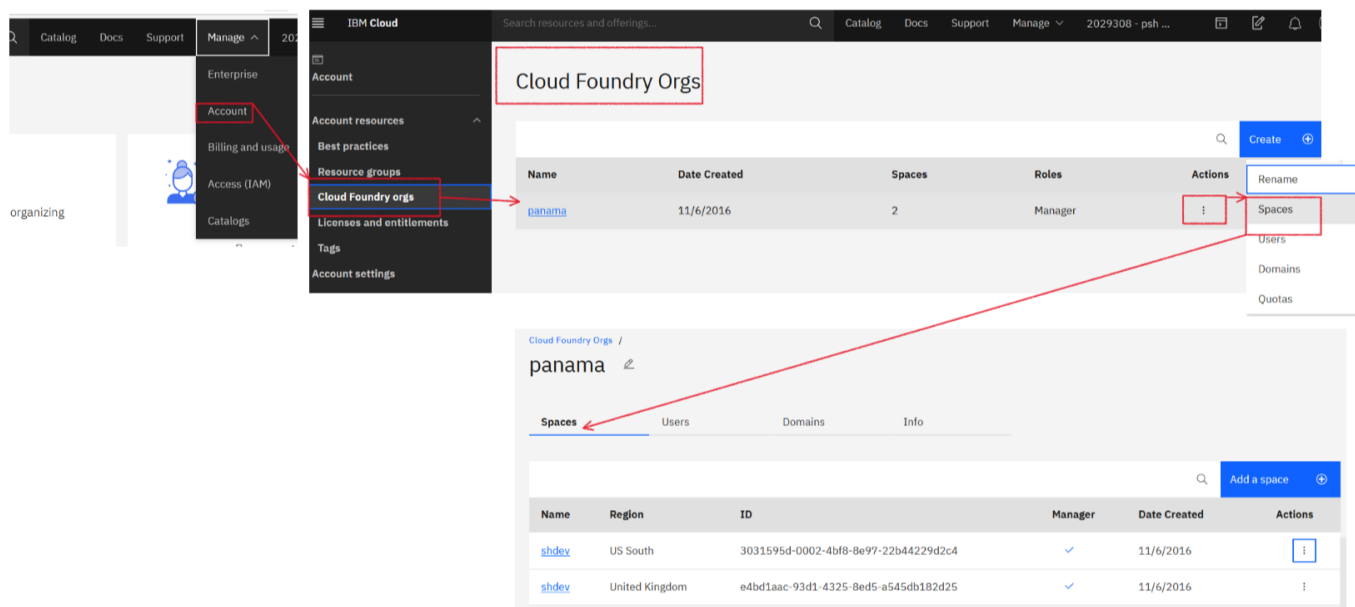
- -a API endpoint указывается <https://cloud.ibm.com>
- -u user mail вказуємо е-mail з яким користувач заходить в IBM Cloud dashboard
- -p user password вказуємо пароль користувача, з яким він заходить в IBM Cloud dashboard
- -r region вказуємо регіон, в якому будуть розгорнуті ваші сервіси
- -g resource group вказуємо групу ресурсів (якщо вона є). За замовчуванням має назву default. Можна пропустити, не обов'язковий
- -o cloudfoundry organisation
- -s space in cloudfoundry organisation

Реквізити: region, resource group, cloudfoundry organisation, space можна отримати из таблиці відображення ресурсів (pic-1)



pic-1

Або, зайти у свій account в меню: "Manage/Account/Cloud Foundry Orgs" (pic-2)



pic-2

Приклад команди:

```
ibmcloud login -a https://cloud.ibm.com -u zzzz@gmail.com -r eu-gb -g default -o panama -s shdev
```

Результат роботи команди показантй на pic-3

```

API endpoint: https://cloud.ibm.com
Password>
Authenticating...
OK

Targeted account psh: panama's Account (20201-222e9875e405230e1-07517650) <-> 2020208
Targeted resource group default
Targeted region eu-gb
Targeted Cloud Foundry (https://api.eu-gb.cf.cloud.ibm.com)
Targeted org panama
Targeted space shdev

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:              panama
Space:            shdev
PS C:\PSHDEV\PSH-WorkShops\THINK2020\repo\wa\think2020-wa>
  
```

pic-3

Вводити персональний логін та пароль кожен раз не комфортно. Більш того, це не краще рішення для виконання автоматизованих devops операцій. Тому, є можливість зайти в CLI і з допомогою згенерованого [API-KEY](#).

```
ibmcloud iam api-key-create NAME [-d DESCRIPTION] [--file FILE] [--lock]
```

- NAME (required) Найменування API key, який буде створений.
- -d DESCRIPTION (опціонально) Опис API-KEY.
- --file FILE Зберегти інформацію про API key у заданий файл на laptop.
- --lock Заблокувати API key.

Приклади: Create an API key and save to a file:

```
ibmcloud iam api-key-create MyKey -d "this is my API key" --file key_file
```

Файл **key_file** являє собой JSON.

Таким яином створений API-KEY в локальний файл и будем его использовать в дальнейших операциях. Генерируем, выковавши команду:

```
ibmcloud iam api-key-create devopskey -d "APIKEY FOR DEVOPS" --file psh-devops-key.json
```

Результат виконання команди показано на рис-4 та у файлі **psh-devops-key.json**

```
PS C:\PSHDEV\PSH-WorkShops\THINK2020\repo\wa\think2020-wa> ibmcloud iam api-key-create devopskey -d "APIKEY FOR DEVOPS" --file psh-devops-key.json
Creating API key devopskey under 23e20422b0773d105030-d0731465d as panama@gmail.com...
OK
API key devopskey was created
Successfully save API key information to psh-devops-key.json
PS C:\PSHDEV\PSH-WorkShops\THINK2020\repo\wa\think2020-wa>
```

pic-4

Теперь модифікуємо команду логін для використання в ній API-KEY. API вибирається з отриманого JSON-файла, поле: **"apikey"**

```
ibmcloud login -a https://cloud.ibm.com --apikey yourApiKey -r eu-gb -g default -o panama -s shdev
```

Результат роботи команди показано на рис-5.

```
PS C:\PSHDEV\PSH-WorkShops\THINK2020\repo\wa\think2020-wa> ibmcloud login -a https://cloud.ibm.com --apikey MY-API-KEY -r eu-gb -g default -o panama -s shdev
Warning: option -o or -s is deprecated. Use command C:\Program Files\IBM\Cloud\bin\ibmcloud.exe target -o ORG -s SPACE instead.
```

```
API endpoint: https://cloud.ibm.com
Authenticating...
OK
```

```
Targeted account panama@gmail.com (23e20422b0773d105030-d0731465d) < 2029208
```

```
Targeted resource group default
```

```
Targeted Cloud Foundry (https://api.eu-gb.cf.cloud.ibm.com)
```

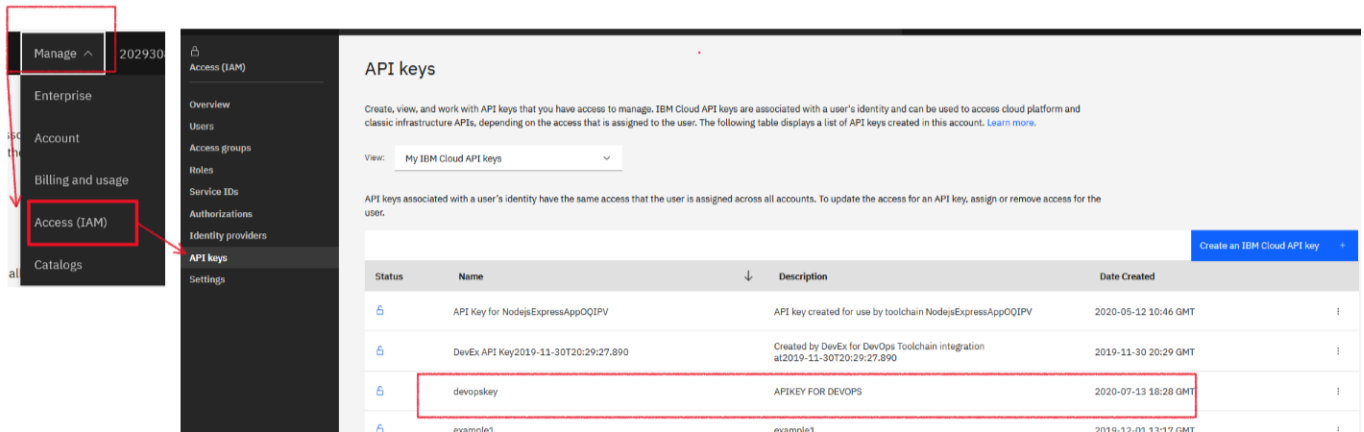
```
Targeted org panama
```

```
Targeted space shdev
```

```
API endpoint: https://cloud.ibm.com
Region: eu-gb
User: panama@gmail.com
Account: panama's Account (23e20422b0773d105030-d0731465d) < 2029208
Resource group: default
CF API endpoint: https://api.eu-gb.cf.cloud.ibm.com (API version: 2.148.0)
Org: panama
Space: shdev
PS C:\PSHDEV\PSH-WorkShops\THINK2020\repo\wa\think2020-wa>
```

pic-5

Список створених API-KEYs можна побачити у своєму account, по шляху: Manage/Access IAM/API keys.



pic-6

Разробка функцій, які використовують інші http-сервіси

В якості тестового прикладу розробимо пакет функцій які обробляють до відкритих сервісів Національного банку України для отримання курсів валют.

Так, на сторінці [API для розробників] (<https://bank.gov.ua/ua/open-data/api-dev>) присутній опис сервісу для отримання курсів валют НБУ:

Офіційний курс гривні до іноземних валют та банківських металів
 Курс на поточну дату
 Курс на дату (формат файлу Статзвітності #99), дата задається у форматі: ddmmuuuu, де dd - день, mm - місяць, uuuu - рік
 Курс на дату, дата задається у форматі: uuuummdd, де uuuu - рік, mm - місяць, dd - день
 Курс на дату по валюті (код валют літерний, реєстр значення не має)

Примітка: Поточного дня буде відображатися офіційний курс гривні до іноземних валют, встановлений НА ЗАВТРА за схемою:

До 16:00 – відображається лише офіційний курс гривні до іноземних валют, що встановлюється 1 раз на місяць.

Після 16:00 - офіційний курс, зазначений у п.1, та офіційний курс гривні до іноземних валют, що встановлюється щодня.

Інструкція до сервісу отримання курсів гривні до іноземних валют з першоджерела в момент підписання

Довідкове значення курсу гривні до долара США на 12:00, дата задається у форматі uuuummdd, де uuuu - рік, mm - місяць, dd - день

Необхідно розробити набір функцій, які будуть отримувати курси НБУ:

- Поточний курс НБУ по всіх валютах
- Курс НБУ за всіма валютами за вказану дату
- Курс НБУ по вказаній валюті за зазначену дату

- Функцію конвертації суми в одній валюті в суму в іншій валюті за курсом НБУ за задану дату. Функції необхідно помістити в один пакет, так як логічно вони представляють єдиний сервіс.

Вихідний код функцій нах'одітся в каталозі **. / NbuServices **

- GetCurrentExchRate.js - отримання поточного курсу валют по всім валютам;
- GetExchRateByDate.js - отримання курсу по всіх валютах за задану дату;
- GetExchRateByCCY.js - отримання курсу по вказаній валюті за вказану дату;
- GetConvertCCY.js - конвертація валют по курсу НБУ.

Використання CLI

Розробка функцій спершу ведеться локально з використанням стандартних інструментів тестування.

Таким чином, в корені проекту лежить стандартний package.json з установленими потрібним пакетами для забезпечення пакетів тестування та потрібних залежностей:

```
{
  "name": "Lab-02",
  "version": "1.0.0",
  "description": "**Lab-02 **",
  "main": "./func/f3.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "repository": {
    "type": "",
    "url": ""
  },
  "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 каталозі **test** знаходяться тестові кайси для функцій, що запускаються локально

В каталозі **NbuServices** знаходиться програмний код функцій

В каталозі **params** знаходяться json файли з параметрами функцій, що передаються на вхід.

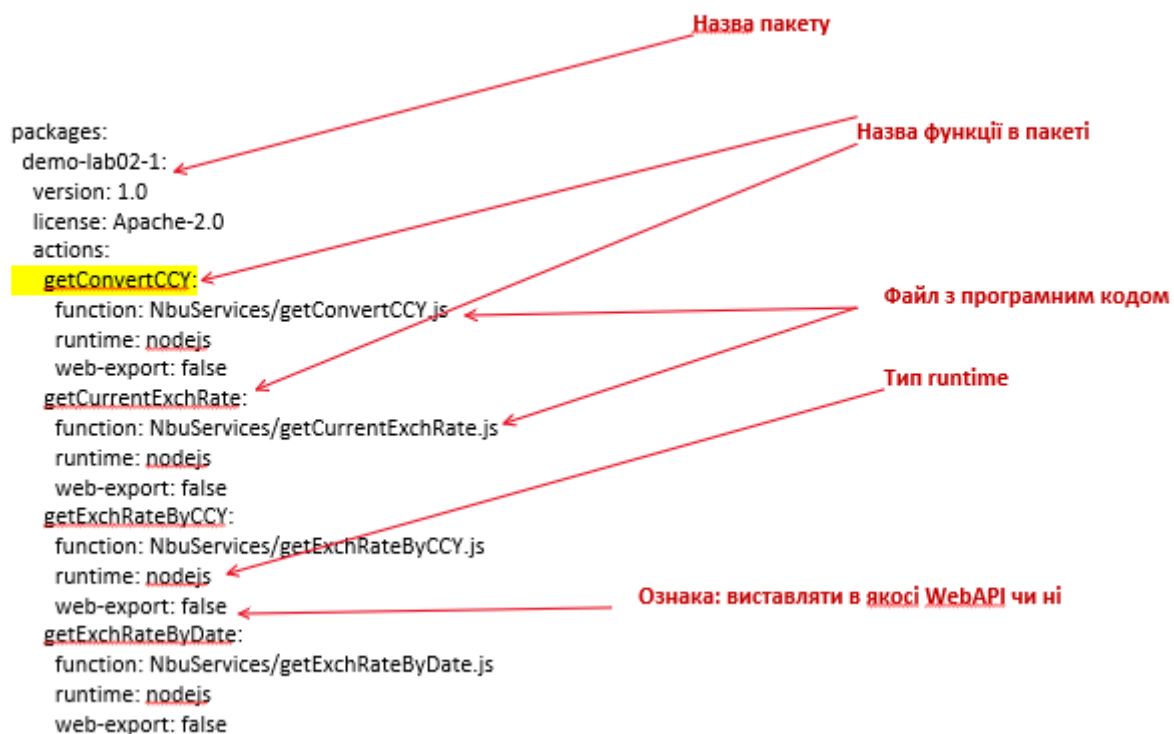
В корені проекту знаходиться **yaml** файл, що описує деплоймент пакету в хмару з допомогою CLI.

```

packages:
  demo-lab02-1:
    version: 1.0
    license: Apache-2.0
    actions:
      getConvertCCY:
        function: NbuServices/getConvertCCY.js
        runtime: nodejs
        web-export: false
      getCurrentExchRate:
        function: NbuServices/getCurrentExchRate.js
        runtime: nodejs
        web-export: false
      getExchRateByCCY:
        function: NbuServices/getExchRateByCCY.js
        runtime: nodejs
        web-export: false
      getExchRateByDate:
        function: NbuServices/getExchRateByDate.js
        runtime: nodejs
        web-export: false

```

Призначення компонентів файлу показані на малюнку



pic-6

Якщо вже виконали команду

```
ibmcloud login
```

to deployment виконується дуже просто:

```
ibmcloud fn deploy --manifest demo-lab02-1.yml
```

Переглянути список пакетів та функцій можна ,відповідно, командами

```
ibmcloud fn package list  
ibmcloud fn action list
```

Запустити на виконання окремо взяту функцію можна командою:

```
ibmcloud fn action invoke ACTION_NAME [--blocking] [--param KEY VALUE] [--param-  
file FILE] [--result]
```

Для прикладу:

Запустить функцію без параметрів і поверне id запуску

```
ibmcloud fn action invoke demo-lab02-1/getCurrentExchRate
```

По отриманому id можна прочитати лог виконання функції та отримати результат

```
## лог виконання  
ibmcloud fn activation logs 914d4c45944342468d4c4594439246b9  
  
## результат виконання  
ibmcloud fn activation result db355334a09348c7b55334a09368c70b
```

Результат можна отримати і зразу, запустивши функцію з ключем **--result**

```
ibmcloud fn action invoke demo-lab02-1/getCurrentExchRate --result
```


Типове питання, яке лежить на поверхні: "як у функцію передати параметри"? Для цього є каталог `params` в якому знаходяться `json` файли з параметрами запуску функцій. Їх і викристаємо.

А бо ж можна прямо з командного ряку задати:

```
ibmcloud fn action invoke demo-lab02-1/getCurrentExchRate --param date 2020-06-16 --result
```

Або ж з використанням файлів за параметрами:

```
ibmcloud fn action invoke demo-lab02-1/getCurrentExchRate --param-file ./params/getExchRateByDate.json --result
```

```
ibmcloud fn action invoke demo-lab02-1/getExchRateByCCY --param-file ./params/getExchRateByCCY.json --result
```

```
ibmcloud fn action invoke demo-lab02-1/getConvertCCY --param-file ./params/getConvertCCY.json --result
```