Yannis GOVINDA – yannis.govinda@iphc.cnrs.fr
Jérôme PANSANEL – jerome.pansanel@iphc.cnrs.fr
Rémi CAILLETAUD – remi.cailletaud@univ-grenoble-alpes.fr

Mise en place de Clusters GPU à la demande pour du calcul de données

Sommaire

Sommaire

1. Objectif – Automatisation des déploiements applicatifs GPUs
2. Outils et technologies
3. Les Clusters GPU
4. Vidéo
5. Conclusion

Objectif

Automatiser la chaine de mise à disposition des ressources GPU via des conteneurs (notebook Jupyter) dans un contexte multi-cloud.

Nous allons déployer un JupyterHub, avec l'accès au notebook tensorflow-notebook par le biais de Terraform, RKE2 et Helm. Les étapes présentées sont :

- Récupération du code
- Choix des gabarits et images
- Lancement du cluster
- Accès au JupyterHub et lancement du notebook

Outils et technologies

Pré-requis :
- Terraform 0.13+

Distribution Kubernetes :
- RKE2

Module Terraform :
- remche/rke2/openstack

https://registry.terraform.io/modules/remche/rke2/openstack/latest

Kubernetes applications manager :
- Helm

GPU Driver Manager :
- GPU Operator

Les clusters GPUs

Master :
- Image : Ubuntu 20.04
- Gabarit : m1.large (4 VCPU, 8Go Ram, 20Go stockage), m1.large-2d (40Go stockage)

Worker :
- Image : Ubuntu 20.04
- Gabarit : g1.xlarge-4mem (Tesla P100), g2.xlarge-4mem (RTX 2080TI), g4.xlarge-4mem (Tesla T4) // (tous les gabarits GPU ont 8 VCPU, 64Go Ram)

Charts Helm
- gpu-operator : prépare les machines et le cluster pour des podsGPU
- Zero-to-jupyterhub : Configuration avec réservation d'une ressource GPU

Vidéo

Conclusion

Ergonomie : L'outil ici utilisé n'est pas facilement utilisable par des utilisateurs non avertis. C'est pourquoi nous sommes actuellement entrain d'étudier des solutions permettant de déployer des logiciels à la demande (ex : Rancher, KubeApps, …)

Réutilisabilité : L'outil ici fourni est réutilisable sur n'importe quelle plateforme Cloud Openstack. Le code pourra alors être adapté si besoin est à cette configuration.