

# Architecture distribuée et virtualisation

Ahmed Alotaibi

## TD 1 : Une application Node.js

- 1) J'ai réalisé les différentes commandes décrites dans le readme et voici les différents résultats obtenus :

```
● @Ahmed11uu →/workspaces/i-want-typescript (main) $ npm run lint && npm run format

> i-want-typescript@1.0.0 lint
> eslint "src/**/*.ts" --fix

> i-want-typescript@1.0.0 format
> prettier --write .

.eslintrc.json 42ms
.jestrc.json 4ms
.nodemon.json 3ms
.prettierrc.json 3ms
package-lock.json 304ms
package.json 15ms
README.md 36ms
src/hello-word.spec.ts 153ms
src/hello-world.ts 9ms
src/index.ts 6ms
tsconfig.json 6ms
○ @Ahmed11uu →/workspaces/i-want-typescript (main) $
```

```
● @Ahmed11uu →/workspaces/i-want-typescript (main) $ npm run test

> i-want-typescript@1.0.0 test
> jest --config .jestrc.json

PASS src/hello-word.spec.ts
  typeScript test suite
    ✓ should return "Hello world!" (3 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        1.951 s, estimated 2 s
Ran all test suites.
```

```

● @Ahmed11uu →/workspaces/i-want-typescript (main) $ npm run test:coverage

> i-want-typescript@1.0.0 test:coverage
> jest --config .jestrc.json --coverage

PASS src/hello-world.spec.ts
  typeScript test suite
    ✓ should return "Hello world!" (2 ms)

-----|-----|-----|-----|-----|-----
File      | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
-----|-----|-----|-----|-----|-----
All files |    100   |    100   |    100   |    100   |
hello-world.ts |    100   |    100   |    100   |    100   |
-----|-----|-----|-----|-----|
Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        2.068 s
Ran all test suites.

```

```

● @Ahmed11uu →/workspaces/i-want-typescript (main) $ npm run build

> i-want-typescript@1.0.0 build
> tsc

```

```

● @Ahmed11uu →/workspaces/i-want-typescript (main) $ npm run start

> i-want-typescript@1.0.0 start
> node dist/index.js

Hello, World!

```

```

○ @Ahmed11uu →/workspaces/i-want-typescript (main) $ npm run watch

> i-want-typescript@1.0.0 watch
> nodemon --config .nodemon.json src/index.ts

[nodemon] 2.0.20
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): src/**/*.ts
[nodemon] watching extensions: ts,json
[nodemon] starting `ts-node src/index.ts`
Hello, World!
[nodemon] clean exit - waiting for changes before restart

```

- 2) Le fichier package.json est un fichier important dans la gestion des projets node.js. Il regroupe l'ensemble des informations importantes sur le projet comme le nom, la version, la description, le typescript, les scripts ou bien encore les dépendances. Ce fichier est important pour la gestion, le développement et le partage des projets. Il est géré manuellement.

Le fichier package-lock.json est fichier qui se trouve créé automatiquement et mis à jour par npm lors de la création d'un projet. Il permet d'obtenir des installations identiques à chaque utilisation en gardant les mêmes versions de dépendances.

3) J'ai installé la bibliothèque systeminformation :

```
@Ahmed11uu → /workspaces/i-want-typescript (main) $ npm install system information
added 2 packages, and audited 414 packages in 3s

58 packages are looking for funding
  run `npm fund` for details

6 vulnerabilities (4 moderate, 1 high, 1 critical)

To address issues that do not require attention, run:
  npm audit fix

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
```

Dans le fichier package.json,

- La partie devDependencies regroupe l'ensemble des outils nécessaires pour les tests et la gestion d'un projet. Ces outils ne sont pas nécessaires pour produire le projet
- Par contre, la partie dependencies regroupe l'ensemble des informations nécessaires pour produire le projet.

4) Pour obtenir l'application demandée, j'ai créé un fichier server.js sous Visual Studio Code et écrit le programme suivant :

```
const express = require('express');
const si = require('systeminformation');

const app = express();
const port = 5000;

app.get('/api/v1/systemInformation', async (req, res) => {
  try {
    const cpu = await si.cpu();
    const system = await si.system();
    const mem = await si.mem();
    const os = await si.osInfo();
    const currentLoad = await si.currentLoad();
    const processes = await si.processes();
    const diskLayout = await si.diskLayout();
    const networkInterfaces = await si.networkInterfaces();
```

```

const systemInformation = {
  cpu,
  system,
  mem,
  os,
  currentLoad,
  processes,
  diskLayout,
  networkInterfaces
};

res.json(systemInformation);
} catch (error) {
  console.error('Error fetching system information:', error);
  res.status(500).json({ error: 'Server Error' });
}
});

app.use((req, res) => {
  res.status(404).json();
});

app.listen(port, () => {
  console.log(`Server listening on the port: `, port);
});

```

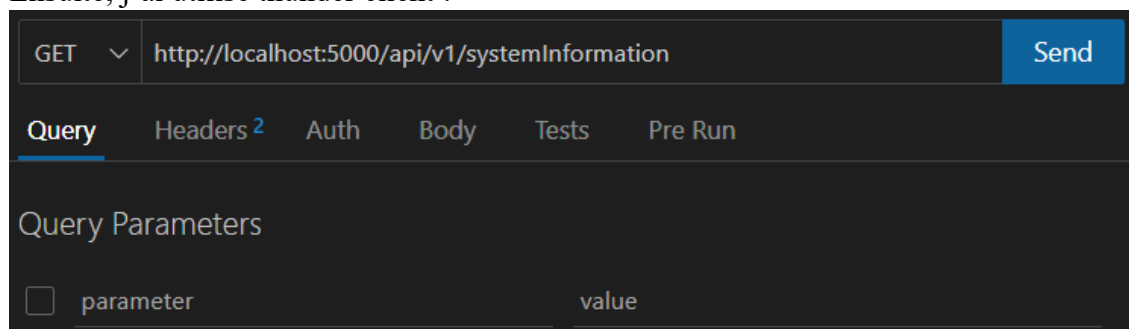
Ensuite, j'ai mis le server en route :

```

PS C:\Users\Administrateur\Desktop\tp-programmer\tp-programmer> node server.js
Server listening on the port: 5000

```

Ensuite, j'ai utilisé thunder client :



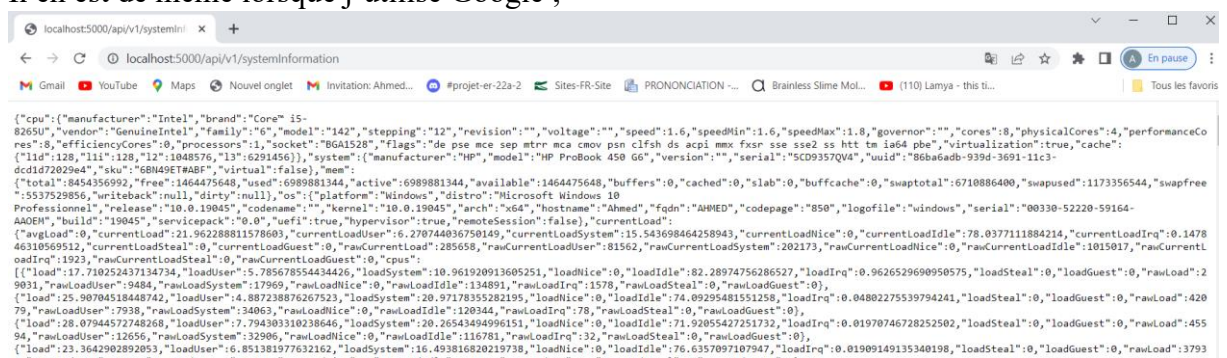
Et, ci-dessous une partie de la réponse obtenue :

```
Status: 200 OK   Size: 150.06 KB   Time: 26.58 s

Response  Headers 6  Cookies  Results  Docs  {}  ≡

27 },
30 "system": {
31   "manufacturer": "HP",
32   "model": "HP ProBook 450 G6",
33   "version": "",
34   "serial": "5CD9357QV4",
35   "uuid": "86ba6adb-939d-3691-11c3-dcd1d72029e4",
36   "sku": "6BN49ET#ABF",
37   "virtual": false
38 },
39 "mem": {
40   "total": 8454356992,
41   "free": 1447186432,
42   "used": 7007170560,
43   "active": 7007170560,
44   "available": 1447186432,
45   "buffers": 0,
46   "cached": 0,
47   "slab": 0,
48   "buffcache": 0,
49   "swaptotal": 6710886400,
50   "swapused": 1284505600,
51   "swapfree": 5426380800,
52   "writeback": null,
53   "dirty": null
54 },
55 "os": {
56   "platform": "Windows",
57   "distro": "Microsoft Windows 10 Professionnel",
58   "release": "10.0.19045",
59   "codename": "",
60   "kernel": "10.0.19045",
61   "arch": "x64",
62   "hostname": "Ahmed",
63   "fqdn": "AHMED",
```

Il en est de même lorsque j'utilise Google ;



```
{
  "cpu": {
    "manufacturer": "Intel",
    "brand": "Core",
    "i5-8265U",
    "vendor": "GenuineIntel",
    "family": "6",
    "model": "142",
    "stepping": "12",
    "revision": "",
    "voltage": "",
    "speed": 1.6,
    "speedMin": 1.6,
    "speedMax": 1.8,
    "governor": "",
    "cores": 8,
    "physicalCores": 4,
    "performanceCores": 8,
    "efficiencyCores": 0,
    "processors": 1,
    "socket": "BG41528",
    "flags": "de pse mce sep mtrr mca cmov psn clfsd ds acpi mmx fxsr sse2 ss htt tm ia64 pbe",
    "virtualization": true,
    "cache": {
      "l1d": 128,
      "l1i": 128,
      "l2": 1048576,
      "l3": 6291456
    },
    "system": {
      "manufacturer": "HP",
      "model": "HP ProBook 450 G6",
      "version": "",
      "serial": "5CD9357QV4",
      "uuid": "86ba6adb-939d-3691-11c3-dcd1d72029e4",
      "sku": "6BN49ET#ABF",
      "virtual": false
    },
    "mem": {
      "total": 8454356992,
      "free": 1464475648,
      "used": 6989881344,
      "active": 6989881344,
      "available": 1464475648,
      "buffers": 0,
      "cached": 0,
      "slab": 0,
      "buffcache": 0,
      "swaptotal": 6710886400,
      "swapused": 1173356544,
      "swapfree": 5537529856,
      "writeback": null,
      "dirty": null
    },
    "os": {
      "platform": "Windows",
      "distro": "Microsoft Windows 10 Professionnel",
      "release": "10.0.19045",
      "codename": "",
      "kernel": "10.0.19045",
      "arch": "x64",
      "hostname": "Ahmed",
      "fqdn": "AHMED",
      "codepage": "850",
      "logfile": "windows",
      "serial": "00330-52220-59164-AAOEH",
      "build": "19045",
      "servicepack": "0.0",
      "uefi": true,
      "hypervisor": true,
      "remoteSession": false
    },
    "load": {
      "avgLoad": 0,
      "currentLoad": 21.962288811578603,
      "currentLoadUser": 6.270744036750149,
      "currentLoadSystem": 15.543698464258943,
      "currentLoadNice": 0,
      "currentLoadIdle": 78.0377111884214,
      "currentLoadIrq": 0.147846310565912,
      "currentLoadGuest": 0,
      "rawCurrentLoad": 285658,
      "rawCurrentLoadSystem": 81562,
      "rawCurrentLoadNice": 0,
      "rawCurrentLoadIdle": 1015017,
      "rawCurrentLoadIrq": 1923,
      "rawCurrentLoadGuest": 0,
      "rawCurrentLoadSystem": 0,
      "cpu": {
        "load": 17.710252437134734,
        "loadUser": 5.785678554434426,
        "loadSystem": 10.961920913605251,
        "loadNice": 0,
        "loadIdle": 82.28974756286527,
        "loadIrq": 0.9626529690995875,
        "loadSteal": 0,
        "loadGuest": 0,
        "rawLoad": 29031,
        "rawLoadUser": 9484,
        "rawLoadSystem": 17969,
        "rawLoadNice": 0,
        "rawLoadIdle": 134891,
        "rawLoadIrq": 1578,
        "rawLoadSteal": 0,
        "rawLoadGuest": 0
      },
      "load": 25.90704518448742,
      "loadUser": 4.887238876267523,
      "loadSystem": 20.97178355282195,
      "loadNice": 0,
      "loadIdle": 74.09295481551258,
      "loadIrq": 0.04802275539794241,
      "loadSteal": 0,
      "loadGuest": 0,
      "rawLoad": 42079,
      "rawLoadUser": 7938,
      "rawLoadSystem": 34063,
      "rawLoadNice": 0,
      "rawLoadIdle": 120344,
      "rawLoadIrq": 78,
      "rawLoadSteal": 0,
      "rawLoadGuest": 0
    },
      "load": 26.07944572382668,
      "loadUser": 7.794303310238646,
      "loadSystem": 20.26543494996151,
      "loadNice": 0,
      "loadIdle": 71.92055427251732,
      "loadIrq": 0.01970746728252502,
      "loadSteal": 0,
      "loadGuest": 0,
      "rawLoad": 45594,
      "rawLoadUser": 12656,
      "rawLoadSystem": 32906,
      "rawLoadNice": 0,
      "rawLoadIdle": 116781,
      "rawLoadIrq": 32,
      "rawLoadSteal": 0,
      "rawLoadGuest": 0
    },
      "load": 23.3642902892053,
      "loadUser": 6.851381977632162,
      "loadSystem": 16.493816820219738,
      "loadNice": 0,
      "loadIdle": 76.6357097107947,
      "loadIrq": 0.01909149135340198,
      "loadSteal": 0,
      "loadGuest": 0,
      "rawLoad": 3793
```

- 5) J'ai utilisé l'outil curl sous sur l'invite de commande de Windows :

```
C:\Users\Administrateur\Desktop\tp-programmer\tp-programmer>curl http://localhost:5000/api/v1/systemInformation
```

Et, j'obtiens :

```
{
  "cpu": {
    "manufacturer": "Intel",
    "brand": "Core",
    "family": "6",
    "model": "142",
    "stepping": "12",
    "revision": "",
    "voltage": "",
    "speed": 1.6,
    "speedMin": 1.6,
    "speedMax": 1.8,
    "governor": ""
  },
  "cores": {
    "physicalCores": 8,
    "performanceCores": 8,
    "efficiencyCores": 0,
    "processors": 1,
    "socket": "BGAI528",
    "flags": "de pse mce sep mtrr mca cmov psn clflush ds acpi max fxsr sse sse2 ss htt tm la64 pbe",
    "virtualization": true,
    "cache": {
      "l1d": 128,
      "l1i": 128,
      "l2": 1048576,
      "l3": 6391456
    },
    "system": {
      "manufacturer": "HP",
      "model": "HP ProBook 450 G6",
      "version": "",
      "serial": "5C09357QW4",
      "uid": "868a5adb-939d-3691-11c3-dcdd72809a4",
      "sku": "GBN49ET#A8F",
      "virtual": false,
      "mem": {
        "total": 8454356992,
        "free": 1243017216,
        "used": 7211339776,
        "active": 7211339776,
        "available": 1243017216,
        "buffers": 0,
        "cached": 0,
        "slab": 0,
        "buffcache": 0,
        "swaptotal": 6719886480,
        "swapped": 1179648000,
        "writeback": null,
        "dirty": null
      },
      "os": {
        "platform": "Windows",
        "distro": "Microsoft Windows 10 Professional",
        "release": "10.0.19045",
        "codename": "",
        "kernel": "10.0.19045",
        "arch": "x64",
        "hostname": "Ahmed",
        "fqdn": "AHMED",
        "codepage": "858",
        "logfile": "Windows",
        "serial": "08038-52220-59164-A00EM",
        "build": "19045",
        "servicepack": "0.0",
        "uefi": true,
        "hypervisor": true,
        "remoteSession": false,
        "currentLoad": {
          "rawLoad": 0,
          "currentLoad": 22.816616521105833,
          "currentLoadGuest": 0,
          "currentLoadSystem": 16.488838581934486,
          "currentLoadNice": 0,
          "currentLoadIdle": 77.18338347889497,
          "currentLoadIrq": 0.18298835719278332,
          "currentLoadSteal": 0,
          "rawCurrentLoad": 22098738,
          "rawCurrentLoadUser": 5951464,
          "rawCurrentLoadSystem": 15970843,
          "rawCurrentLoadNice": 0,
          "rawCurrentLoadIdle": 74754965,
          "rawCurrentLoadIrq": 177231,
          "rawCurrentLoadSteal": 0,
          "rawCurrentLoadGuest": 0,
          "cpus": [
            {
              "load": 18.505959691066852,
              "loadUser": 5.918274134573184,
              "loadSystem": 11.524811968027995,
              "loadNice": 0,
              "loadIdle": 81.49404830893315,
              "loadIrq": 1.062873584656749,
              "loadSteal": 0,
              "loadGuest": 0,
              "rawLoad": 2260468,
              "rawLoadUser": 722996,
              "rawLoadSystem": 1407734,
              "rawLoadNice": 0,
              "rawLoadIdle": 10954343,
              "rawLoadIrq": 129823,
              "rawLoadSteal": 0,
              "rawLoadGuest": 0,
              "load": 26.663155183739232,
              "loadUser": 24.9011849995357695,
              "loadSystem": 21.67283615367616,
              "loadNice": 0,
              "loadIdle": 73.33684489676977,
              "loadIrq": 0.0891339582730985,
              "loadSteal": 0,
              "loadGuest": 0,
              "rawLoad": 3224983,
              "rawLoadUser": 592812,
              "rawLoadSystem": 2621390,
              "rawLoadNice": 0,
              "rawLoadIdle": 8870296,
              "rawLoadIrq": 10781,
              "rawLoadSteal": 0,
              "rawLoadGuest": 0,
              "load": 27.887769556238185,
              "loadUser": 7.581342770280893,
              "loadSystem": 20.240344527034405,
              "loadNice": 0,
              "loadIdle": 72.11223844976981,
              "loadIrq": 0.06080222291488842,
              "loadSteal": 0,
              "loadGuest": 0,
              "rawLoad": 3372123,
              "rawLoadUser": 916718,
              "rawLoadSystem": 2448146,
              "rawLoadNice": 0,
              "rawLoadIdle": 8719649,
              "rawLoadIrq": 7265,
              "rawLoadSteal": 0,
              "rawLoadGuest": 0,
              "load": 20.78635867735646,
              "loadUser": 6.224632949628744,
              "loadSystem": 14.485977288724676,
              "loadNice": 0,
              "loadIdle": 79.21364132264354,
              "loadIrq": 0.045748439083046315,
              "loadSteal": 0,
              "loadGuest": 0,
              "rawLoad": 2513877,
              "rawLoadUser": 756187,
              "rawLoadSystem": 1751359,
              "rawLoadNice": 0,
              "rawLoadIdle": 19576953,
              "rawLoadIrq": 5531,
              "rawLoadSteal": 0
            }
          ]
        }
      }
    }
  }
}
```

On utilise ce formalisme pour construire l'URL de l'API car elle est claire. On sait :

- Que l'on utilise un protocole http
- Que le serveur est en cours d'exécution sur un port donné sur une machine locale
- Qu'il s'agit de la première version d'une application
- Que cette application renvoie des informations sur le système

Ce formalisme permet d'avoir un routage simple.

- 6) Pour pouvoir écrire le jeu de test, j'ai commencé par installer Jest.

```
PS C:\Users\Administrateur\Desktop\tp-programmer> npm install jest
added 290 packages, and audited 291 packages in 51s
32 packages are looking for funding
run `npm fund` for details
found 0 vulnerabilities
PS C:\Users\Administrateur\Desktop\tp-programmer>
```

J'ai créé un fichier systemInformation.test.js avec le code suivant ;

```
const request = require('supertest');
const app = require('./app');

describe('API systemInformation tests', () => {
  test('GET /api/v1/systemInformation should return system information', async () => {
    const response = await request
      (app).get('/api/v1/systemInformation');
    expect(response.statusCode).toBe(200);
    expect(response.type).toBe('application/json');
    expect(response.body).toHaveProperty('cpu');
    expect(response.body).toHaveProperty('system');
    expect(response.body).toHaveProperty('mem');
    expect(response.body).toHaveProperty('os');
    expect(response.body).toHaveProperty('currentLoad');
    expect(response.body).toHaveProperty('processes');
```

```

    expect(response.body).toHaveProperty('diskLayout');
    expect(response.body).toHaveProperty('networkInterfaces');
  });
  //test ('GET /api/v1/unknown should 404 error', async()=>{
  //    const response = await request(server).get
  ('/api/v1/unknown');
  //    expect(response.statusCode).toBe(404);
  //});
});

```

## En exécutant le test on a

```

PS C:\Users\Administrateur\Desktop\tp-programmer\tp-programmer> npm test
> i-want-typescript@1.0.0 test
> jest

FAIL src/hello-world.spec.ts
  ● Test suite failed to run

    Jest encountered an unexpected token

    Jest failed to parse a file. This happens e.g. when your code or its dependencies use non-standard JavaScript syntax, or when Jest is not configured to support such syntax.

    Out of the box Jest supports Babel, which will be used to transform your files into valid JS based on your Babel configuration.

    By default "node_modules" folder is ignored by transformers.

    Here's what you can do:
    • If you are trying to use ECMAScript Modules, see https://jestjs.io/docs/ecmascript-modules for how to enable it.
    • If you are trying to use TypeScript, see https://jestjs.io/docs/getting-started#using-typescript
    • To have some of your "node_modules" files transformed, you can specify a custom "transformIgnorePatterns" in your config.
    • If you need a custom transformation specify a "transform" option in your config.
    • If you simply want to mock your non-JS modules (e.g. binary assets) you can stub them out with the "moduleNameMapper" config option.

    You'll find more details and examples of these config options in the docs:
    https://jestjs.io/docs/configuration
    For information about custom transformations, see:
    https://jestjs.io/docs/code-transformation

    Details:

    SyntaxError: Cannot use import statement outside a module

       at Runtime.createScriptFromCode (node_modules/jest-runtime/build/index.js:1449:14)

PASS dist/hello-world.spec.js
FAIL ./systemInformation.test.js
  ● Test suite failed to run

    Cannot find module './app' from 'systemInformation.test.js'

       1 | const request = require('supertest');
     >  2 | const app = require('./app');
         |           ^
       3 |
       4 | describe('API systemInformation tests', ()=>{
       5 |   test('GET /api/v1/systemInformation should return system information', async()=>{

       at Resolver._throwModNotFoundError (node_modules/jest-resolve/build/resolver.js:425:11)
       at Object.require (systemInformation.test.js:2:13)

Test Suites: 2 failed, 1 passed, 3 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        2.647 s
Ran all test suites.
PS C:\Users\Administrateur\Desktop\tp-programmer\tp-programmer>

```

On constate la présence d'une erreur lorsque j'exécute le programme et je ne sais pas comment la résoudre.



## TD2 : Conteneurisation avec Docker

- 1) Après avoir suivi toutes les étapes d'installation du docker sur ma machine virtuelle linux, j'obtiens au moment de la vérification de son fonctionnement :

```
ahmed@ahmed-VirtualBox:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2023-11-04 19:05:35 GMT; 41s ago
     TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
    Main PID: 37135 (dockerd)
      Tasks: 8
     Memory: 27.9M
        CPU: 513ms
    CGroup: /system.slice/docker.service
            └─37135 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
```

Et, lorsque que je veux exécuter la commande `sudo docker run hello-world`, j'obtiens le résultat suivant :

```
ahmed@ahmed-VirtualBox:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
719385e32844: Pull complete
Digest: sha256:88ec0acaa3ec199d3b7eaf73588f4518c25f9d34f58ce9a0df68429c5af48e8d
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```

Où il est possible de lire que le docker client est connecté au docker daemon.

- 2) J'ai créé un fichier Dockerfile avec la commande `nano` :

```
ahmed@ahmed-VirtualBox:~$ nano Dockerfile
```

```
//
```

```
ahmed@ahmed-VirtualBox:~$ nano DockerFile
```

```
//
```

Dans ce fichier, j'ai écrit le code suivant :



```
GNU nano 6.2
FROM alpine:3.15

WORKDIR /app

RUN apk add --update node.js npm procps
RUN adduser -D node
COPY package*.json

RUN npm install
COPY . .

RUN npm run build
USER node

CMD ["npm", "start", "node"]
```

//

```
#image de départ
FROM node:16-alpine

#chemin de travail
WORKDIR /usr/src/app

#installation des paquets système
RUN npm install run build

#Installation des dépendances
RUN npm ci --only=production

#Copier des fichiers
COPY package*.json ./

#Informer sur le port du conteneur
EXPOSE 8000

#Executer
CMD ["npm", "start", "node", "server.js"]
```

//

3) J'ai créé une image comme demandé :

```
ahmed@ahmed-VirtualBox:~$ sudo docker build . -t sysinfo-api:0.0.1
[sudo] Mot de passe de ahmed :
[+] Building 0.3s (2/2) FINISHED                                docker:default
=> [internal] load build definition from Dockerfile             0.1s
=> == transferring dockerfile: 235B                             0.0s
=> [internal] load .dockerignore                                0.0s
=> == transferring context: 2B                                   0.0s
Dockerfile:7
-----
5 | RUN apk add --update node.js npm procps
6 | RUN adduser -D node
7 | >>> COPY package*.json
8 |
9 | RUN npm install
-----
ERROR: failed to solve: dockerfile parse error on line 7: COPY requires at least two arguments, but only one was provided. Destination could not be determined
```

//

```
ahmed@ahmed-VirtualBox:~$ sudo docker build . -t sysinfo-api:0.0.1
[+] Building 13.9s (10/10) FINISHED                                docker:default
=> [internal] load build definition from Dockerfile             0.1s
=> == transferring dockerfile: 387B                             0.0s
=> [internal] load .dockerignore                                0.0s
=> == transferring context: 2B                                   0.0s
=> [internal] load metadata for docker.io/library/node:16-alpine 0.5s
=> [1/5] FROM docker.io/library/node:16-alpine@sha256:a1f9d027912b58a7c75be7716c97cfbc6d3099f3a97ed84aa490be9d 0.0s
=> [internal] load build context                                0.2s
=> == transferring context: 2B                                   0.2s
=> CACHED [2/5] WORKDIR /usr/src/app                           0.0s
=> [3/5] RUN npm install run build                             9.2s
=> [4/5] RUN npm ci --only=production                          2.9s
=> [5/5] COPY package*.json ./                                 0.4s
=> exporting to image                                           0.6s
=> == exporting layers                                           0.6s
=> == writing image sha256:008b9f391576c1aed4649924141e2de4cc8cf8c051e17a66b0deddd8eb64b4d7 0.0s
=> == naming to docker.io/library/sysinfo-api:0.0.1            0.0s
```

//

4) Création du conteneur :

```
ahmed@ahmed-VirtualBox:~$ sudo docker run -p 8123/8000 -m1024m --cpus=1 sysinfo-api:0.0.1
docker: Invalid proto: 8000.
See 'docker run --help'.
```

Signification des différents flags utilisés :

- -p 8123 :8000 : au niveau du conteneur, le port d'écoute est 8000 alors que sur la machine hôte le port d'écoute est 8123.
- -m1024m : la mémoire du conteneur sera limitée à 1024 méga octets.
- --cpus=1 : le conteneur sera limité à une seule unité de CPU.

Modifier ces valeurs pourrait avoir un impact sur l'application car :

- Si on utilise un numéro de port déjà utilisé, l'accès à l'application pourrait se trouver bloqué.
- Si la mémoire est insuffisante, l'application pourrait devenir lente.
- Si le nombre de CPU est trop petit, les performances pourraient être diminuée.

## 5) Inspection des images :

```
ahmed@ahmed-VirtualBox:~$ sudo docker image history sysinfo-api:0.0.1
```

IMAGE	CREATED	CREATED BY	SIZE	COMMENT
008b9f391576	13 days ago	CMD ["npm" "start" "node" "server.js"]	0B	buildkit.dockerfile.v0
<missing>	13 days ago	EXPOSE map[8000/tcp:{}]	0B	buildkit.dockerfile.v0
<missing>	13 days ago	COPY package*.json ./ # buildkit	0B	buildkit.dockerfile.v0
<missing>	13 days ago	RUN /bin/sh -c npm ci --only=production # bu...	3.61MB	buildkit.dockerfile.v0
<missing>	13 days ago	RUN /bin/sh -c npm install run build # build...	8.03MB	buildkit.dockerfile.v0
<missing>	13 days ago	WORKDIR /usr/src/app	0B	buildkit.dockerfile.v0
<missing>	3 months ago	/bin/sh -c #(nop) CMD ["node"]	0B	
<missing>	3 months ago	/bin/sh -c #(nop) ENTRYPOINT ["docker-entry...	0B	
<missing>	3 months ago	/bin/sh -c #(nop) COPY file:4d192565a7220e13...	388B	
<missing>	3 months ago	/bin/sh -c apk add --no-cache --virtual .bui...	7.76MB	
<missing>	3 months ago	/bin/sh -c #(nop) ENV YARN_VERSION=1.22.19	0B	
<missing>	3 months ago	/bin/sh -c addgroup -g 1000 node && addu...	102MB	
<missing>	3 months ago	/bin/sh -c #(nop) ENV NODE_VERSION=16.20.2	0B	
<missing>	3 months ago	/bin/sh -c #(nop) CMD ["/bin/sh"]	0B	
<missing>	3 months ago	/bin/sh -c #(nop) ADD file:32ff5e7a78b890996...	7.34MB	

Après avoir suivi les différentes instructions, j'ai utilisé la commande dive :

Layers		Current Layer Contents		Size	Filetree
Cmp	Size	Permission	UID:GID	Size	Filetree
7.3 MB	FROM c84e4ec3858c91e	drwxr-xr-x	0:0	817 kB	bin
102 MB	addgroup -g 1000 node	-rwxrwxrwx	0:0	0 B	arch → /bin/busybox
7.8 MB	apk add --no-cache --virtual .build-deps-yarn curl gnupg tar && export	-rwxrwxrwx	0:0	0 B	ash → /bin/busybox
388 B	#(nop) COPY file:4d192565a7220e135cab6c77fbc1c73211b69f3d9fb37e62857b2c6	-rwxrwxrwx	0:0	0 B	base64 → /bin/busybox
0 B	WORKDIR /usr/src/app	-rwxrwxrwx	0:0	0 B	bbconfig → /bin/busybox
8.0 MB	RUN /bin/sh -c npm install run build # buildkit	-rwxrwxrwx	0:0	817 kB	busybox
3.6 MB	RUN /bin/sh -c npm ci --only=production # buildkit	-rwxrwxrwx	0:0	0 B	cat → /bin/busybox
0 B	COPY package*.json ./ # buildkit	-rwxrwxrwx	0:0	0 B	chattr → /bin/busybox
		-rwxrwxrwx	0:0	0 B	chgrp → /bin/busybox
		-rwxrwxrwx	0:0	0 B	chmod → /bin/busybox
		-rwxrwxrwx	0:0	0 B	chown → /bin/busybox
		-rwxrwxrwx	0:0	0 B	cp → /bin/busybox
		-rwxrwxrwx	0:0	0 B	date → /bin/busybox
		-rwxrwxrwx	0:0	0 B	dd → /bin/busybox
		-rwxrwxrwx	0:0	0 B	df → /bin/busybox
		-rwxrwxrwx	0:0	0 B	dmesg → /bin/busybox
		-rwxrwxrwx	0:0	0 B	dnsdomainname → /bin/busybox
		-rwxrwxrwx	0:0	0 B	dumpkmap → /bin/busybox
		-rwxrwxrwx	0:0	0 B	echo → /bin/busybox
		-rwxrwxrwx	0:0	0 B	ed → /bin/busybox
		-rwxrwxrwx	0:0	0 B	egrep → /bin/busybox
		-rwxrwxrwx	0:0	0 B	false → /bin/busybox
		-rwxrwxrwx	0:0	0 B	fattr → /bin/busybox
		-rwxrwxrwx	0:0	0 B	fdflush → /bin/busybox
		-rwxrwxrwx	0:0	0 B	fgrep → /bin/busybox
		-rwxrwxrwx	0:0	0 B	fsync → /bin/busybox
		-rwxrwxrwx	0:0	0 B	getopt → /bin/busybox
		-rwxrwxrwx	0:0	0 B	grep → /bin/busybox
		-rwxrwxrwx	0:0	0 B	gunzip → /bin/busybox
		-rwxrwxrwx	0:0	0 B	gzip → /bin/busybox
		-rwxrwxrwx	0:0	0 B	hostname → /bin/busybox
		-rwxrwxrwx	0:0	0 B	ionice → /bin/busybox
		-rwxrwxrwx	0:0	0 B	iostat → /bin/busybox
		-rwxrwxrwx	0:0	0 B	ipcalc → /bin/busybox
		-rwxrwxrwx	0:0	0 B	kbd_node → /bin/busybox
		-rwxrwxrwx	0:0	0 B	kill → /bin/busybox
		-rwxrwxrwx	0:0	0 B	link → /bin/busybox
		-rwxrwxrwx	0:0	0 B	linux32 → /bin/busybox
		-rwxrwxrwx	0:0	0 B	linux64 → /bin/busybox
		-rwxrwxrwx	0:0	0 B	ln → /bin/busybox
		-rwxrwxrwx	0:0	0 B	login → /bin/busybox
		-rwxrwxrwx	0:0	0 B	ls → /bin/busybox
		-rwxrwxrwx	0:0	0 B	lsattr → /bin/busybox
		-rwxrwxrwx	0:0	0 B	lzop → /bin/busybox
		-rwxrwxrwx	0:0	0 B	maketime → /bin/busybox

Count	Total Space	Path
3	644 kB	/etc/ssl/certs/ca-certificates.crt
2	450 kB	/usr/src/app/node_modules/async/dist/async.js
2	401 kB	/usr/src/app/node_modules/async/dist/async.mjs
2	224 kB	/usr/src/app/node_modules/timespan/docs/time-span.html
2	185 kB	/usr/src/app/node_modules/run/test/screenshot.png
2	149 kB	/usr/src/app/node_modules/uglify-js/lib/process.js
2	143 kB	/usr/src/app/node_modules/minimatch/dist/mjs/index.js.map
2	143 kB	/usr/src/app/node_modules/minimatch/dist/cjs/index.js.map
2	110 kB	/usr/src/app/node_modules/uglify-js/lib/consolidator.js
2	104 kB	/usr/src/app/node_modules/js-yaml/support/pyyaml-src/scanner.py
2	103 kB	/usr/src/app/node_modules/ts-arrayish/yarn-error.log
2	102 kB	/usr/src/app/node_modules/moo-server/mootools.js
2	101 kB	/usr/src/app/node_modules/js-yaml/lib/js-yaml/scanner.js
2	93 kB	/usr/src/app/node_modules/uglify-js/lib/parse.js.js
2	86 kB	/usr/src/app/node_modules/js-yaml/support/pyyaml-src/emitter.py
2	85 kB	/usr/src/app/node_modules/uglify-js/test/testconsolidator.js
2	82 kB	/usr/src/app/node_modules/uglify-js/README.html
2	81 kB	/usr/src/app/node_modules/minimatch/dist/cjs/index.js
2	80 kB	/usr/src/app/node_modules/minimatch/dist/mjs/ast.js.map
2	80 kB	/usr/src/app/node_modules/minimatch/dist/cjs/ast.js.map

Image Details	
Image name: sysinfo-api:0.0.1	
Total image size: 129 MB	
Potential wasted space: 7.9 MB	
Image efficiency score: 96 %	

Tags	Id	Digest	Command
(unavailable)	c84e4ec3858c91e5f506f811e5f5208b5a187f83356303a4af6c314e61c2	sha256:4693057ce2364720d39e57e85a5b8e0bd9ac3573716237736d6470ec5b7b7230	#(nop) ADD file:32ff5e7a78b890996ee4681cc0a26185d3e9acdb4eb1e2aacbc2411f922fed6b in

```
Image Source: docker://sysinfo-api:0.0.1
Fetching image... (this can take a while for large images)
Analyzing image...
Building cache...
```

Mais les images sont trop grandes et cela peut prendre du temps.

Il serait possible de réduire la taille des images en ;

- En utilisant des images de départ plus petites.
- En ne gardant que les ou l'image nécessaire.
- En apportant des modifications au fichier Dockerfile comme en regroupant les commandes run ou bien encore en supprimant les fichiers caches.

## 6) J'ai modifié le fichier Dockerfile :

```
FROM alpine:3.15
WORKDIR /app
RUN apk add --no-cache node.js npm
COPY package*.json
RUN npm install
COPY . .
RUN npm run build

FROM alpine:3.15 as runner
WORKDIR /app
RUN apk add --no-cache node.js
RUN adduser -D node
COPY --from-builder --chown=node /app/node_modules ./node_modules
COPY --from-builder --chown=node /app/dist ./dist
USER node

CMD ["node", "dist/index.js"]
```

//

```
#image de départ
FROM node:16-alpine as builder

WORKDIR /app

COPY package*.json ./

RUN npm install
RUN npm run build

FROM alpine:3.15 as runner

RUN apk add --no-cache node.js npm && \
    adduser -D node && \
    npm install --global npm

WORKDIR /app

COPY --from=builder /app/node_modules ./node_modules
COPY --from=builder /app/build ./build
COPY --from=builder /app/package*.json ./

USER node

CMD ["node", "build/server"]
```

//

- 7) J'ai ensuite essayé de publier l'image de Docker sur mon compte DockerHub Ahmed11uu

```
ahmed@ahmed-VirtualBox:~$ sudo docker tag sysinfo-api:0.0.2 Ahmed11uu/sysinfo-api:0.0.2
Error parsing reference: "Ahmed11uu/sysinfo-api:0.0.2" is not a valid repository/tag: invalid reference format: repository name must be lowercase
```

Mais un problème persiste.

Pour publier cette image, j'aurai utilisé la commande `sudo docker push Ahmed11uu/sysinfo-api : 0.0.2`.



## TD3 : CI/CD avec GitHub

1) Après avoir suivi toutes les étapes décrites dans le tutoriel de GitHub Actions, j'obtiens :

The screenshot shows the GitHub Actions interface for a workflow named 'Explore-GitHub-Actions'. The workflow is in a 'succeeded' state, having completed 3 hours ago in 3 seconds. The interface includes a sidebar with navigation options: Summary, Jobs, Run details, Usage, and Workflow file. The 'Jobs' section is active, showing a list of jobs. The main panel displays the job details for 'Explore-GitHub-Actions', which is a 'Set up job' job. The job log shows the following steps:

- Set up job (1s)
- Run echo "🚀 The job was automatically triggered by a push event." (0s)
- Run echo "🌐 This job is now running on a Linux server hosted by GitHub!" (0s)
- Run echo "💡 The name of your branch is refs/heads/octocat-patch-l-1 and your rep..." (0s)
- Check out repository code (1s)
- Run echo "💡 The Ahmed11uu/.github-workflows repository has been cloned to the r..." (0s)
- Run echo "🖨 The workflow is now ready to test your code on the runner." (0s)
- List files in the repository (0s)
- Run echo "🍏 This job's status is success." (0s)
- Post Check out repository code (0s)
- Complete job (0s)

2) Dans le dossier `-github/workflows`, j'ai créé un fichier `github-actions-TD3.yml` contenant le code suivant :

```
name: TD3
on: [ push ]
jobs:
  build:
    runs-on: ubuntu-latest
    strategy:
      matrix:
        node-version: ['18.18.2']
    steps:
      - uses: actions/checkout@v3
      - name: Setup Node.js
        uses: actions/setup-node@v3
        with:
          node-version: ${ matrix.node-version }
          cache: 'npm'
      - name: Install dependencies
        run: npm ci
      - name: Build
        run: npm run build
      - name: Run tests
        run: npm test
      - name: Start application
        run: npm run start &
    env:
      CI: true
```



```

  40  .github/workflows/github-actions-TP3.yml
...  ...  @@ -0,0 +1,40 @@
1  + name: Github-Action-TP3
2  + on:
3  +   push:
4  +     branches:
5  +       - master
6  +
7  + jobs:
8  +   run-install:
9  +     name: Run Install
10  +     runs-on: ubuntu-latest
11  +
12  +     steps:
13  +       - uses: actions/checkout@v2
14  +
15  +       - name: Use Node.js 18.x
16  +         uses: actions/setup-node@v2
17  +         with:
18  +           node-version: 18.x
19  +           cache: 'npm'
20  +           cache-dependency-path: package-lock.json
21  +
22  +       - name: Install dependencies
23  +         run: npm ci
24  +
25  +   run-action:
26  +     name: Run Action
27  +     runs-on: ubuntu-latest
28  +
29  +     steps:
30  +       - uses: actions/checkout@v2
31  +
32  +       - name: Use Node.js 18.x
33  +         uses: actions/setup-node@v2
34  +         with:
35  +           node-version: 18.x
36  +
37  +       - name: Build
38  +         run: npm ci
39  +
40  +       - name: Run my test
41  +         run: npm run test

```

//Mais, lorsque j'appuie sur action pour lancer la création de ce code, il y a un problème de construction au niveau de la version de node.js.

✖

Create githut-actions-TD3.yml #1

Re-run jobs

...

Summary

Jobs

✖

build (18.18.2)

Run details

Usage

Workflow file

build (18.18.2)

failed 4 minutes ago in 5s

Search logs

🔄

⚙️

> ✔️ Set up job1s

> ✔️ Run actions/checkout@v31s

> ❌ Setup Node.js1s

1 ▶ Run actions/setup-node@v3

8 Found in cache @ /opt/hostedtoolcache/node/18.18.2/x64

9 ▶ Environment details

13 /opt/hostedtoolcache/node/18.18.2/x64/bin/npm config get cache

14 /home/runner/.npm

15 Error: Dependencies lock file is not found in /home/runner/work/.github-workflows/.github-workflows. Supported file patterns: package-lock.json,npm-shrinkwrap.json,yarn.lock

✔️ Install dependencies0s

✔️ Build0s

✔️ Run tests0s

✔️ Start application0s

✔️ Post Setup Node.js0s

> ✔️ Post Run actions/checkout@v30s

> ✔️ Complete job0s


Ma machine dispose pourtant bien de la version 18.18.2.


```
Node.js v18.18.2
//
```

J'obtiens :


17

# Create github-actions-TP3.yml #2


 Open

Ahmed11uu wants to merge 1 commit into `main` from `octocat-patch-1-3` 

 Conversation 0

 Commits 1

 Checks 0

 Files changed 1



Ahmed11uu commented 4 minutes ago

Owner ...

No description provided.



  Create github-actions-TP3.yml

Verified

eb3044f

Add more commits by pushing to the `octocat-patch-1-3` branch on [Ahmed11uu/github-action-TP3](#).



Require approval from specific reviewers before merging

[Branch protection rules](#) ensure specific people approve pull requests before they're merged.

Add rule



This branch has no conflicts with the base branch

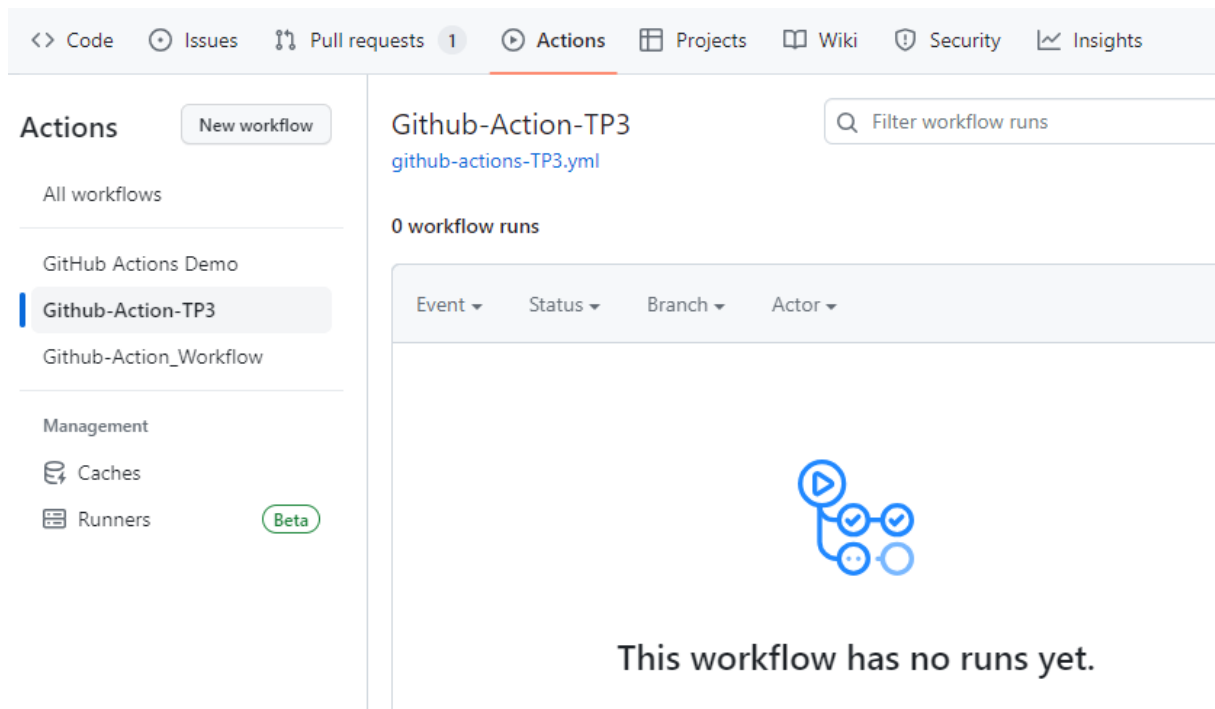
Merging can be performed automatically.

Merge pull request

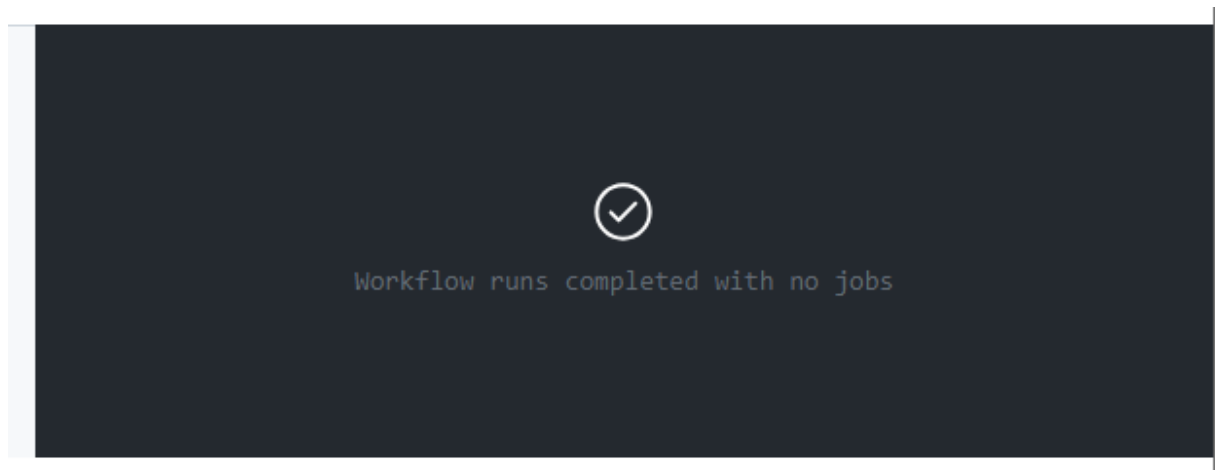


You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

Mais, lorsque j'appuie sur action pour lancer la création de ce code, il y a un message qui m'indique que le travail n'a pas encore été exécuté.

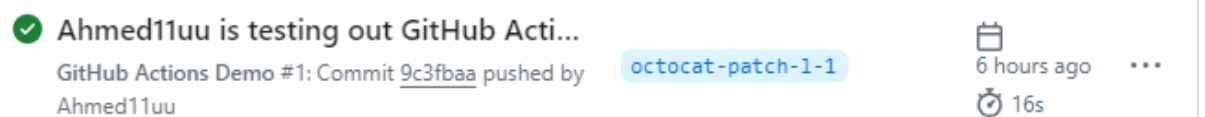


Et, au niveau du Cheks, je peux voir que les exécutions du worflow se sont terminées sans tâches :



- 3) Pour vérifier le bon fonctionnement du code écrit, il faut cliquer sur actions dans GitHub.

Si le fonctionnement est correct, on devrait avoir un point vert comme ci-dessous :



Et au niveau du shecks, je devrais avoir un résultat similaire à celui-ci-dessous :

Create github-action-demo.yml 53adab1

Re-run all jobs

GitHub Actions Demo  
on: push

Explore-GitHub-Actions

### Explore-GitHub-Actions

succeeded 49 minutes ago in 6s

- > Set up job 0s
- > Run echo "🚀 The job was automatically triggered by a push event." 1s
- > Run echo "🌐 This job is now running on a Linux server hosted by GitHub!" 0s
- > Run echo "🔗 The name of your branch is refs/heads/octocat-patch-1-2 a... 0s
- > Check out repository code 0s
- > Run echo "💡 The Ahmed11uu/github-action-TP3 repository has been clo... 0s
- > Run echo "💻 The workflow is now ready to test your code on the runner." 0s
- > List files in the repository 0s
- > Run echo "🍏 This job's status is success." 0s
- > Post Check out repository code 0s
- > Complete job 0s

//Et non un point rouge comme ci-dessous :

Create githut-actions-TD3.yml

TD3 #1: Commit d5be7bd pushed by Ahmed11uu

Ahmed11uu-patch-1

18 minutes ago ...  
 15s

Et lorsqu'on lance l'exécution du workflow, on peut voir son déroulement, son fonctionnement en temps réel.//

## TD4 : Déploiement sur PaaS avec Fly.io

- 1) En suivant le documentation Fly Docs donnée, j'ai utilisé la ligne de commande suivante :

```
C:\Users\Administrateur> powershell -Command "iwr https://fly.io/install.ps1 -useb | iex"  
Downloading flyctl...
```

Flyctl a été installé correctement sur la machine :

```
flyctl was installed successfully to C:\Users\Administrateur\.fly\bin\flyctl.exe  
Run 'flyctl --help' to get started
```

- 2) J'ai commencé par m'authentifier sur l'application Fly.io

```
C:\Users\Administrateur>flyctl auth login  
Opening https://fly.io/app/auth/cli/d54861c2d502df92001af89e270f9144 ...  
Waiting for session... Done  
successfully logged in as aamd9118@gmail.com
```

Etant déjà authentifié au niveau de l'application Docker, j'ai entré la ligne de commande indiquée dans la documentation.

```
C:\Users\Administrateur>flyctl auth docker [flags]  
Authentication successful. You can now tag and push images to registry.fly.io/{your-app}
```

- 3) J'ai publié une image du Docker sur le Fly.io :

```
C:\Users\Administrateur>docker tag apache/hadoop:3.3.5 registry.fly.io/apache/hadoop:3.3.5

C:\Users\Administrateur>docker push registry.fly.io/apache/hadoop:3.3.5
The push refers to repository [registry.fly.io/apache/hadoop]
f6a0e0da2441: Preparing
fc093b0a5145: Preparing
5f70bf18a086: Preparing
4137dd174293: Preparing
a1b13947e810: Preparing
0f28285923f8: Waiting
f83d2a17eaab: Waiting
ed1a1c70f454: Waiting
fdbdd5a1f11f: Waiting
57d1b07c8254: Waiting
32c55d5f44bd: Waiting
c7fc2d735bbc: Waiting
77c76fa09ae5: Waiting
7afb22aa16c8: Waiting
bb0818abbbc2: Waiting
91cbc41ea724: Waiting
b76b23935a3b: Waiting
dd41d0f944ac: Waiting
164052d30ef1: Waiting
9006ada31dc9: Waiting
a7e55935cf0b: Waiting
d69483a6face: Waiting
error parsing HTTP 404 response body: invalid character 'p' after top-level value: "404 page not found\n"
```

4) Je veux dans un premier temps créer mon application :

```
C:\Users\Administrateur>flyctl apps create myapp
automatically selected personal organization: Alotaibi
Error: We need your payment information to continue! Add a credit card or buy credit: https://fly.io/dashboard/alotaibi/billing
```

Mais, on me demande de rentrer mes coordonnées bancaires pour le déploiement de l'application.

Ensuite, j'aurai écrit :

```
C:\Users\Administrateur>flyctl deploy -app myapp --image registry.fly.io/myapp:3.3.5
Error: failed changing working directory: chdir C:\Users\Administrateur\myapp: The system cannot find the file specified.
```

5) Pour visiter mon application, j'utilise :

```
C:\Users\Administrateur>curl https://myapp.fly.dev
curl: (6) Could not resolve host: myapp.fly.dev
```

Pour inspecter les journaux de l'application, il faudrait écrire :

```
C:\Users\Administrateur>flyctl logs --app myapp
Error: 401 Unauthorized
```

Les journaux de l'application peuvent nous donner des informations sur la nature des problèmes rencontrés comme :

- Des problèmes de connexion
- Des problèmes de configuration
- Des problèmes de ressources comme la mémoire ou le nombre de CPU