

LAB-1. IBM Cloud Functions. Ознайомлення з IBM Cloud Dashboard

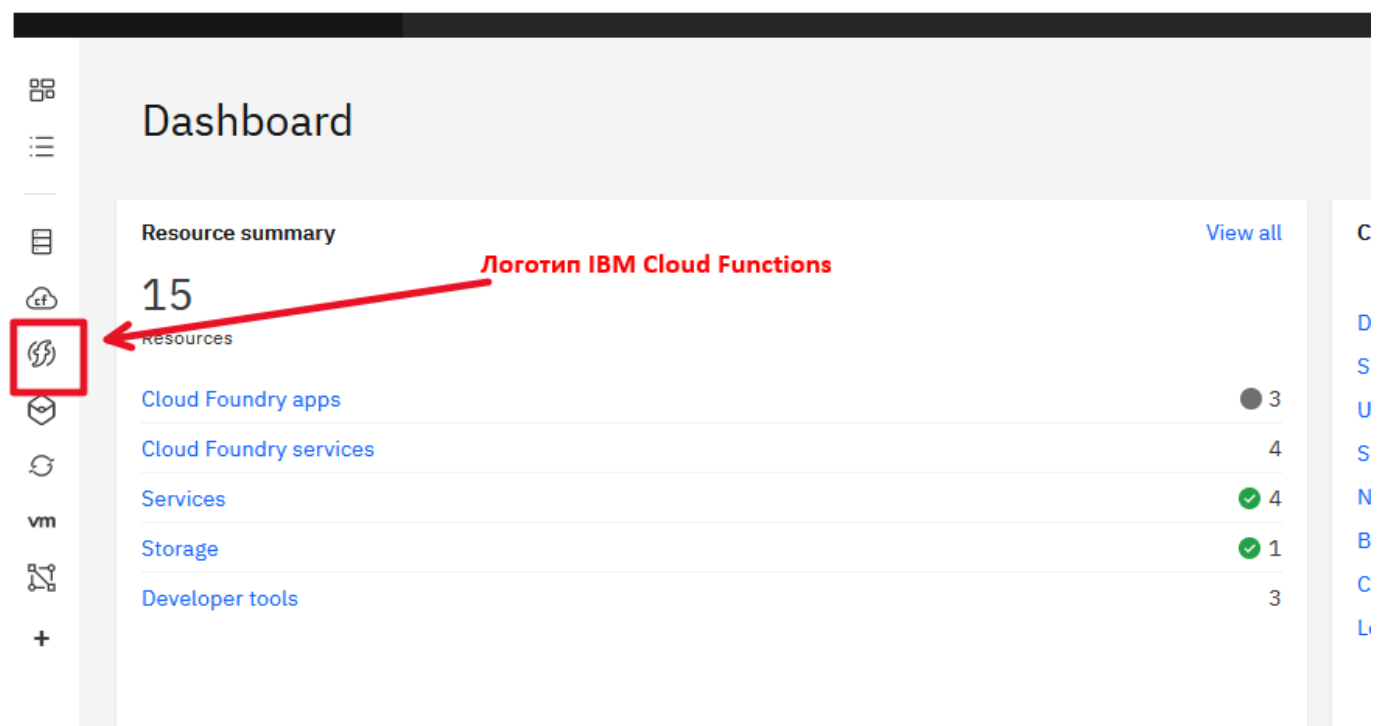
- 1 [Вступ](#)
- 2 [IBM Cloud Functions на IBM Cloud Dashboard](#)
- 3 [Создание простой функции из шаблона](#)
- 4 [Создание собственной функции](#)
- 5 [Выставить функцию, как REST API](#)

Вступ

В лабораторній роботі ознайомимося з IBM Cloud Dashboard для створення і управління IBM Cloud Functions.

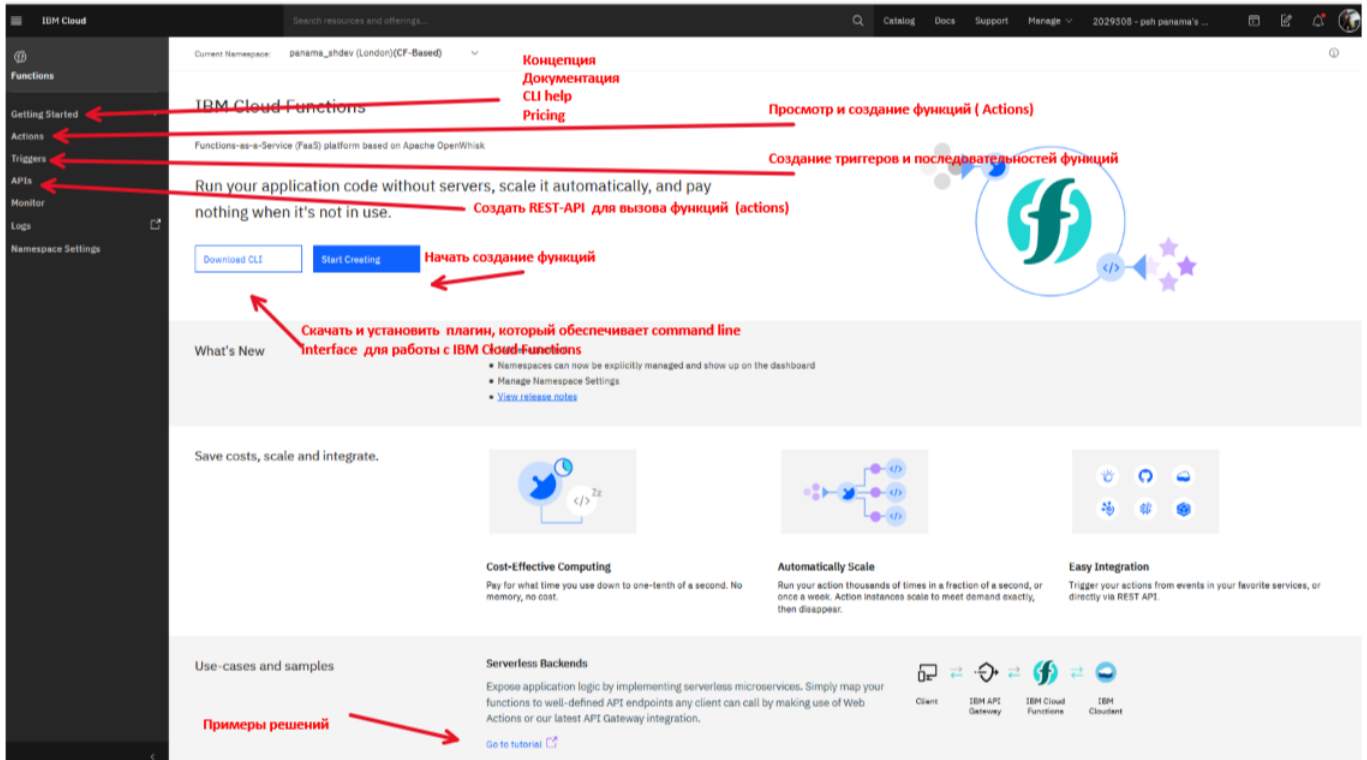
IBM Cloud Functions на IBM Cloud Dashboard

Якщо зайти в IBM Cloud Dashboard, то IBM Cloud Functions стають зразу доступними. На піс-1 показано, де можна найти їх.



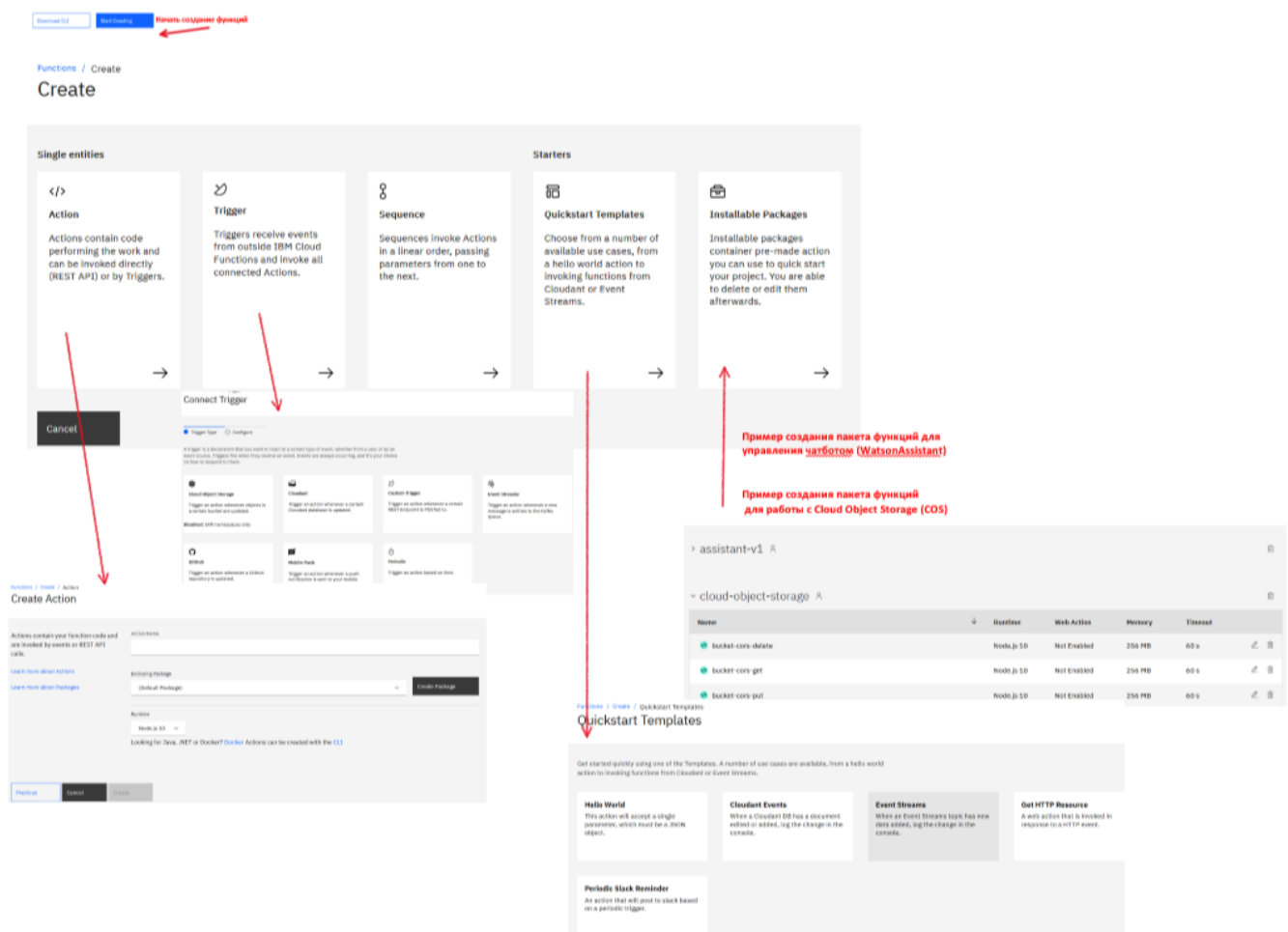
pic-1

В результате натискання на піктограму переходимо в главну консоль IBM Cloud Functions. На піс-2 показані основні елементи управління, що необхідні для створення функцій.



pic-2

Нажимаємо на кнопку "Start Creating" і попадаємо в головний екран для створення функцій (pic-3).

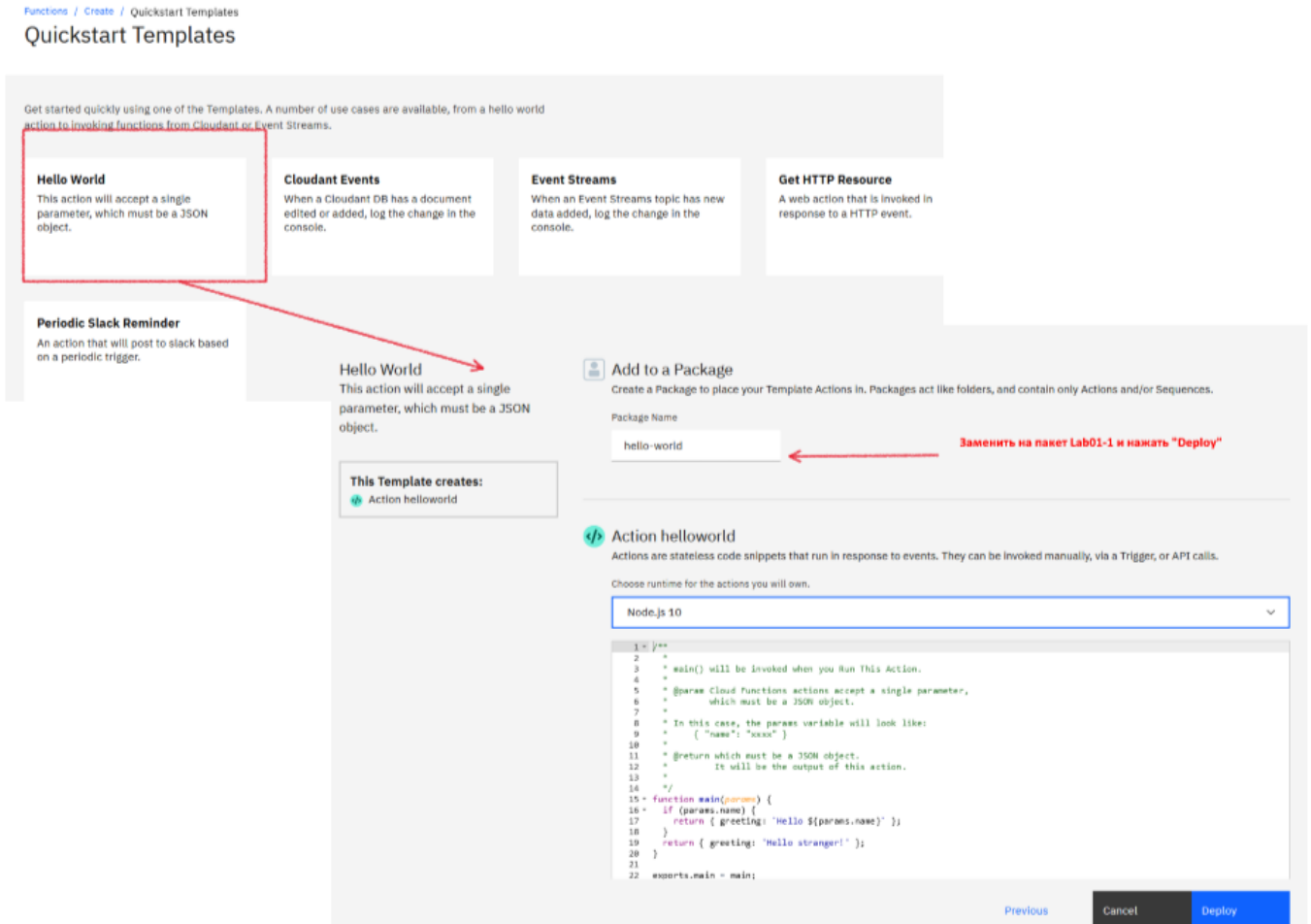


pic-3

На цьому скриншоті показані всі можливості по створенню функцій.

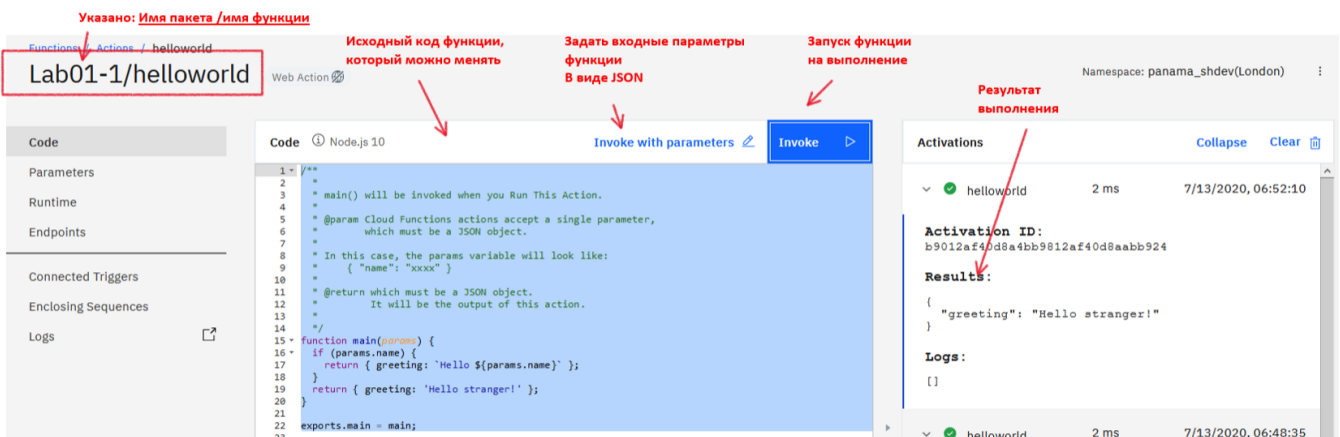
Створення простої функції з уже існуючого шаблону

Для цього поріно знову зайти "Start Creating", вибрати "From Template" и выбираємо "Hello World" (pic-4).



pic-4

Після успішного створення попадаємо у вікно з програмним кодом функції (pic-5).



pic-5

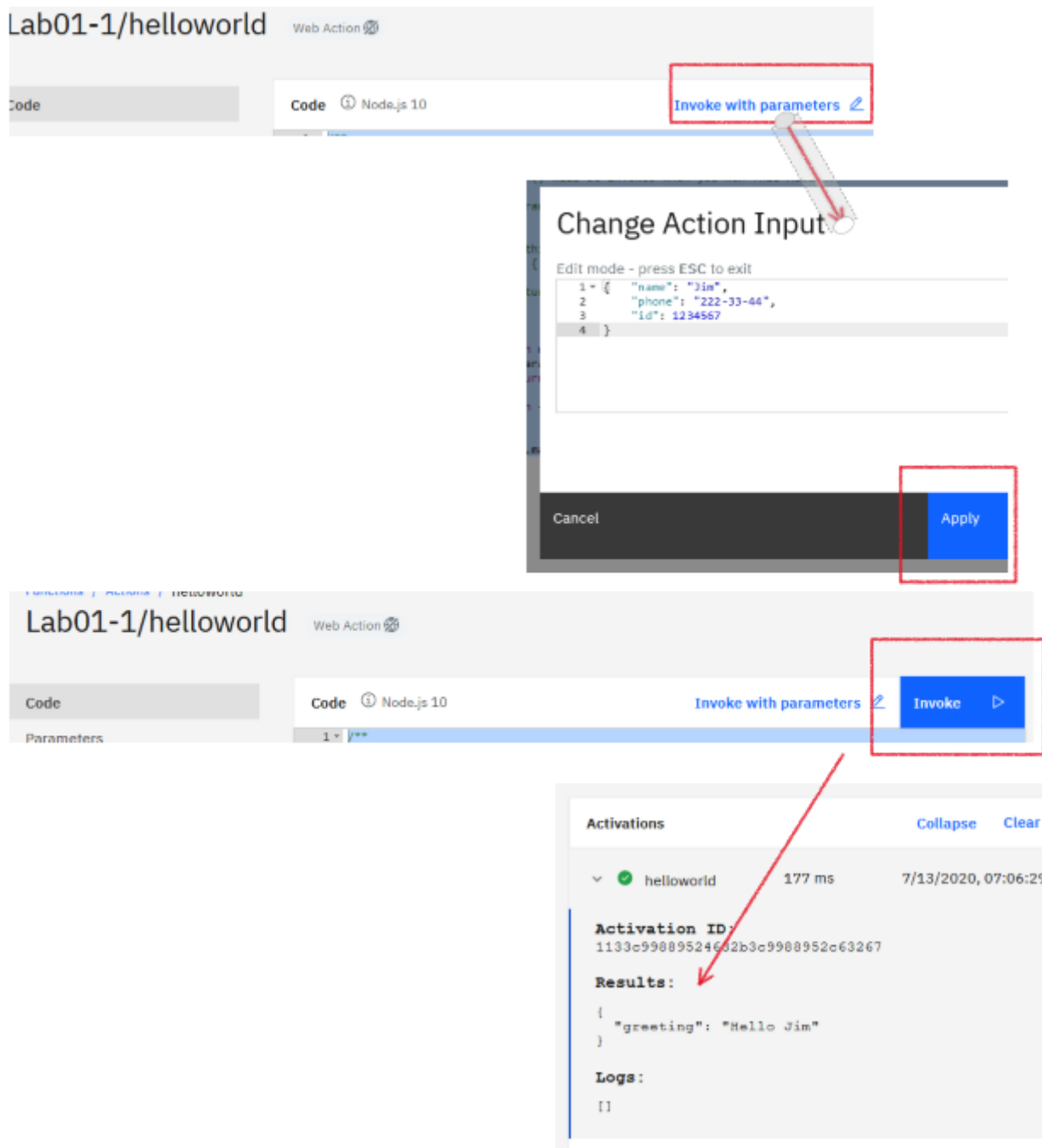
В цьому прикладі, при натисканні на кнопку **"Invoke"**, виконується запуск функції и результаті роботи функції побачиом в правій частині вікна. Параметрів у функцію не передавалось, тому зразу побачиом відповідь:

```
{  
  "greeting": "Hello stranger!"  
}
```

Тепер по кнопці: **"Invoke with parameters"** задаємо параметри запуску функції. Параметри задаються у вигляді плоского JSON файла. Далі показано приклад json параметрів.

```
{  
  "name": "Jim",  
  "phone": "222-33-44",  
  "id": 1234567  
}
```

Дані цього json потрібно вставити у вікно вводу параметрів і повторити запуск функції.



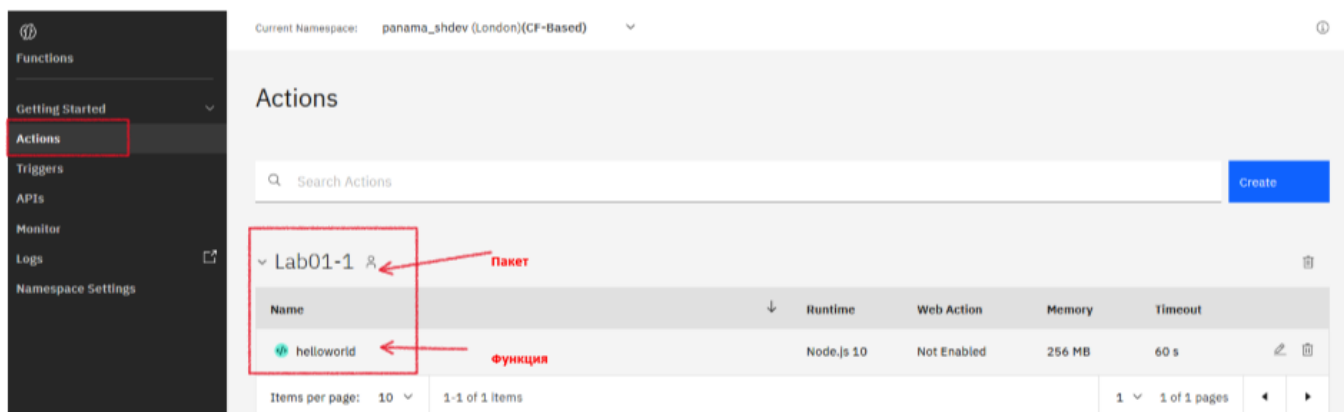
pic-6

В цьому випадку відповідь уже змінилася, і ми отримали у відповіді **"Hello Jim"**, тобто параметр **"name"** був прочитаний

```
{
  "greeting": "Hello Jim"
}
```

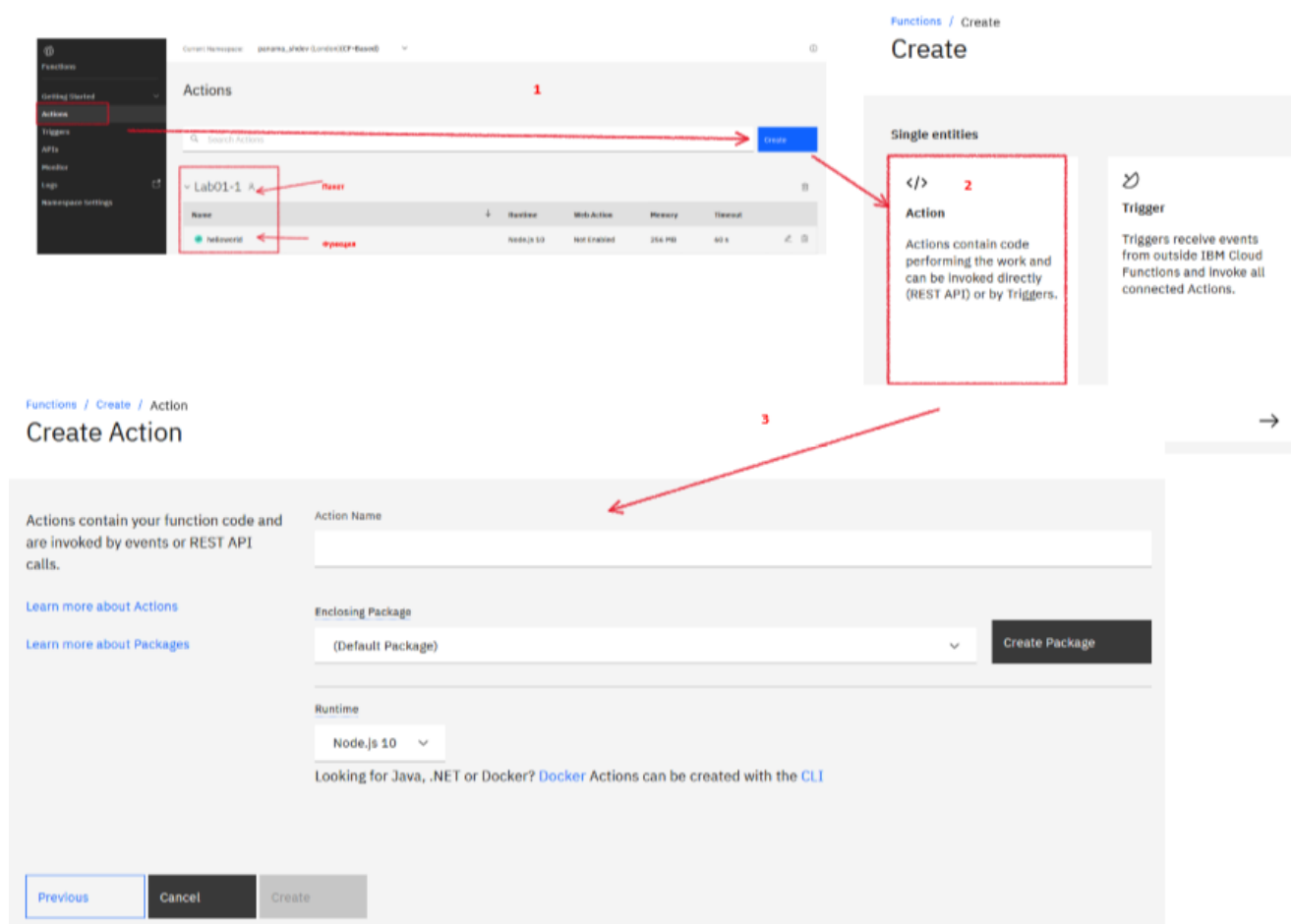
Створення власної функції

Перейдемо до списку **"Actions"**. Ми бачимо тільки одну функцію helloworld в пакеті Lab01-1 (pic-7).



pic-7

Для створення власної функції потрібно натиснути кнопку **"Create"** и вибрати опцію **"Action"** (pic-8).



pic-8

Далі потрібно:

- внести ім'я Action, для прикладу: **"GetCustomerByPhone"**;
- вибрати ім'я існуючого пакету, або створити новий;
- натиснути **"Create"**.

На (pic-9) показане заповнене вікно вводу.

pic-9

Натискаємо кнопку **"Create"**, отримаємо шаблон для написання свого програмного коду (pic-10).

pic-10

Для написання своєї функції (action), придумуємо абстрактну задачу: Розробити асинхронну функцію, яка по номеру телефона поверне данні клієнта.

На вхід функція приймає json об'єкт:

```
{"phone": "222-33-44"}
```

У випадку успішного виконання, функція поверне json-об'єкт:

```
{
  "cif": "100012340",
  "cif_address": "Kyiv, Khreschatic, 26, flat 123",
  "cif_firstname": "Pavlo",
  "cif_lastname": "Petrenko"
}
```

В випадку, коли телефон не переданий, функція генерує помилку з повідомленням: *"Не передан параметр params.phone!"*

Програмний код функції знаходиться у файлі: [/LAB-1/GetCustomerByPhone.js](#) Програмний код необхідно вставити у вікно браузера. В **"Invoke with parameters"** необхідно вставити json с параметрами запуску

```
{ "phone": "222-33-44" }
```

та запустити на виконання по кнопці **"Invoke"**. В результаті побачимо екран як на ріс-11.

The screenshot shows the Google Cloud Functions console for the function 'Lab01-1/GetCustomerByPhone'. The 'Code' tab is selected, showing the following JavaScript code:

```

5
6 * @param {object} params Параметр, который принимает action в виде json-объекта
7 * @param {string} params.phone - Номер телефона
8 *
9 * @return {object} result - JSON объект с параметрами клиента .
10 * @return {string} result.cif - Внутренний код клиента
11 * @return {string} result.cif_firstname - Имя клиента
12 * @return {string} result.cif_lastname - Фамилия клиента
13 * @return {string} result.cif_address - Адресс клиента
14 */
15
16 function main(params) {
17   //return { message: 'Hello World' };
18   return new Promise ( (resolve, reject) => {
19     setTimeout(() => {
20       if ( !isDefined(params.phone) ){
21         var err = new Error("Не передан параметр params.phone!");
22         reject( err );
23       }
24       else {
25         var result = {
26           cif: '100012340',
27           cif_firstname: 'Pavlo',
28           cif_lastname: 'Petrenko',
29           cif_address: 'Kyiv, Khreschatic, 26, flat 123'
30         };
31         resolve(result);
32       }
33     }, 1000);
34   });
35 }

```

The 'Invoke with parameters' button is highlighted. The 'Activations' tab shows two successful invocations:

- GetCustomerByPhone 1296 ms 7/13/2020, 08:41:32
- GetCustomerByPhone 1005 ms 7/13/2020, 08:30:12

pic-11

Функція працює. Тепер необхідно функцію виставити в якості REST API для публічного використання.

Виставити функцію, как REST API

Для цього необхідно

- в розділі Endpoints включити Web Action (pic-12).

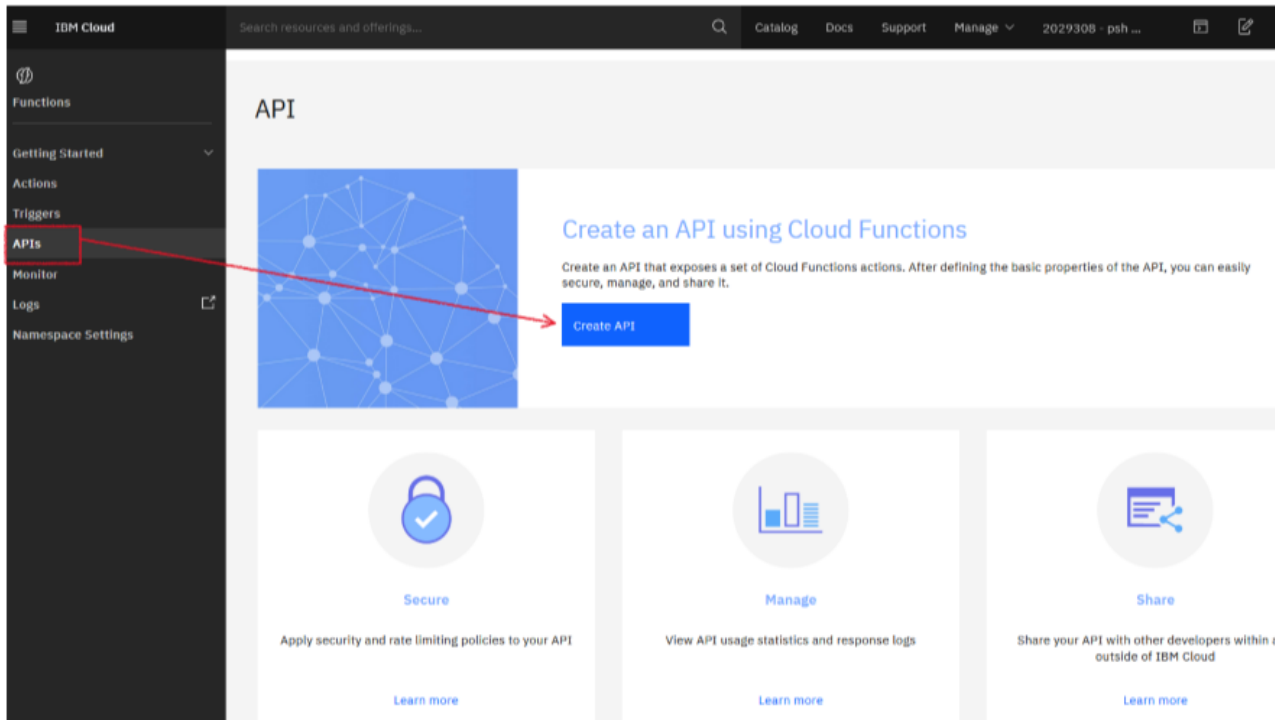
The screenshot shows the Google Cloud Functions console for the function 'Lab01-1/GetCustomerByPhone'. The 'Endpoints' tab is selected, and the 'Enable as Web Action' checkbox is checked. The 'Save' button is highlighted.

The 'Endpoints' tab shows the following configuration:

HTTP Method	Auth	URL
ANY	Public	https://eu-gb.functions.cloud.ibm.com/api/v1/web/panama_shdev/Lab01-1/GetCustomerByPhone

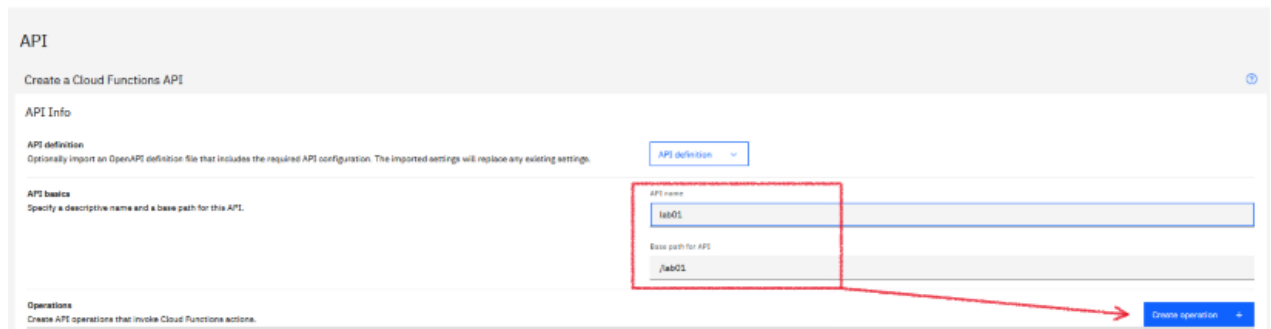
pic-12

- вернутися в розділ основного меню functions и выбрать розділ API (pic-13).



pic-13

- вводимо найменування API и BaseURL.



pic-14

- Встановлюємо http суфікс і http метод.

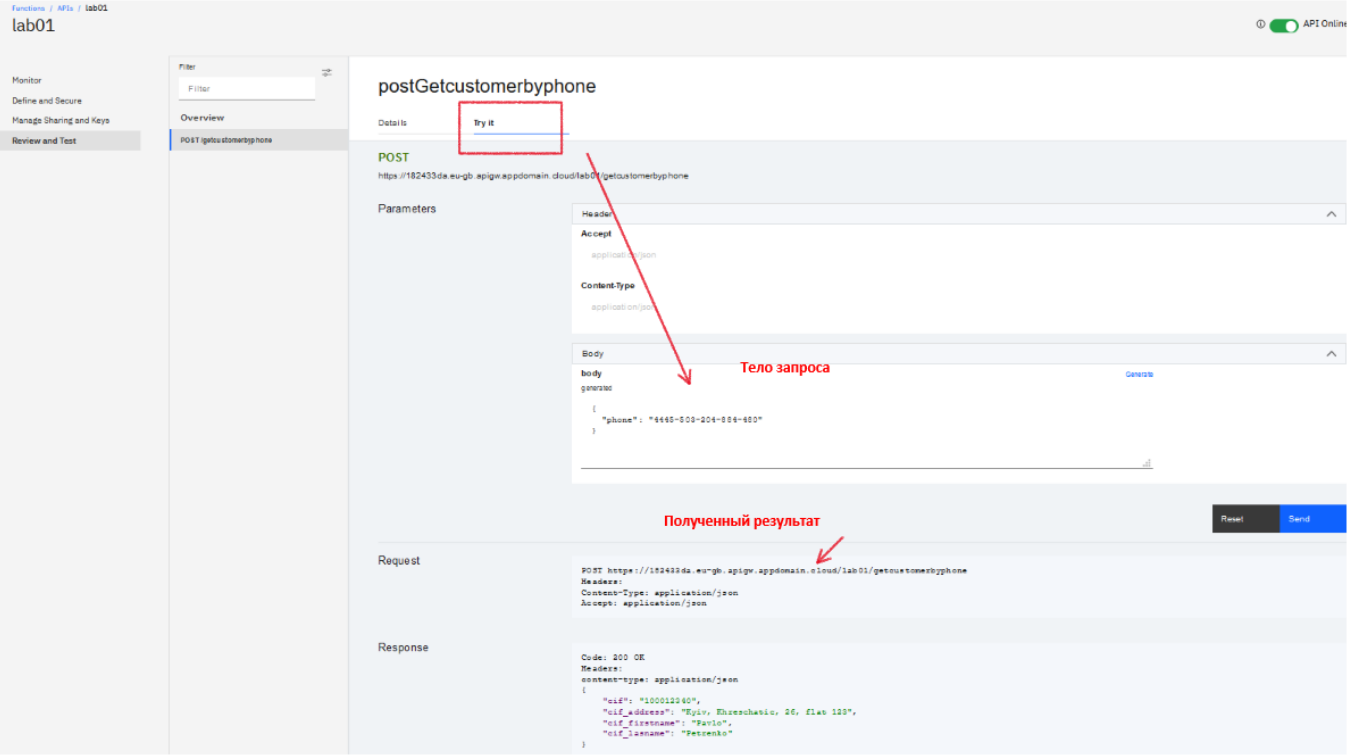
pic-15

- Авторизацію ми не передбачаємо зараз, тому опускаємося вниз і нажимаємо **"Create"**

В результаті виконання кроків ми отримуємо екран моніторингу та управління API (pic-16).

pic-16

Зходимо в меню **review and test**, вибираємо вкладку **try it**, вводимо параметри запиту та запускаємо на виконання. Результат показано на pic-17.



pic-17