PROJET INTEGRATION DES SYSTEMES D'INFORMATION

RAPPORT

ELABORE PAR

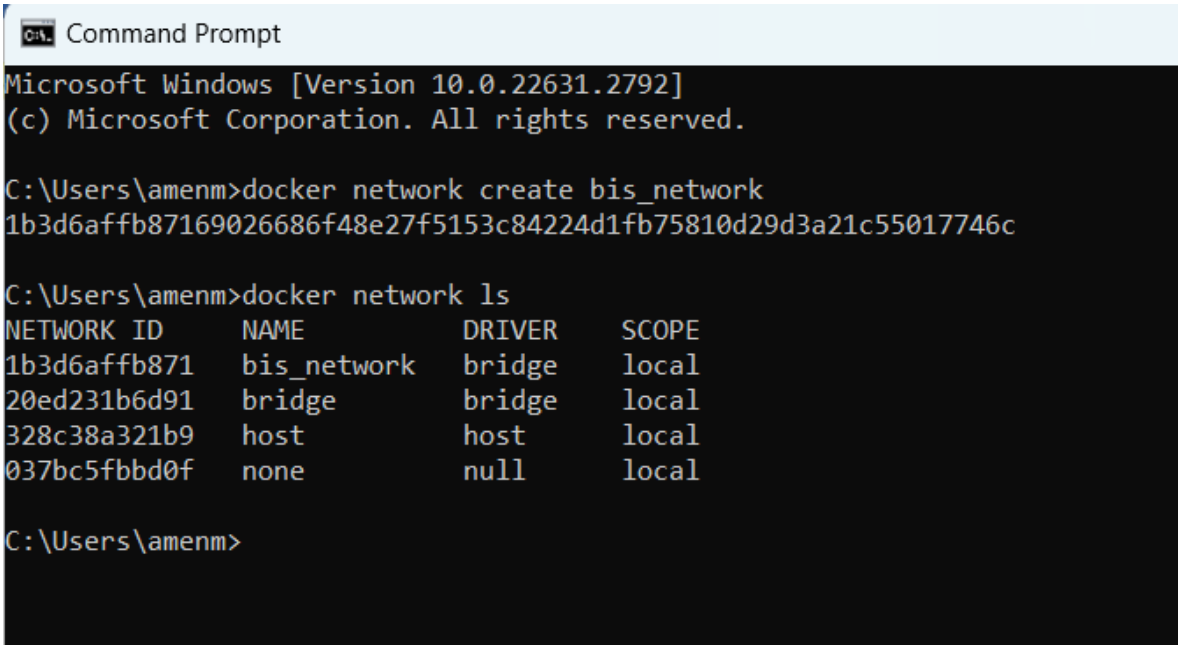
MRAD AMENALLAH

BEN ALI FOURAT

NIVEAU D'ETUDE

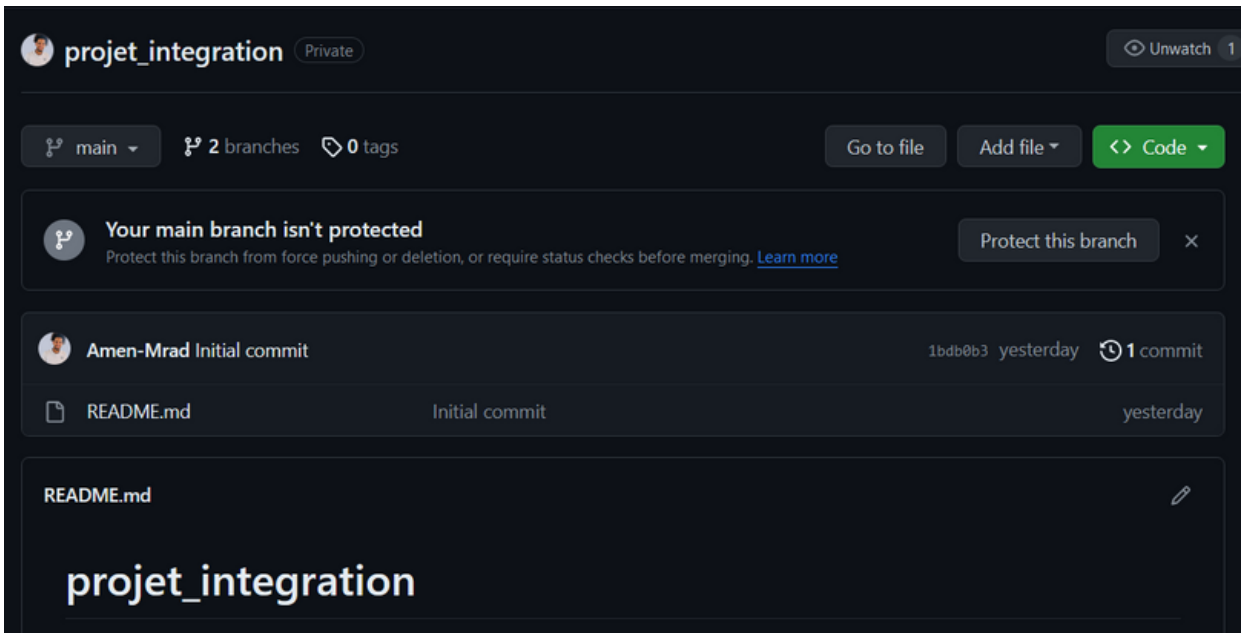
3 BIS 2

1- Créer un réseau docker (bis_network) (Tous les conteneurs qui seront créés appartiendront à ce réseau)



La commande “docker network create bis_network” faire la création du réseau “bis_network” .

2- Créer un repo github et une branche develop



3- Créer un conteneur Jenkins avec les paramètres suivants :

-Partager les ports de gestion des jobs avec le localhost.
-Exposer les ports de communication avec les autres conteneurs.
-Créer 2 volumes pour les fichiers de configuration et les fichiers de données.

```
C:\Users\amenm>docker run --name Jenkins_projet_container -p 8080:8060 -p 50000:50000 --network bis_network -v jenkins_config:/var/Je
nkins_home/config -v jenkins_data:/var/Jenkins_home/data -d jenkins/jenkins
Unable to find image 'jenkins/jenkins:latest' locally
latest: Pulling from jenkins/jenkins
90e5e7d8b87a: Pull complete
0027de9dea4b: Pull complete
878961003307: Pull complete
a066500963e3: Pull complete
1b8599e5f393: Pull complete
4038a9f4acc3: Pull complete
c14feb3b2f69: Pull complete
f26b2089197c: Pull complete
b9103bd4fe45: Pull complete
038a6010845f: Pull complete
0db960315795: Pull complete
098588bad7fb: Pull complete
Digest: sha256:29cecd36f3113efda690b5f117cc51a6ffe7da3b8eaed1bddf87319940283b5d
Status: Downloaded newer image for jenkins/jenkins:latest
62754df0d5fe829d42888b49391819c1b07baebe25667e4faed0f9e0468037c4
```

-p 8080:8060 : Cela fait correspondre le port 8060 du conteneur au port 8080 de la machine hôte. Cela signifie que vous pouvez accéder au serveur Jenkins en cours d'exécution à l'intérieur du conteneur en visitant http://localhost:8080 sur la machine hôte.

-p 50000:50000 : Cela fait correspondre le port 50000 du conteneur au port 50000 de la machine hôte. Ce port est utilisé par Jenkins pour la communication avec d'autres nœuds Jenkins, tels que les esclaves.

-v jenkins_config:/var/Jenkins_home/config : Cela fait correspondre le répertoire jenkins_config de la machine hôte au répertoire /var/Jenkins_home/config à l'intérieur du conteneur. Cela vous permet de conserver la configuration de Jenkins entre les redémarrages du conteneur.

–v jenkins_data:/var/Jenkins_home/data : Cela mappe le répertoire jenkins_data de la machine hôte sur le répertoire /var/Jenkins_home/data à l'intérieur du conteneur. Cela vous permet de persister les données de Jenkins entre les redémarrages du conteneur.

-d jenkins/jenkins : Cela spécifie l'image à utiliser pour le conteneur. Dans ce cas, il s'agit de l'image jenkins/jenkins provenant de Docker Hub. L'option -d spécifie que le conteneur doit être exécuté en mode détaché, ce qui signifie qu'il s'exécutera en arrière-plan.

4- Créer un conteneur mongodb

-Partager les ports d'interrogation avec le localhost

```
C:\Users\amenm>docker run -p 27017:27017 --network bis_network --name mongodb_isi -d mongo
Unable to find image 'mongo:latest' locally
latest: Pulling from library/mongo
cbe3537751ce: Pull complete
a80d99d2ce19: Pull complete
cdb44dc221f3: Pull complete
52cece2eeeb6: Pull complete
9484737e86c4: Pull complete
43ad935b75c0: Pull complete
a3ac6a8edff6: Pull complete
90580617c703: Pull complete
3c932f959341: Pull complete
Digest: sha256:b679b96ec8a2692ebb6f7622b9097974c1f751b413b3db5a0629a244ae2c6950
Status: Downloaded newer image for mongo:latest
ef93ca9fa9c26577513c257e973316668dcab244fae495fea0398b03b638e1ac
```

La commande "docker run -p 27017:27017 --network bis_network --name mongodb_isi -d mongo" faire la création du réseau "bis_network" faire la création du conteneur Mongodb.

5- Créer deux conteneurs web nginx

-Partager les ports 80 et 443 pour exposer une application de votre choix (une page html est suffisante pour visualiser les données de la collection restaurants) => il faut éviter le conflit de ports.

La création du premiére conteneurs :

```
C:\Users\amenm>docker run -p 80:80 --network bis-network --name nginx1 -d nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
1f7ce2fa46ab: Pull complete
9b16c94bb686: Pull complete
9a59d19f9c5b: Pull complete
9ea27b074f71: Pull complete
c6edf33e2524: Pull complete
84b1ff10387b: Pull complete
517357831967: Pull complete
Digest: sha256:10d1f5b58f74683ad34eb29287e07dab1e90f10af243f151bb50aa5dbb4d62ee
Status: Downloaded newer image for nginx:latest
2cf1d3d544547a758b152ace98530ef3d3a8866cd9caf456a466c8f38d58724b
```

La création du deuxiéme conteneurs :

```
Users\amenm>docker run -p 443:443  --network bis_network --name nginx2 -d ngi
1e8564aec40b6cc390b97a4722fc7f61cfa085c8007e180b560ef190c91da
```

Creation d'un fichier HTML e l'editer pour afficher le contenu du fichier JSON

```
C:\Users\amenm\Desktop>cd C:\Users\amenm\OneDrive\Bureau\integration

C:\Users\amenm\OneDrive\Bureau\integration>
C:\Users\amenm\OneDrive\Bureau\integration>docker cp restaurants.json nginx1:/usr/share/nginx/html/
Successfully copied 2.03MB to nginx1:/usr/share/nginx/html/

C:\Users\amenm\OneDrive\Bureau\integration>docker exec -it nginx1 bash
Error response from daemon: Container 1a6526cb99d0741ea459bf2cbe8b934efdf15d3b79e39b4393a5937b3f17be48 is not running

C:\Users\amenm\OneDrive\Bureau\integration>docker exec -it nginx1 bash
root@1a6526cb99d0:/# cd /usr/share/nginx/html
root@1a6526cb99d0:/usr/share/nginx/html#  touch index.html
root@1a6526cb99d0:/usr/share/nginx/html# cat - > /usr/share/nginx/html/index.html
```

```
root@1a6526cb99d0:/usr/share/nginx/html# touch index.html
root@1a6526cb99d0:/usr/share/nginx/html# cat - > /usr/share/nginx/html/index.html
<!DOCTYPE html>
<html>
<head>
<title>JSON File Display</title>
</head>
<body>
<h1>JSON File Display</h1>
<div id="json-display"></div>
<script>
fetch('restaurants.json')
.then(response => response.json())
.then(data => {
document.getElementById('json-display').innerHTML = JSON.stringify(data, null, 2);
});
</script>
</body>
</html>
root@1a6526cb99d0:/usr/share/nginx/html# service nginx restart
Restarting nginx: nginx
C:\Users\amenm\OneDrive\Bureau\integration>
```

"docker cp restaurants.json nginx1:/usr/share/nginx/html" cette commande va copier le fichier JSON dans le conteneur nginx

"docker exec -it nginx1 bash" Il vous permet d'exécuter une commande dans un conteneur Docker en cours d'exécution. Le drapeau -i signifie "interactif" et le drapeau -t signifie "tty" (terminal). Lorsqu'ils sont combinés, ces drapeaux vous permettent d'exécuter une session interactive en ligne de commande dans le conteneur.

La même étape pour le deuxiéme conteneur:

```
C:\Users\amenm\OneDrive\Bureau\integration>docker cp restaurants.json nginx2:/usr/share/nginx/html/
Successfully copied 2.03MB to nginx2:/usr/share/nginx/html/

C:\Users\amenm\OneDrive\Bureau\integration>docker exec -it nginx2 bash
Error response from daemon: Container ea01e8564aec40b6cc390b97a4722fc7f61cfa085c8007e180b560ef190c91da is not running

C:\Users\amenm\OneDrive\Bureau\integration>docker exec -it nginx2 bash
root@ea01e8564aec:/# cd /usr/share/nginx/html
root@ea01e8564aec:/usr/share/nginx/html#
root@ea01e8564aec:/usr/share/nginx/html#
root@ea01e8564aec:/usr/share/nginx/html# touch insex.html
root@ea01e8564aec:/usr/share/nginx/html# touch index.html
root@ea01e8564aec:/usr/share/nginx/html# cat - > /usr/share/nginx/html/index.html
<!DOCTYPE html>
<html>
<head>
<title>JSON File Display</title>
</head>
<body>
<h1>JSON File Display</h1>
<div id="json-display"></div>
<script>
fetch('restaurants.json')
.then(response => response.json())
```

```
<title>JSON File Display</title>
</head>
<body>
<h1>JSON File Display</h1>
<div id="json-display"></div>
<script>
fetch('restaurants.json')
.then(response => response.json())
.then(data => {
document.getElementById('json-display').innerHTML = JSON.stringify(data, null, 2);
});
</script>
</body>
</html>
root@ea01e8564aec:/usr/share/nginx/html# service nginx restart
Restarting nginx: nginx
C:\Users\amenm\OneDrive\Bureau\integration>
```

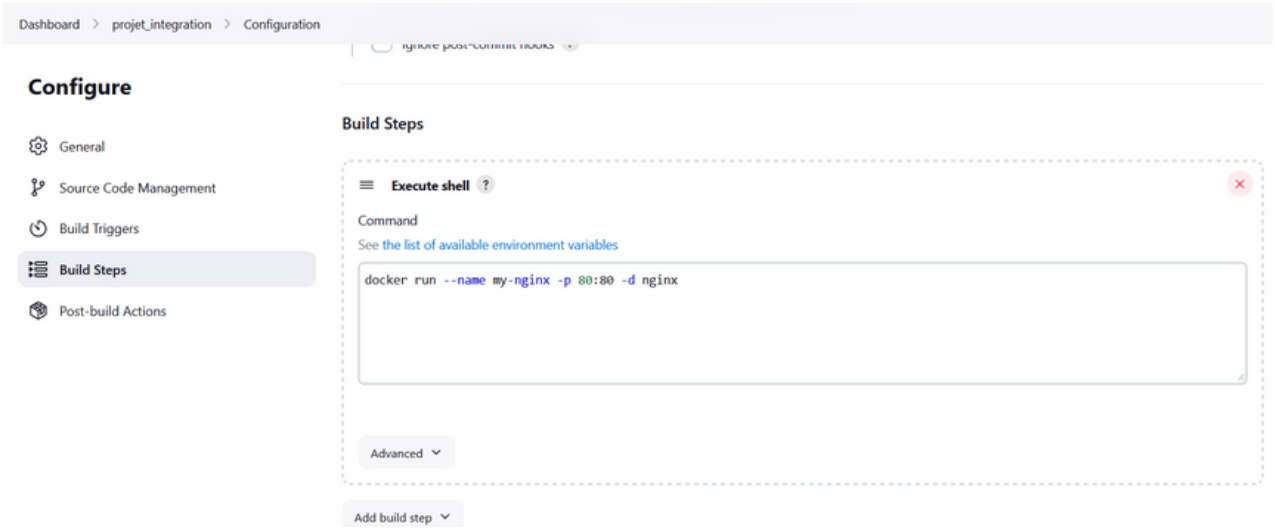
L’affichage :

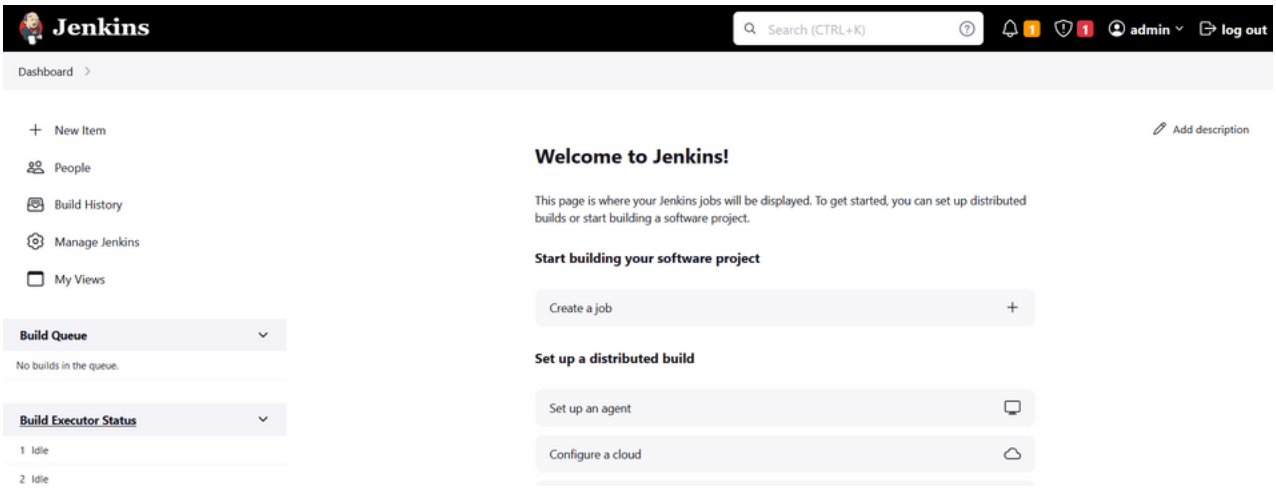
```
C:\Users\amenm\OneDrive\Bureau\integration>docker start nginx1
nginx1

C:\Users\amenm\OneDrive\Bureau\integration>docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS         PORTS                  NAMES
1a6526cb99d0   nginx     "/docker-entrypoint.…"   20 hours ago   Up 8 seconds   0.0.0.0:80->80/tcp     nginx1

C:\Users\amenm\OneDrive\Bureau\integration>docker start nginx2
nginx2

C:\Users\amenm\OneDrive\Bureau\integration>docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS         PORTS                                  NAMES
ea01e8564aec   nginx     "/docker-entrypoint.…"   20 hours ago   Up 2 seconds   80/tcp, 0.0.0.0:443->443/tcp           nginx2
1a6526cb99d0   nginx     "/docker-entrypoint.…"   20 hours ago   Up 38 seconds  0.0.0.0:80->80/tcp                     nginx1

C:\Users\amenm\OneDrive\Bureau\integration>
```


[{ "address": { "building": "1007", "coord": [-73.856077, 40.848447], "street": "Morris Park Ave", "zipcode": "10462" }, "borough": "Bronx", "cuisine": "Bakery", "grades": [{ "date": { "$date": 1393804800000 }, "grade": "A", "score": 2 }, { "date": { "$date": 1378857600000 }, "grade": "A", "score": 6 }, { "date": { "$date": 1358985600000 }, "grade": "A", "score": 10 }, { "date": { "$date": 1322006400000 }, "grade": "A", "score": 9 }, { "date": { "$date": 1299715200000 }, "grade": "B", "score": 14 }], "name": "Morris Park Bake Shop", "restaurant_id": "30075445" }, { "address": { "building": "469", "coord": [-73.961704, 40.662942], "street": "Flatbush Avenue", "zipcode": "11225" }, "borough": "Brooklyn", "cuisine": "Hamburgers", "grades": [{ "date": { "$date": 1419897600000 }, "grade": "A", "score": 8 }, { "date": { "$date": 1404172800000 }, "grade": "B", "score": 23 }, { "date": { "$date": 1367280000000 }, "grade": "A", "score": 12 }, { "date": { "$date": 1336435200000 }, "grade": "A", "score": 12 }], "name": "Wendy'S", "restaurant_id": "30112340" }, { "address": { "building": "351", "coord": [-73.98513559999999, 40.7676919], "street": "West 57 Street", "zipcode": "10019" }, "borough": "Manhattan", "cuisine": "Irish", "grades": [{ "date": { "$date": 1409961600000 }, "grade": "A", "score": 2 }, { "date": { "$date": 1374451200000 }, "grade": "A", "score": 11 }, { "date": { "$date": 1343692800000 }, "grade": "A", "score": 12 }, { "date": { "$date": 1325116800000 }, "grade": "A", "score": 12 }], "name": "Dj Reynolds Pub And Restaurant", "restaurant_id": "30191841" }, { "address": { "building": "2780", "coord": [-73.98241999999999, 40.579505], "street": "Stillwell Avenue", "zipcode": "11224" }, "borough": "Brooklyn", "cuisine": "American ", "grades": [{ "date": { "$date": 1402358400000 }, "grade": "A", "score": 5 }, { "date": { "$date": 1370390400000 }, "grade": "A", "score": 7 }, { "date": { "$date": 1334275200000 }, "grade": "A", "score": 12 }, { "date": { "$date": 1318377600000 }, "grade": "A", "score": 12 }], "name": "Riviera Caterer", "restaurant_id": "40356018" }, { "address": { "building": "97-22", "coord": [-73.8601152, 40.7311739], "street": "63 Road", "zipcode": "11374" }, "borough": "Queens", "cuisine": "Jewish Kosher", "grades": [{ "date": { "$date": 1416787200000 }, "grade": "Z", "score": 20 }, { "date": { "$date": 1358380800000 }, "grade": "A", "score": 13 }, { "date": { "$date": 1343865600000 }, "grade": "A", "score": 13 }, { "date": { "$date": 1323907200000 }, "grade": "B", "score": 25 }], "name": "Tov Kosher Kitchen", "restaurant_id": "40356068" }, { "address": { "building": "8825", "coord": [-73.8803827, 40.7643124], "street": "Astoria Boulevard", "zipcode": "11369" }, "borough": "Queens", "cuisine": "American ", "grades": [{ "date": { "$date": 1416009600000 }, "grade": "Z", "score": 38 }, { "date": { "$date": 1398988800000 }, "grade": "A", "score": 10 }, { "date": { "$date": 1362182400000 }, "grade": "A", "score": 7 }, { "date": { "$date": 1328832000000 }, "grade": "A", "score": 13 }], "name": "Brunos On The Boulevard", "restaurant_id": "40356151" }, { "address": { "building": "2206", "coord": [-74.1377286, 40.6119572], "street": "Victory Boulevard", "zipcode": "10314" }, "borough": "Staten Island", "cuisine": "Jewish Kosher", "grades": [{ "date": { "$date": 1412553600000 }, "grade": "A", "score": 9 }, { "date": { "$date": 1400544000000 }, "grade": "A", "score": 12 }, { "date": { "$date": 1365033600000 }, "grade": "A", "score": 12 }, { "date": { "$date": 1327363200000 }, "grade": "A", "score": 9 }], "name": "Kosher Island", "restaurant_id": "40356442" }, { "address": { "building": "7114", "coord": [-73.9068506, 40.6199034], "street": "Avenue U", "zipcode": "11234" }, "borough": "Brooklyn", "cuisine": "Delicatessen", "grades": [{ "date": { "$date": 1401321600000 }, "grade": "A", "score": 10 }, { "date": { "$date": 1389657600000 }, "grade": "A", "score": 10 }, { "date": { "$date": 1375488000000 }, "grade": "A", "score": 8 }, { "date": { "$date": 1342569600000 }, "grade": "A", "score": 10 }, { "date": { "$date": 1331251200000 }, "grade": "A", "score": 13 }, { "date": { "$date": 1318550400000 }, "grade": "A", "score": 9 }], "name": "Wilken'S Fine Food", "restaurant_id": "40356483" }, { "address": { "building": "6409", "coord": [-74.00528899999999, 40.628886], "street": "11 Avenue", "zipcode": "11219" }, "borough": "Brooklyn", "cuisine": "American ", "grades": [{ "date": { "$date": 1405641600000 }, "grade": "A", "score": 12 }, { "date": { "$date": 1375142400000 }, "grade": "A", "score": 12 }, { "date": { "$date": 1360713600000 }, "grade": "A", "score": 11 }, { "date": { "$date": 1345075200000 }, "grade": "A", "score": 2 }, { "date": { "$date": 1313539200000 }, "grade": "A", "score": 11 }], "name": "Regina Caterers", "restaurant_id": "40356649" }, { "address": { "building": "1839", "coord": [-73.9482609, 40.6408271], "street": "Nostrand Avenue", "zipcode": "11226" }, "borough": "Brooklyn", "cuisine": "Ice Cream, Gelato, Yogurt, Ices", "grades": [{ "date": { "$date": 1405296000000 }, "grade": "A", "score": 12 }, { "date": { "$date": 1373414400000 }, "grade": "A", "score": 8 }, { "date": { "$date": 1341964800000 }, "grade": "A", "score": 5 }, { "date": { "$date": 1329955200000 }, "grade": "A", "score": 8 }], "name": "Taste The Tropics Ice Cream", "restaurant_id": "40356731" }, { "address": { "building": "2300", "coord": [-73.8786113, 40.8502883], "street": "Southern Boulevard", "zipcode": "10460" }, "borough": "Bronx", "cuisine": "American ", "grades": [{ "date": { "$date": 1401235200000 }, "grade": "A", "score": 11 }, { "date": { "$date": 1371600000000 }, "grade": "A", "score": 4 }, { "date": { "$date": 1339718400000 }, "grade": "A", "score": 3 }], "name": "Wild Asia", "restaurant_id": "40357217" }, { "address": { "building": "7715", "coord": [-73.9973325, 40.61174889999999], "street": "18 Avenue", "zipcode": "11214" }, "borough": "Brooklyn", "cuisine": "American ", "grades": [{ "date": { "$date": 1397606400000 }, "grade": "A", "score": 5 }, { "date": { "$date": 1366675200000 }, "grade": "A", "score": 2 }, { "date": { "$date": 1335225600000 }, "grade": "A", "score": 5 }, { "date": { "$date": 1323993600000 }, "grade": "A", "score": 2 }], "name": "C & C Catering Service", "restaurant_id": "40357437" }, { "address": { "building": "1269", "coord": [-73.871194, 40.6730975], "street":

[{ "address": { "building": "1007", "coord": [-73.856077, 40.848447], "street": "Morris Park Ave", "zipcode": "10462" }, "borough": "Bronx", "cuisine": "Bakery", "grades": [{ "date": { "$date": 1393804800000 }, "grade": "A", "score": 2 }, { "date": { "$date": 1378857600000 }, "grade": "A", "score": 6 }, { "date": { "$date": 1358985600000 }, "grade": "A", "score": 10 }, { "date": { "$date": 1322006400000 }, "grade": "A", "score": 9 }, { "date": { "$date": 1299715200000 }, "grade": "B", "score": 14 }], "name": "Morris Park Bake Shop", "restaurant_id": "30075445" }, { "address": { "building": "469", "coord": [-73.961704, 40.662942], "street": "Flatbush Avenue", "zipcode": "11225" }, "borough": "Brooklyn", "cuisine": "Hamburgers", "grades": [{ "date": { "$date": 1419897600000 }, "grade": "A", "score": 8 }, { "date": { "$date": 1404172800000 }, "grade": "B", "score": 23 }, { "date": { "$date": 1367280000000 }, "grade": "A", "score": 12 }, { "date": { "$date": 1336435200000 }, "grade": "A", "score": 12 }], "name": "Wendy'S", "restaurant_id": "30112340" }, { "address": { "building": "351", "coord": [-73.98513559999999, 40.7676919], "street": "West 57 Street", "zipcode": "10019" }, "borough": "Manhattan", "cuisine": "Irish", "grades": [{ "date": { "$date": 1409961600000 }, "grade": "A", "score": 2 }, { "date": { "$date": 1374451200000 }, "grade": "A", "score": 11 }, { "date": { "$date": 1343692800000 }, "grade": "A", "score": 12 }, { "date": { "$date": 1325116800000 }, "grade": "A", "score": 12 }], "name": "Dj Reynolds Pub And Restaurant", "restaurant_id": "30191841" }, { "address": { "building": "2780", "coord": [-73.98241999999999, 40.579505], "street": "Stillwell Avenue", "zipcode": "11224" }, "borough": "Brooklyn", "cuisine": "American ", "grades": [{ "date": { "$date": 1402358400000 }, "grade": "A", "score": 5 }, { "date": { "$date": 1370390400000 }, "grade": "A", "score": 7 }, { "date": { "$date": 1334275200000 }, "grade": "A", "score": 12 }, { "date": { "$date": 1318377600000 }, "grade": "A", "score": 12 }], "name": "Riviera Caterer", "restaurant_id": "40356018" }, { "address": { "building": "97-22", "coord": [-73.8601152, 40.7311739], "street": "63 Road", "zipcode": "11374" }, "borough": "Queens", "cuisine": "Jewish Kosher", "grades": [{ "date": { "$date": 1416787200000 }, "grade": "Z", "score": 20 }, { "date": { "$date": 1358380800000 }, "grade": "A", "score": 13 }, { "date": { "$date": 1343865600000 }, "grade": "A", "score": 13 }, { "date": { "$date": 1323907200000 }, "grade": "B", "score": 25 }], "name": "Tov Kosher Kitchen", "restaurant_id": "40356068" }, { "address": { "building": "8825", "coord": [-73.8803827, 40.7643124], "street": "Astoria Boulevard", "zipcode": "11369" }, "borough": "Queens", "cuisine": "American ", "grades": [{ "date": { "$date": 1416009600000 }, "grade": "Z", "score": 38 }, { "date": { "$date": 1398988800000 }, "grade": "A", "score": 10 }, { "date": { "$date": 1362182400000 }, "grade": "A", "score": 7 }, { "date": { "$date": 1328832000000 }, "grade": "A", "score": 13 }], "name": "Brunos On The Boulevard", "restaurant_id": "40356151" }, { "address": { "building": "2206", "coord": [-74.1377286, 40.6119572], "street": "Victory Boulevard", "zipcode": "10314" }, "borough": "Staten Island", "cuisine": "Jewish Kosher", "grades": [{ "date": { "$date": 1412553600000 }, "grade": "A", "score": 9 }, { "date": { "$date": 1400544000000 }, "grade": "A", "score": 12 }, { "date": { "$date": 1365033600000 }, "grade": "A", "score": 12 }, { "date": { "$date": 1327363200000 }, "grade": "A", "score": 9 }], "name": "Kosher Island", "restaurant_id": "40356442" }, { "address": { "building": "7114", "coord": [-73.9068506, 40.6199034], "street": "Avenue U", "zipcode": "11234" }, "borough": "Brooklyn", "cuisine": "Delicatessen", "grades": [{ "date": { "$date": 1401321600000 }, "grade": "A", "score": 10 }, { "date": { "$date": 1389657600000 }, "grade": "A", "score": 10 }, { "date": { "$date": 1375488000000 }, "grade": "A", "score": 8 }, { "date": { "$date": 1342569600000 }, "grade": "A", "score": 10 }, { "date": { "$date": 1331251200000 }, "grade": "A", "score": 13 }, { "date": { "$date": 1318550400000 }, "grade": "A", "score": 9 }], "name": "Wilken'S Fine Food", "restaurant_id": "40356483" }, { "address": { "building": "6409", "coord": [-74.00528899999999, 40.628886], "street": "11 Avenue", "zipcode": "11219" }, "borough": "Brooklyn", "cuisine": "American ", "grades": [{ "date": { "$date": 1405641600000 }, "grade": "A", "score": 12 }, { "date": { "$date": 1375142400000 }, "grade": "A", "score": 12 }, { "date": { "$date": 1360713600000 }, "grade": "A", "score": 11 }, { "date": { "$date": 1345075200000 }, "grade": "A", "score": 2 }, { "date": { "$date": 1313539200000 }, "grade": "A", "score": 11 }], "name": "Regina Caterers", "restaurant_id": "40356649" }, { "address": { "building": "1839", "coord": [-73.9482609, 40.6408271], "street": "Nostrand Avenue", "zipcode": "11226" }, "borough": "Brooklyn", "cuisine": "Ice Cream, Gelato, Yogurt, Ices", "grades": [{ "date": { "$date": 1405296000000 }, "grade": "A", "score": 12 }, { "date": { "$date": 1373414400000 }, "grade": "A", "score": 8 }, { "date": { "$date": 1341964800000 }, "grade": "A", "score": 5 }, { "date": { "$date": 1329955200000 }, "grade": "A", "score": 8 }], "name": "Taste The Tropics Ice Cream", "restaurant_id": "40356731" }, { "address": { "building": "2300", "coord": [-73.8786113, 40.8502883], "street": "Southern Boulevard", "zipcode": "10460" }, "borough": "Bronx", "cuisine": "American ", "grades": [{ "date": { "$date": 1401235200000 }, "grade": "A", "score": 11 }, { "date": { "$date": 1371600000000 }, "grade": "A", "score": 4 }, { "date": { "$date": 1339718400000 }, "grade": "A", "score": 3 }], "name": "Wild Asia", "restaurant_id": "40357217" }, { "address": { "building": "7715", "coord": [-73.9973325, 40.61174889999999], "street": "18 Avenue", "zipcode": "11214" }, "borough": "Brooklyn", "cuisine": "American ", "grades": [{ "date": { "$date": 1397606400000 }, "grade": "A", "score": 5 }, { "date": { "$date": 1366675200000 }, "grade": "A", "score": 2 }, { "date": { "$date": 1335225600000 }, "grade": "A", "score": 5 }, { "date": { "$date": 1323993600000 }, "grade": "A", "score": 2 }], "name": "C & C Catering Service", "restaurant_id": "40357437" }, { "address": { "building": "1269", "coord": [-73.871194, 40.6730975], "street":

6- Créer un job Jenkins





=>Afin de garantir que tous les conteneurs partagent le même réseau avec la commande :

"docker inspect bis_network"

Command Prompt

```
C:\Users\amenm\OneDrive\Bureau\integration>docker inspect bis_network
[
    {
        "Name": "bis_network",
        "Id": "1b3d6affb87169026686f48e27f5153c84224d1fb75810d29d3a21c55017746c",
        "Created": "2023-12-08T19:11:29.425255109Z",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": {},
            "Config": [
                {
                    "Subnet": "172.18.0.0/16",
                    "Gateway": "172.18.0.1"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": {
            "1a6526cb99d0741ea459bf2cbe8b934efdf15d3b79e39b4393a5937b3f17be48": {
                "Name": "nginx1",
                "EndpointID": "3d81292900ac03b84993c4e1a0395e0451bcff0c440b2a9cce7fae4f5cee2559",
                "MacAddress": "02:42:ac:12:00:02",
                "IPv4Address": "172.18.0.2/16",
                "IPv6Address": ""
            },
            "ea01e8564aec40b6cc390b97a4722fc7f61cfa085c8007e180b560ef190c91da": {
                "Name": "nginx2",
                "EndpointID": "186469df0358d8bc9e3d2b8000113735d3e402033fa1351064b34fd652c256e9",
                "MacAddress": "02:42:ac:12:00:03",
                "IPv4Address": "172.18.0.3/16",
                "IPv6Address": ""
            }
```

```
                "IPv6Address": ""
            }
        },
        "Options": {},
        "Labels": {}
    }
]

C:\Users\amenm\OneDrive\Bureau\integration>
```