

IBM Cloud Functions Розробка функцій зі сторонніми бібліотеками

Дуже часто потрібно додати свої прт - модулі для розробки функцій. Але їх немає в стандартній поставці. Що робити тоді. Є 2 варіанти:

- пакування zip-файлу і його загрузка;
- створення свого контейнеру і уже жеплоїти функцію у свій контейнер.

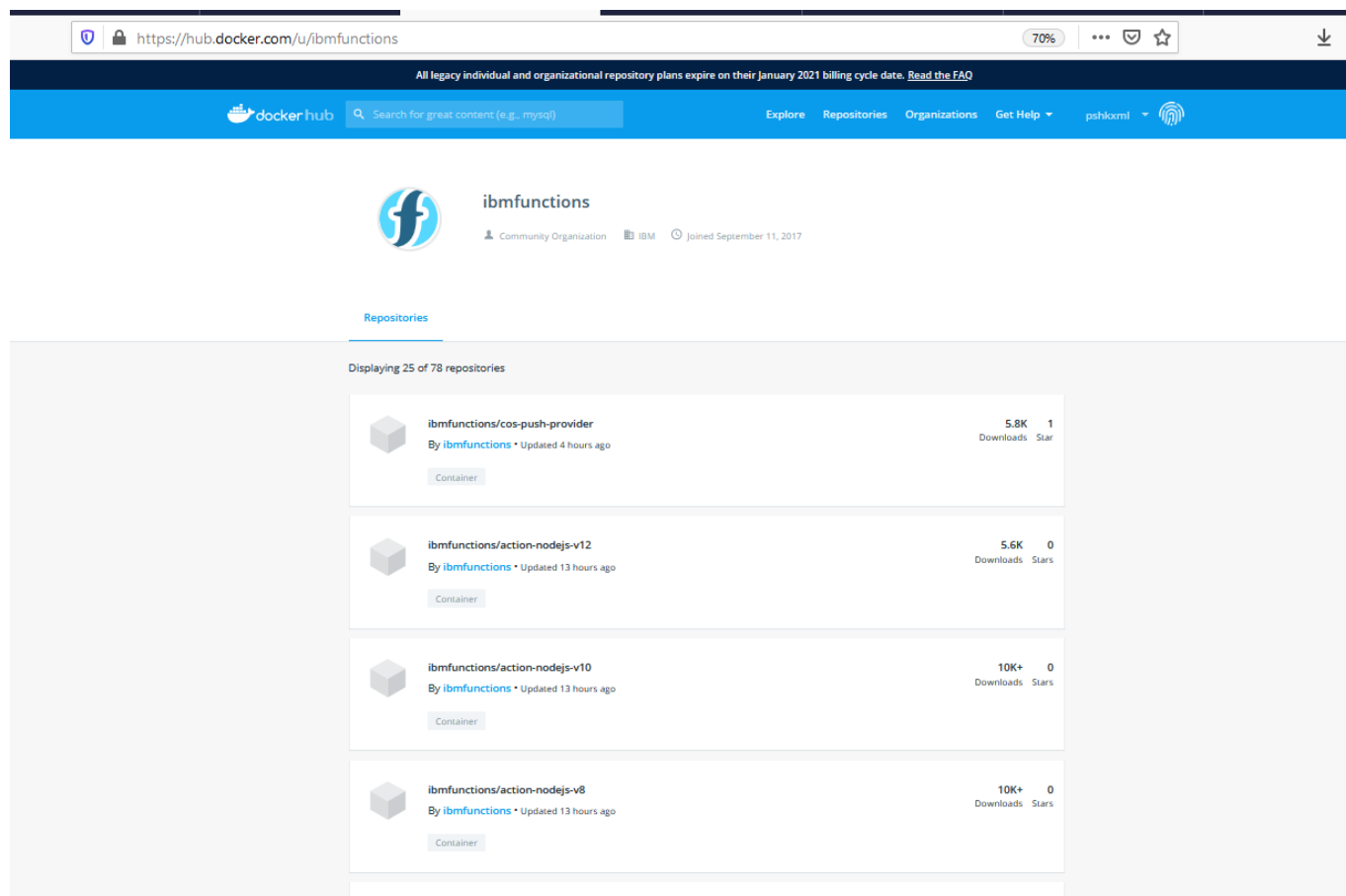
В цьому випадку є одне обмеження - контейнер повинен лежати в публічному docker репозиторії. Використання zip файлу не так очевидно, тому що існують обмеження на його розмір.

Лінки на документацію:

[Перелік стандартних пакетів Node.js](#)

[Підготовка вашого app в Docker image](#)

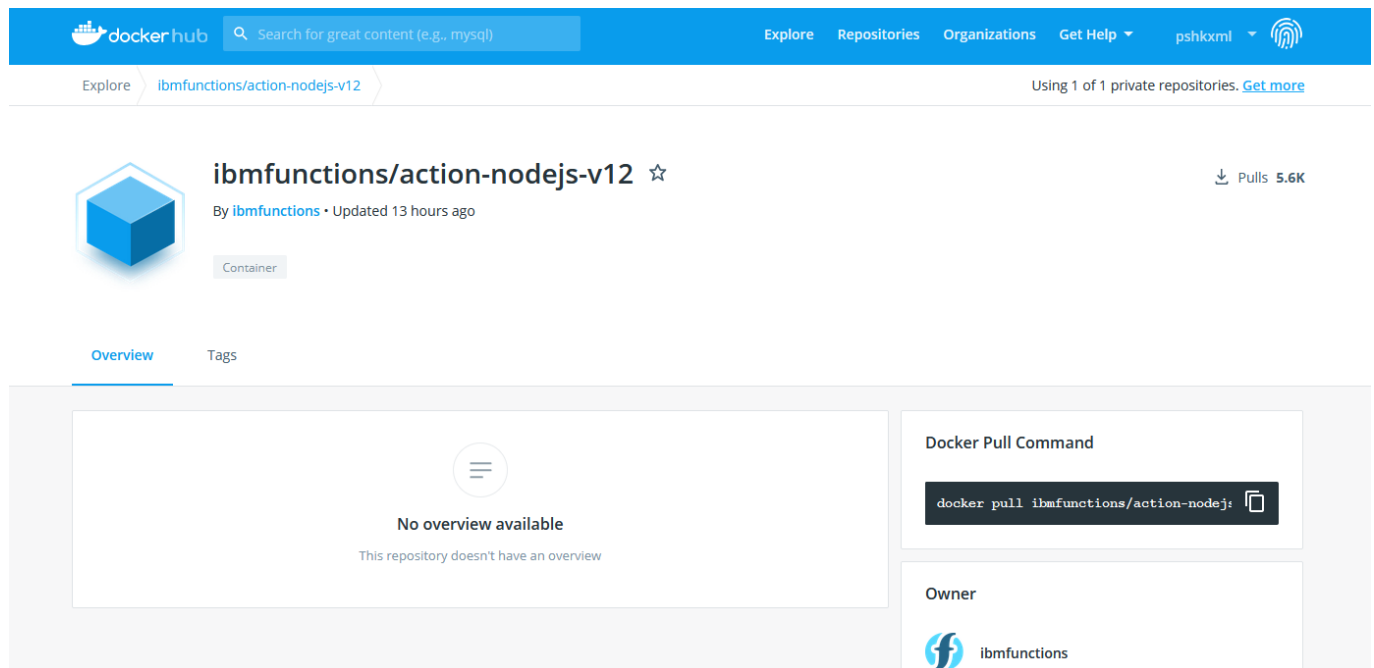
[DockerHub образи для різних runtimes-мов програмування](#)



pic-1

Методика розробки.

Розгортаємо таку ж саму стандартну структуру каталогів, що і для Lab-3, Lab-4. Додається каталога ./Build в якому знаходиться dockerfile та cmd файл для побудови свого образу та розміщення його в свій публічний docker репозиторій

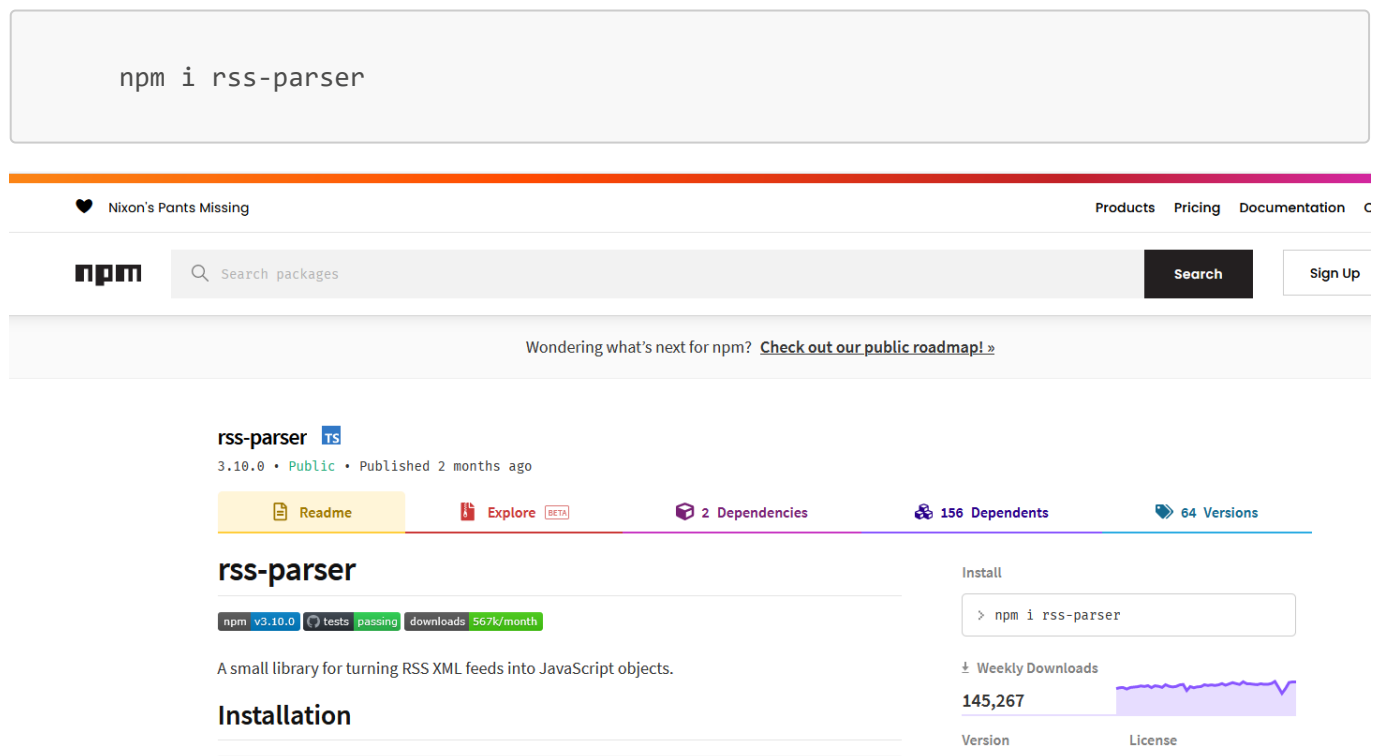


pic-2

Постанова задачі

Потрібно розробити функцію, що по буде скачувати RSS - потік новин, трансформувати її з XML в JSON. Та, записати отримани JSON на Cloud Object Storage.

Для трансформації XML RSS в JSON використаємо npm пакет [rss-parser](#)



pic-3

Розробка та тестування функцій локально

Для цього виконуємо традиційний

```
npm i rss-parser
```

та пишемо функцію і тестові кейси

- RSS-ExtrObject.js функція
- test-RSS-ExtrObject.js тестовий кейс
- Параметри функції - RSS-ExtrObject.json - RSS-ExtrObject_1.json - RSS-ExtrObject_2.json

```
{
  "pub_endpoint": "s3.eu-de.cloud-object-storage.appdomain.cloud",
  "bucket": "cloud-object-storage-bi-cos-static-web-hosting-c2i",
  "rss_url": "https://censor.net.ua/includes/resonance_en.xml",
  "rss_name": "censor_net.json"
}

{
  "pub_endpoint": "s3.eu-de.cloud-object-storage.appdomain.cloud",
  "bucket": "cloud-object-storage-bi-cos-static-web-hosting-c2i",
  "rss_url": "http://feeds.bbc.co.uk/news/education/rss.xml",
  "rss_name": "bbc_education.json"
}

{
  "pub_endpoint": "s3.eu-de.cloud-object-storage.appdomain.cloud",
  "bucket": "cloud-object-storage-bi-cos-static-web-hosting-c2i",
  "rss_url": "http://feeds.bbc.co.uk/news/rss.xml",
  "rss_name": "bbc_news.json"
}

{
  "pub_endpoint": "s3.eu-de.cloud-object-storage.appdomain.cloud",
  "bucket": "cloud-object-storage-bi-cos-static-web-hosting-c2i",
  "rss_url": "http://feeds.bbc.co.uk/news/rss.xml",
  "rss_name": "bbc_news.json"
}
```

Коли функція розроблена та протестована настав час збирання образу

Створення образу Docker

- Написати простий Docker файл При побудові образу ми додаємо потрібний пакет

./build/Docker

```
FROM ibmfunctions/action-nodejs-v12
RUN npm install rss-parser
```

- Побудувати образ, помітити його tag та помістити в публічний Docker - репозиторій

./build/build.cmd

```
##### Побудувати отобраз локально
docker build -t pshkxml/icfrss .

##### Поставити tag для розміщення в публічному репозиторії
docker tag pshkxml/icfrss:latest pshkxml/icfrss:1.0.1

##### push to docker Відправити в публічний репозиторій
docker push pshkxml/icfrss:1.0.1
```

The screenshot shows the Docker Hub interface for the repository `pshkxml/icfrss`. The repository is public and has no description. The 'Tags and Scans' section shows a table with one tag, `1.0.1`, which was pushed 7 hours ago. The 'Recent builds' section is empty. The 'Docker commands' section shows the command `docker push pshkxml/icfrss:tagname`.

TAG	OS	PULLED	PUSHED
1.0.1		in a few seconds	7 hours ago

pic-4







Deployemnt функції

В цьому випадку через yml файл деплоймент на проходить Потрібно кожну функцію деплоїти окремо. При цьому в команді action create або action update вказуємо посилання на репозиторій

**** --docker pshkxml/icfrss:1.0.1 ****

```
ibmcloud fn package create rss-srvc-demo
ibmcloud fn action update rss-srvc-demo/RSS-ExtrObject --docker
pshkxml1/icfrss:1.0.1 src/RSS-ExtrObject.js
```

▼ rss-srvc-demo ⓘ

Name	↓ Runtime	Web Action	Memory	Timeout	
 RSS-ExtrObject	Docker	Not Enabled	256 MB	60 s	 
 RSS-ExtrObject-STT	Node.js 12	Not Enabled	256 MB	60 s	 

Items per page: 10 ▼ 1-2 of 2 items

1 ▼ 1 of 1 pages ◀ ▶

pic-5

Functions / Actions / RSS-ExtrObject

rss-srvc-demo/RSS-ExtrObject Web Action ⓘ Namespace: pasha.kxml@gmail.com_dev2(Dallas)

Code

Parameters

Runtime

Endpoints

Code ⓘ Docker

Invoke with parameters ⓘ

Invoke ▶

Currently Docker, Java (.jar), and Compressed (.zip) Actions are unable to be edited from the IBM Cloud Functions UI These Actions can still be Invoked

pic-6