STACK LABS

Pascal Gillet <pascal.gillet@stack-labs.com>

[hippi-spark-k8s] Feedback - Suivi de mission

Pascal Gillet <pascal.gillet@stack-labs.com> 30 novembre 2020 à 17:13
À : Florence BRARD <FBRARD@sii.fr>, David Desviel <david.desviel@stack-labs.com>, Xavier Villalba <xavier.villalba@stack-labs.com>, Pascal ROUANET <PROUANET@sii.fr>, "QUINCHARD, Eric" <eric.quinchard@airbus.com>, "DELBENDE, Marc" <marc.delbende@airbus.com>, "ROUGET, Matthieu" <matthieu.rouget@airbus.com>

Bonjour à Tous,

Voici le compte-rendu de la réunion de ce matin:

État d'avancement: **Nominal**

- **Spark configuration:**
Toute la configuration Spark est maintenant centralisée dans la ConfigMap.

Il reste maintenant à injecter/surcharger dans la conf les paramètres qui seront fournis au runtime depuis le spark_client, en suivant le fonctionnement actuel (~ MesosSparkJobConf), notamment:
- la priorité
- l'UID du job spark
- les node affinities
- l'image Docker
- le nombre d'executors, de cores, quantité de mémoire
- Autres propriétés Spark...

Concernant la configuration statique, et pour coller au maximum à la philosophie Kubernetes ("template-free"), un maximum de paramètres seront codés en dur, notamment:
- le namespace ("spark-jobs")
- le nom du compte de service pour les pods driver ("hippi-spark")
- le nom racine des applications Spark sera toujours le même: "spark"

Une fois instanciés au runtime, les fichiers YAML pourront être exportés dans un répertoire de travail à l'aide d'une méthode utilitaire dans le spark_client.

- Ingress: Traefik vs Nginx
La Spark UI est disponible via une Ingress Nginx.

Ma PR qui fixe un petit problème de Javascript dans la Spark UI quand elle est accédée derrière un reverse proxy: https://github.com/apache/spark/pull/30523

J'ai essayé Traefik, voici mon retour d'expérience (mais ça n'est que mon opinion):
- Nginx est plus simple à installer et à configurer que Traefik: je n'ai pas réussi à configurer Traefik pour la Spark UI (pb du redirect HTTP).
- La doc Nginx est mieux fournie et il y a plus de support de la communauté.

Pour l'instant, je reste sur Nginx et je verrai à configurer Traefik s'il me reste du temps (**[MDE]** Marc, je suis intéressé si tu as déjà une Ingress Traefik qui marche).
On peut installer différents ingress controllers dans un cluster K8s. Si une ingress ne spécifie pas de controller avec l'annotation kubernetes.io/ingress.class, les controllers entrent en compétition pour gérer cette ingress.

[MDE] Marc, tu peux installer le controller Nginx en suivant le guide d'installation (section bare-metal): https://kubernetes.github.io/ingress-nginx/deploy/#bare-metal
La commande suivante devrait suffire (?):
`kubectl apply -f` https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v0.41.2/deploy/static/provider/baremetal/deploy.yaml

- **spark_client: refactoring de l'existant**
J'ai refactoré le spark_client de façon à préparer le support de K8s.
Les changements sont tracés ici: https://gitlab.com/stack-labs/client/airbus-defence-and-space/hippi-spark-k8s/spark_client_python_k8s/-/blob/master/spark_client/CHANGES.TXT

- spark_client: ajout du spark-submit K8s natif

Le support du spark-submit K8s natif est pratiquement terminé, modulo l'injection de la configuration (voir 1er point). Pour le Spark Operator, tout le code sera commun avec la méthode native excepté la méthode `run()`.

- spark_client: façade
L'idée est de fournir une interface unifiée à l'ensemble des implémentations de l'interface `SparkJobRunner` (Mesos, Local, K8s natif, K8s Spark Operator). Cette façade définit donc une interface de niveau supérieur qui facilitera l'utilisation du spark_client, avec notamment un paramètre de conf pour choisir la cible Mesos ou K8s.

- spark_client: logs method
Une nouvelle méthode `logs()` sera ajoutée à l'interface SparkJobRunner pour récupérer les logs du driver Spark. Les logs seront "re-loggés" avec le logger Python passé en paramètre.

[MDE] Marc, j'ai posé un tag dans le repo hippi-spark-k8s pour téléchargement: https://gitlab.com/stack-labs/client/airbus-defence-and-space/hippi-spark-k8s/hippi-spark-k8s/-/tags/20203011

--
STACK LABS
Pascal GILLET | Big Data & Cloud Architect
stack-labs.com
21 Boulevard de la Marquette, 31000 Toulouse
France · Canada