Bonjour, voici mon devoir maison.

N’ayant finalement eu que ce Week-end pour réaliser l’exercice, du fait d’une charge particulièrement importante ces dix derniers jours (période de livraison), je n’ai pas pu réaliser tout ce que je voulais.

Exit donc la dockerisation de l’ensemble et le scaling par Kubernetes que je visais, ainsi que la mise en place de la couche de sécurité.

Je vais néanmoins le mettre en place cette semaine.

Le projet est livré avec les scripts batch de lancement et les jars correspondants [ici](https://1drv.ms/u/s!AjSiPVqJt2RXn9M5RH2eG2gQ-fl1QQ?e=KUlLbs), sur un partage OneDrive. Le code est [ici](https://1drv.ms/u/s!AjSiPVqJt2RXn9M6yP9SQ5McC4QsKA?e=0eg47L).

J’envoie en parallèle, un autre partage sur Dropbox cette fois-ci.

Le libellé de la réalisation demandée précisait que l’api serait accédée par beaucoup d’applications en simultané.

J’ai opté pour une architecture en microservices basée sur Euréka, qui permet une répartition de la charge entre les différentes instances présentes, ainsi que la détection de leur potentielle défection.

Ici, la fonction List<List<String>> partition(List<String> lst, **int** nbre ) sera accédée en ReST, par l’URL de forme suivante :

Examples :

* Accès par le Load-balancer :

http://localhost:5555/**array**?tablo=1,2,3,4,5,6,7,8&nombre=3

* Accès direct :

http://localhost:800x/**array**?tablo=1,2,3,4,5,6,7,8&nombre=3

Pour l’exercice et la résolution DNS, j’ai ajouté à c:\Windows\System32\drivers\etc\hosts, les lignes suivantes :

127.0.0.1 eureka.svr.com

127.0.0.1 loc1.eureka.srv.com

127.0.0.1 loc2.eureka.srv.com

127.0.0.1 loc3.eureka.srv.com

Recompilation

Pour vérifier les tests, importer dans l’IDE à partir du .rar fourni « eurekaregisterserver », « eurekadiscoveryclient » et « eurekaribbon ».

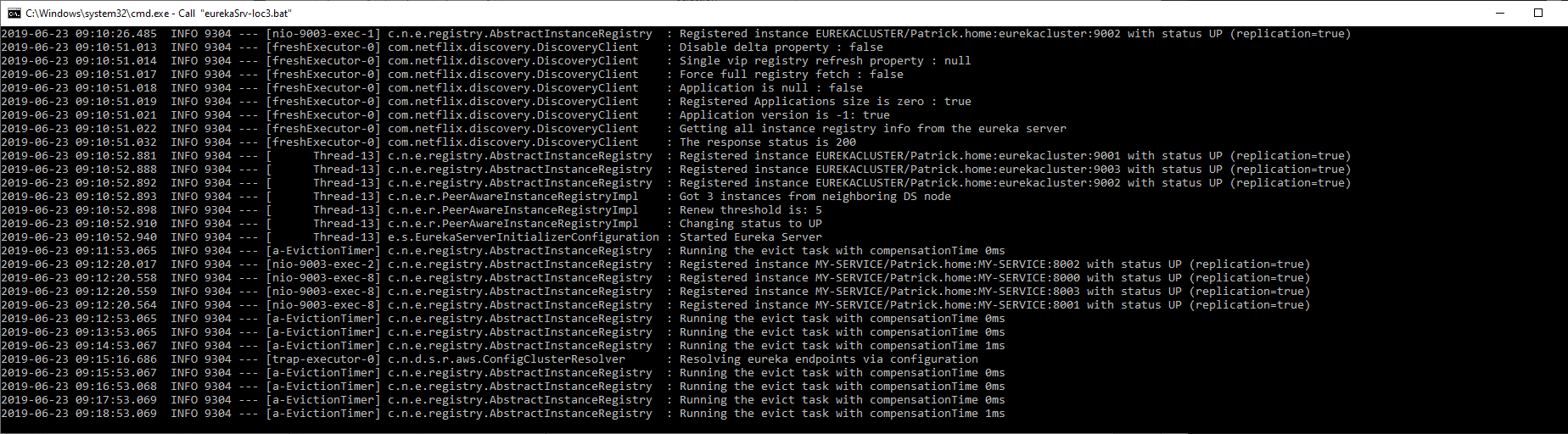
Classiquement, faire click-droit sur chacun d’entre eux/ « run as »/ « maven install ».

Recopier chacun des jars situés dans \target de chacun des projets dans le répertoire des scripts batch de lancement fournis.

Lancement du projet :

Exécuter « \_1-All-eurekaSrv.bat », l’instance de profil « default » est lancée en premier, et 20 secondes plus tard, les autres instances le sont à leur tour. Après moins d’une minute, les serveurs se sont mutuellement enregistrés :

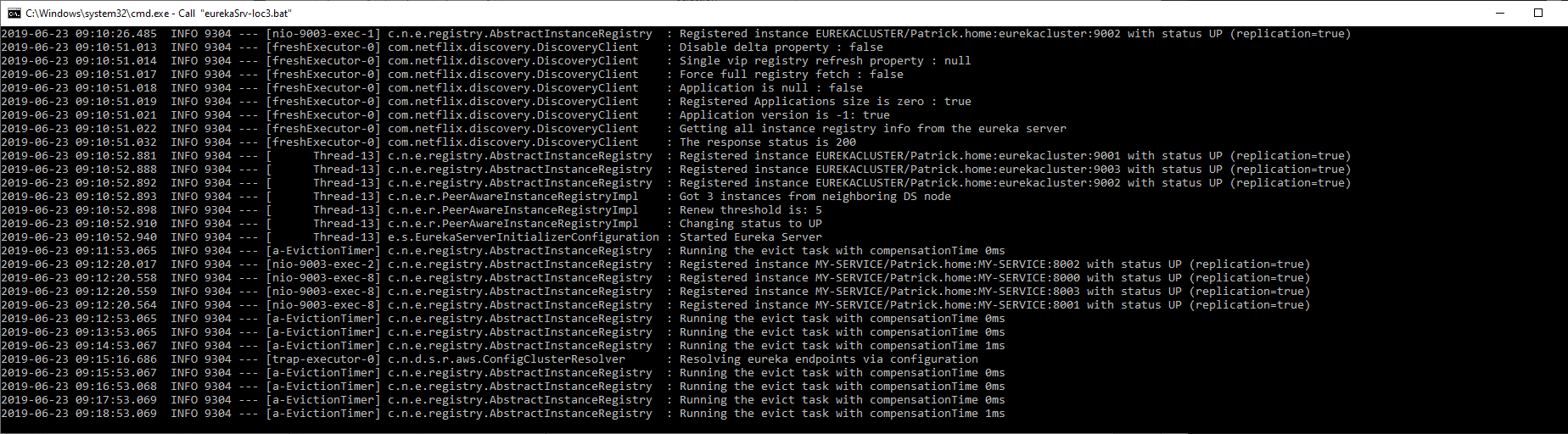
Vérification pour l’une des instances:



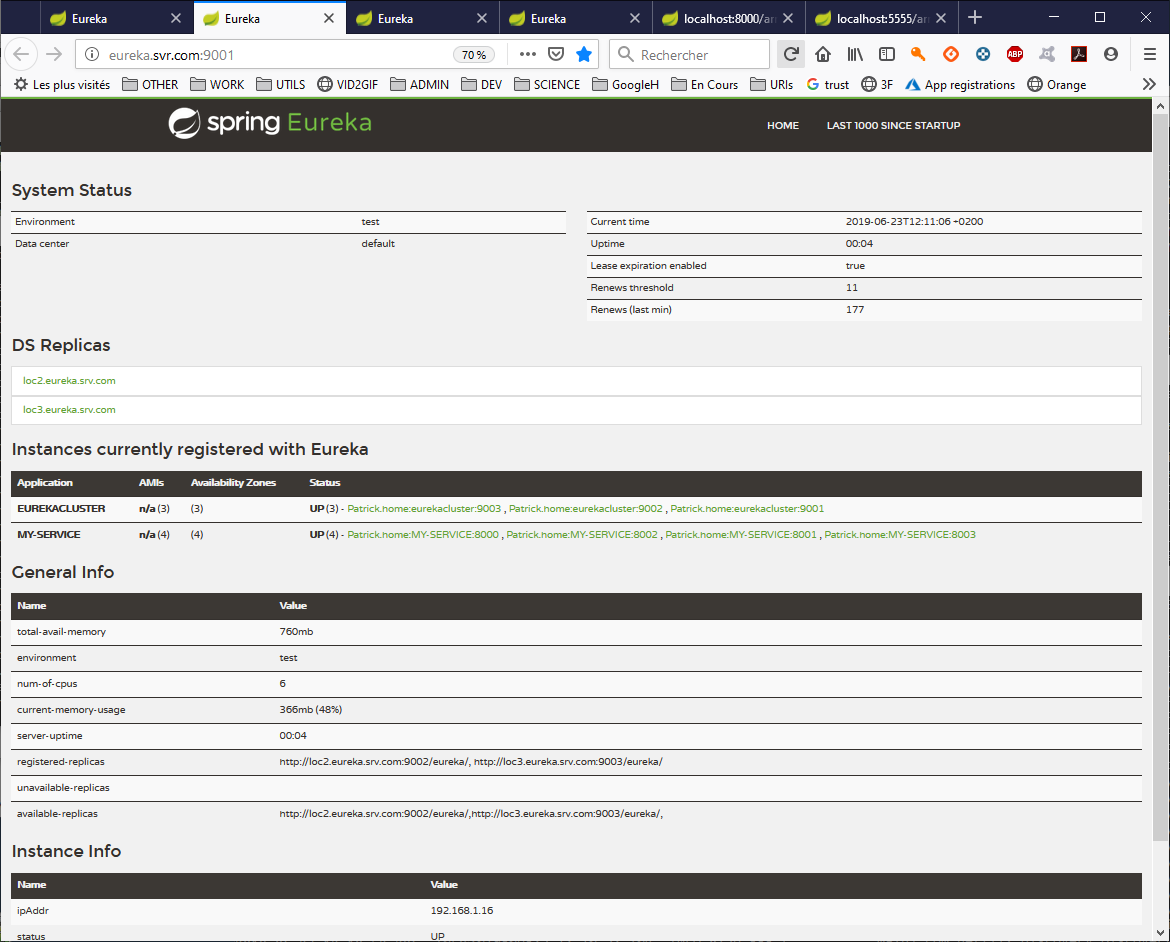
Vérification: attendre que « EUREKACLUSTER » ait bien ses 3 membres :

|  |  |
| --- | --- |
| Server Default | Instance 9001 |
|  |  |
| Instance 9002 | Instance 9003 |
|  |  |

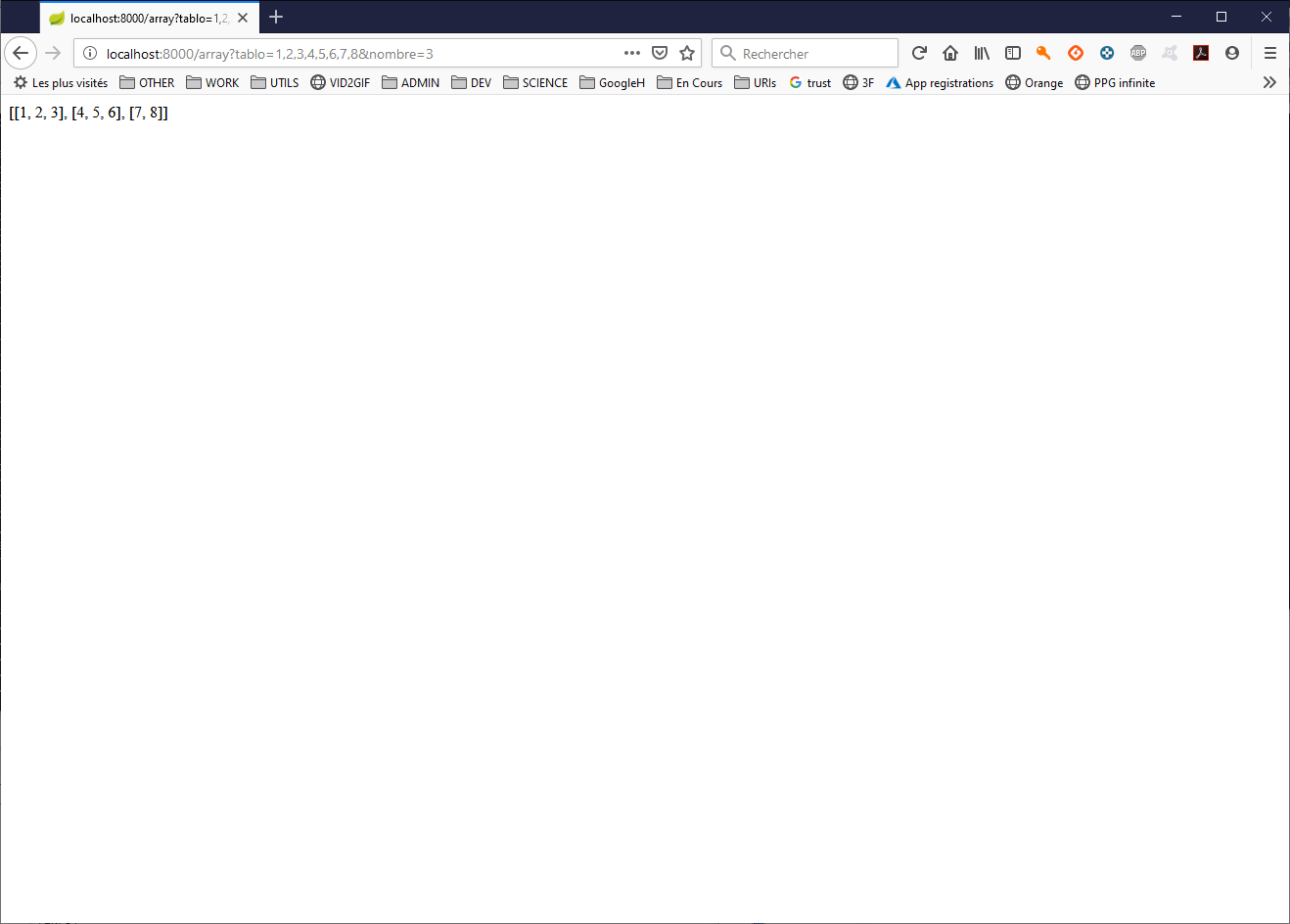
Une fois que les instances se voient, lancer les instances DiscoveryClient (ici 3 instances, sur 5 possibles) : exécuter « \_2-All-eurekaDisco.bat »



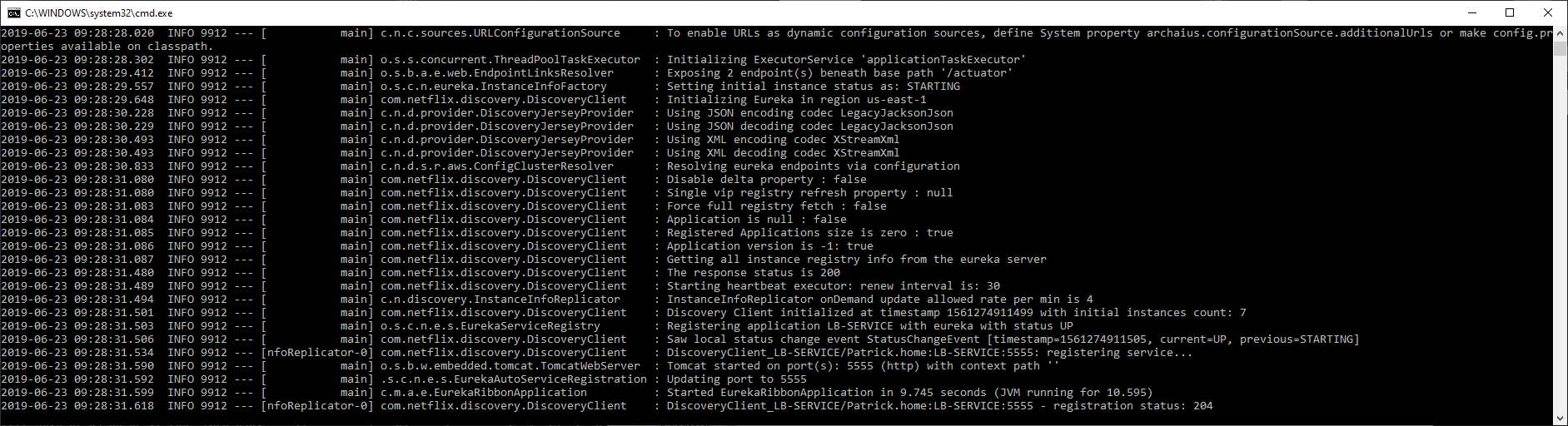
Vérifier que toutes les instances sont bien registered :



Après moins d’une minute, test d’une des instances en direct \*:

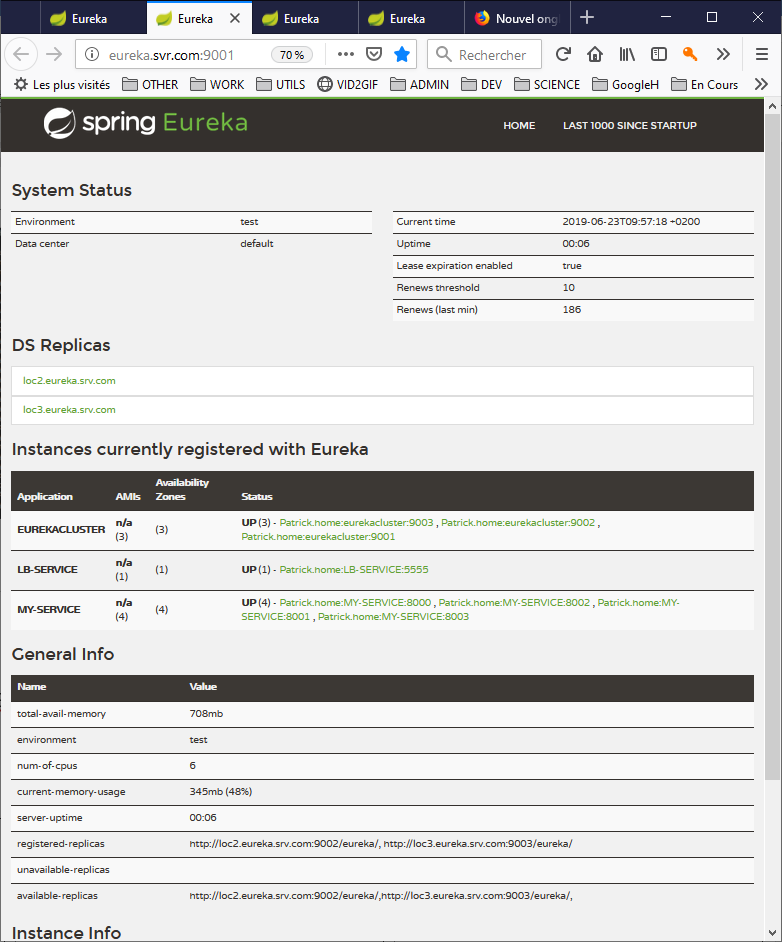


Lancement du Ribbon : exécuter « \_3-eurekaRibbon.bat»

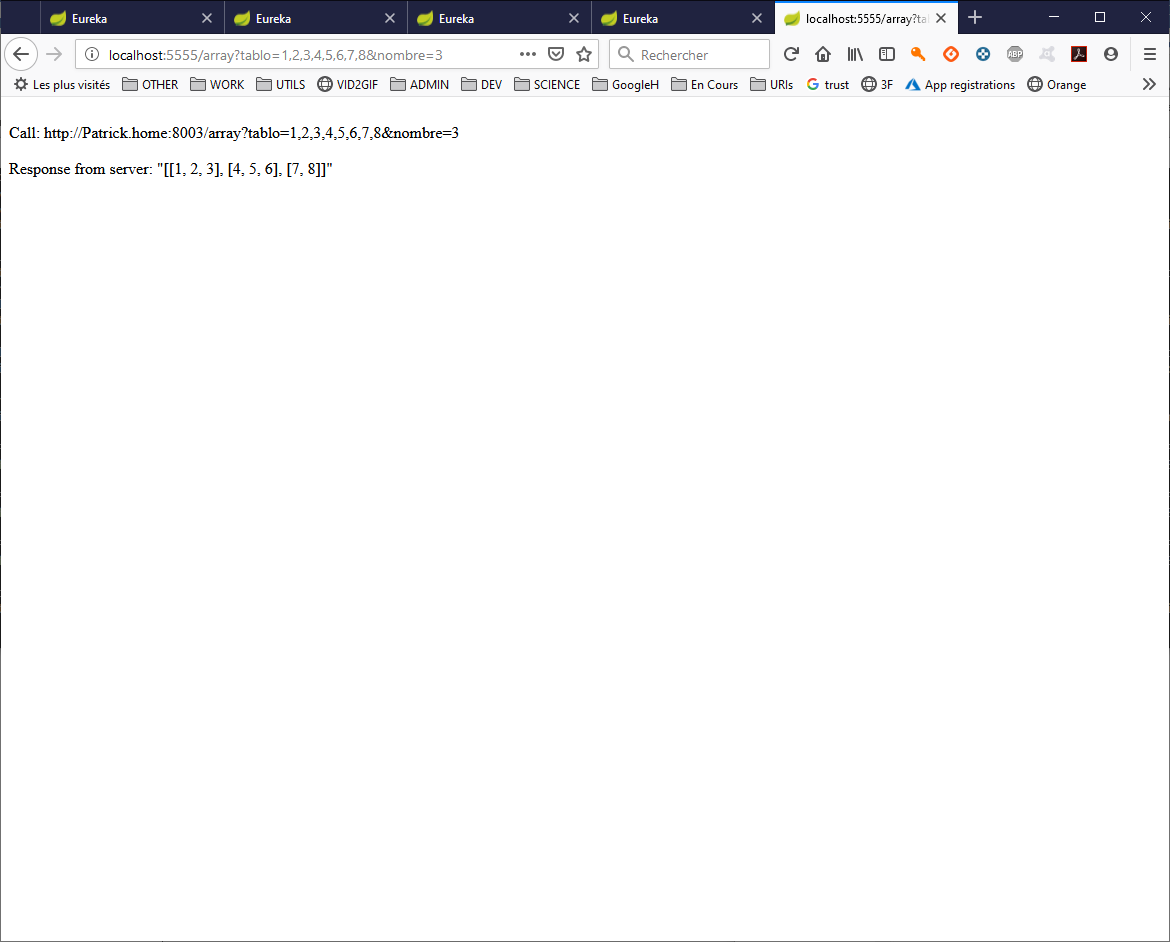


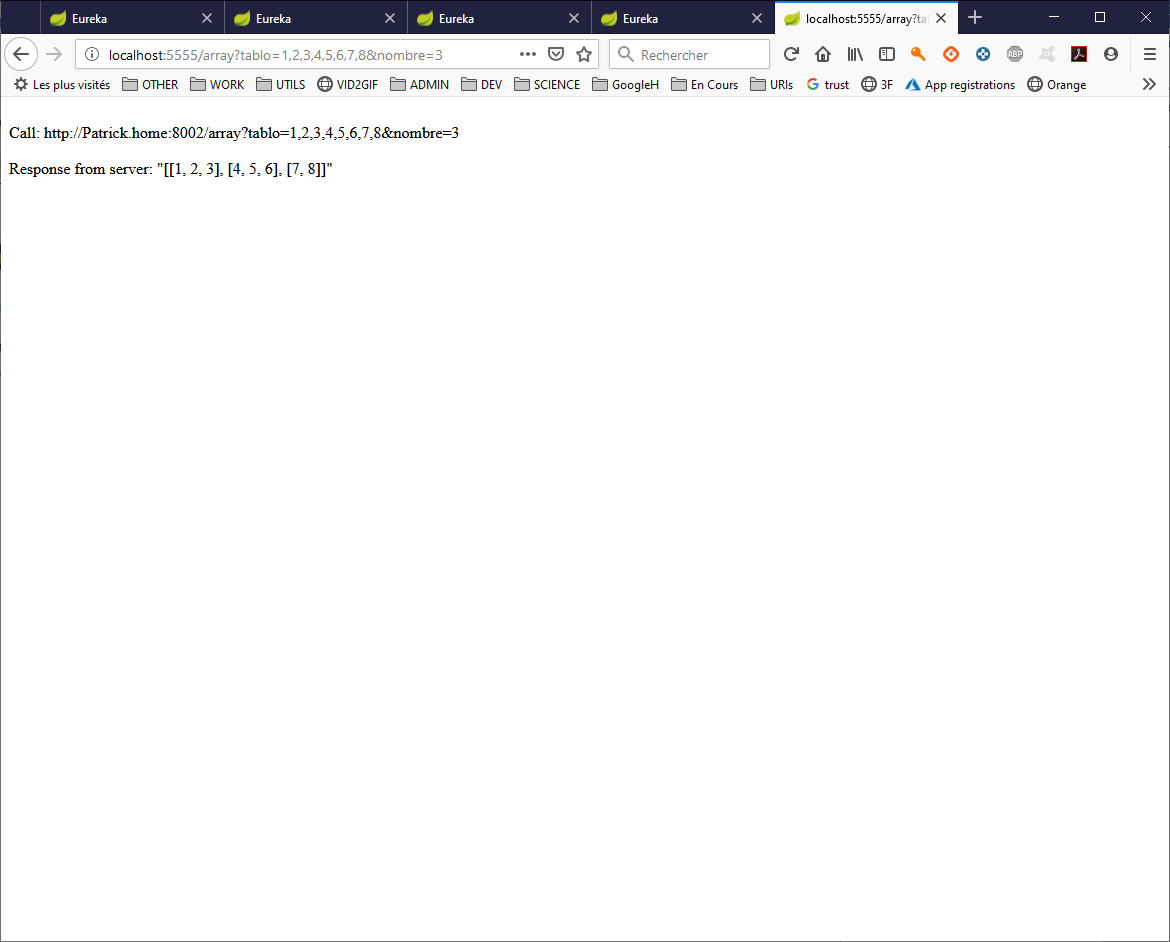
Verification pour une des instances en console:

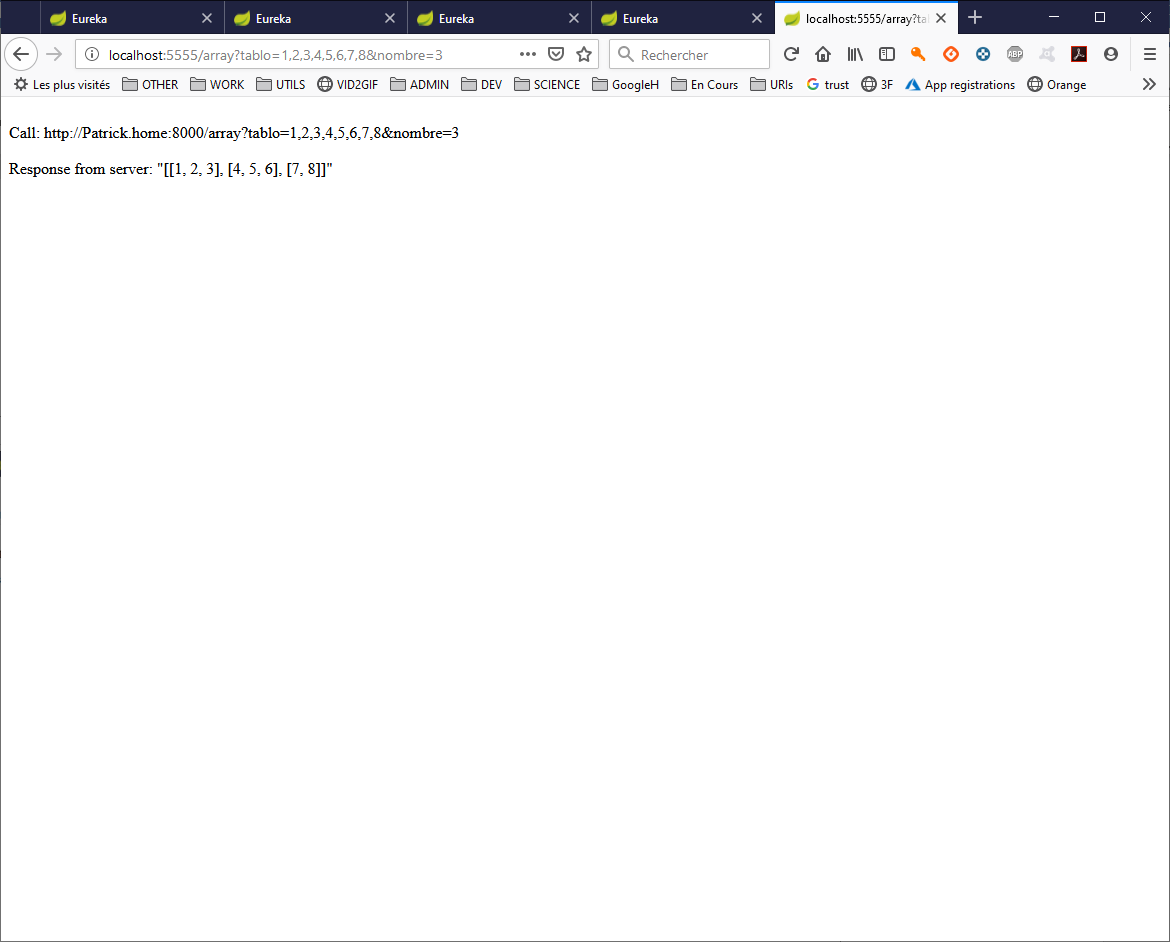
Idem :



Test de la même uri \*, mais en passant par Ribbon, et verification du loadBalancing en rechargeant la page plusieurs fois:







La fenetre console renvoit les statistiques :

