

Centre de Calcul de l'Institut National de Physique Nucléaire et de Physique des Particules

Singularity au CC IN2P3

31/05/2018



- ▶ Rapide présentation du CC et du contexte.
- ▶ Pourquoi la containerisation ?
- ▶ Singularity.
 - Le produit.
 - La question du binding.
 - Les difficultés rencontrées/existantes.
 - Les usages au CC.
- ▶ Enseignements et futur.

- ▶ Centre de calcul de l'IN2P3.
- ▶ Missions
 - Infrastructure de stockage de masse et de traitement de données.
 - Infrastructure réseau IN2P3.
 - Services communs institutionnels (messagerie collaborative, gestion électronique de documents, entrepôt de développement logiciel, gestion de projets etc.).
- ▶ Personnel
 - 84 agents (ingénieurs, techniciens, administratifs et chercheur).
 - 11 équipes dont 8 techniques sur les aspects IT.
- ▶ Le centre fournit des moyens de calcul et stockage pour ~90 expériences (HEP, Astroparticule, Cosmologie, Biologie,.....).



- ▶ 1700 m2 sur deux salles de calcul.
- ▶ Connectivité direct sur Renater.
- ▶ Moyens de calcul.
 - Ferme de calcul de type HTC (+16 000 cores), une ferme HPC (512 cores + Infiniband), Un ferme GPU (160 cores et 40 GPU).
- ▶ Moyens de stockage sur disques.
 - Diverses technologies de stockage.
 - DCACHE (13,8PB), XROOTD (6,4PB), IRODS (1,8PB), SPS/GPFS (2,6PB).
- ▶ Moyens de stockage sur bande.
 - 33 PB utilisés sur 343PB de disponibles.
- ▶ Des services IT.
 - DB, forges, gestionnaire de code,.....



- ▶ La containerisation des services.
 - Via Docker et Kubernetes.
 - Uniquement pour des services.
 - Pour des usages internes au centre mais aussi pour des services ouverts à l'extérieur.
 - Cluster web, outils de GED, outils de développement, outil de gestion divers,...
- ▶ Offrir la containerisation aux utilisateurs.
 - A travers nos fermes de calcul.
 - Une de nos plus importante (en termes de consommation de ressources) expériences exige la présence de singularity pour tourner sur nos infrastructures.

Pourquoi containeriser les workflow (notamment avec singularity) ?

- ▶ Du point de vue des expériences/applications : Ne pas avoir à porter ses codes.
 - Singularity (la containerisation de façon générale) n'introduit pas de pénalités dans l'efficacité des jobs. 😊
 - Sur le papier un applicatif qui tourne sur un OS doit tourner dans un container du même OS sans aucun changement. 😊
 - Reproductibilité facile
- ▶ Du point de vue des sites : Pouvoir déployer l'OS que l'on souhaite sur son matériel indépendamment des requirements applicatifs.
 - Singularity (la containerisation de façon générale) n'introduit pas de pénalités dans l'efficacité des jobs. 😊
 - Singularity ne nécessite pas de déploiement de service . 😊
- ▶ Permettre une isolation forte de l'applicatif.
- ▶ Utiliser des containers doit permettre de s'adresser à un panel d'infrastructures plus large (HPC, Cloud,...).

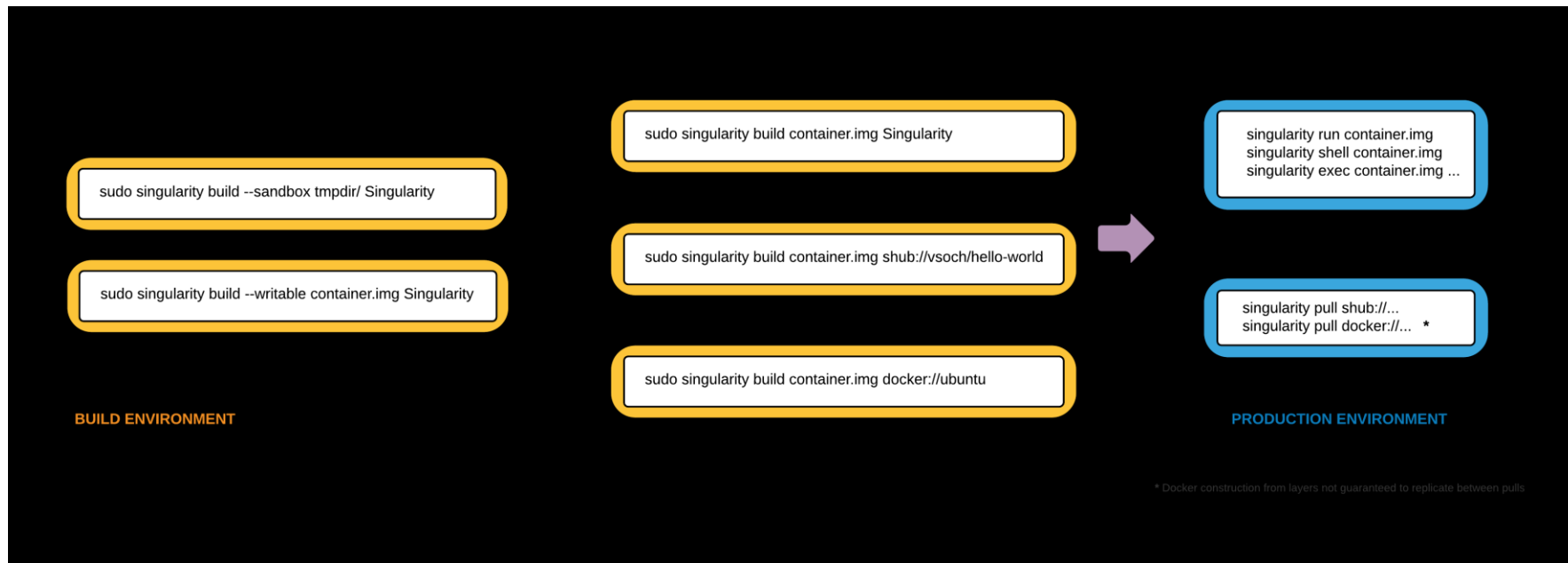
- ▶ Au niveau des utilisateurs singularity permet :
 - De leurs laisser le plein contrôle de leur environnement.
 - De gérer leurs workflows, leurs piles logicielles et librairies.
 - De gérer leurs données.
 - D'utiliser les ressources des plateformes sous-jacentes.
 - Interconnect HPC.
 - File systèmes.
 - GPU/Accélérateurs divers.
 - Etc.
- ▶ Singularity réalise cela :
 - En encapsulant l'environnement d'exécution.
 - En se basant sur des images.
 - En ne modifiant rien au niveau de l'environnement exécution de l'utilisateur.
 - En interdisant toute possibilité de gain de privilèges.
 - En ne nécessitant pas de processus daemon.



- ▶ <http://singularity.lbl.gov/>.
- ▶ Produit issu de la communauté académique.
- ▶ Initialement crée pour être utiliser sur des plateformes HPC.
- ▶ Capable d'utiliser des images docker pour instancier les containers.
 - <https://hub.docker.com/>
 - <https://singularity-hub.org/>
- ▶ La notion de bootstrap existe sous singularity.
- ▶ On trouve dans singularity deux grands sous ensemble.
 - Container Runtime .
 - Container Image Generator.

CONTAINER IMAGE CREATOR

RUNTIME

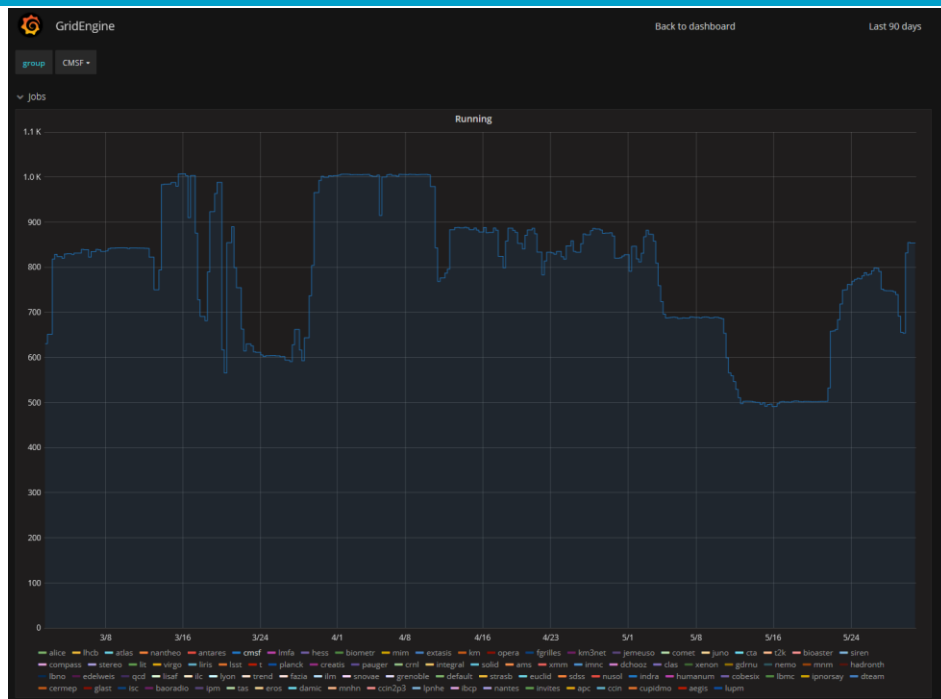


Les images peuvent être de différents formats :
Squashf, ext3, tar.gz, tar, tar.bz2



- ▶ Singularity permet de faire un mapping entre un point de montage de la machine hôte et un répertoire dans le container. Pour faire cela il y a deux approches :
 - Le mapping du point de montage est défini dans l'image du container elle-même et donc seul l'administrateur peut le faire.
 - L'utilisateur peut lui même définir depuis le container des points de montages correspondants à des espaces de la machine hôte. Cela nécessite que des droits particuliers lui aient été donnés.
- ▶ Selon la complexité des workflows applicatifs ces points sont sources d'importantes difficultés.
 - Nécessite de configurer singularity dans un mode dit setUID ou bien de s'appuyer sur des fonctionnalités kernel (namespace) qui ne sont pas encore présentes dans le kernel fournit par défaut.

- ▶ Singularity est disponible depuis quelques mois sur l'ensemble des nœuds de calcul de nos fermes.
 - Version 2.5.
 - Déployé dans le mode setUID.
- ▶ Nous proposons aux utilisateurs un ensemble d'images "de bases".
 - Debian.
 - SL6.
 - CentOS7.
 - Ubuntu.
- ▶ Certaines applications/expériences qui sont organisées au niveau mondial proposent leurs propres images via le système CVMFS.
 - C'est notamment le cas de l'expérience de physiques des hautes énergies qui exige singularity (CMS).
 - Cette expérience exécute quotidiennement des centaines de tâches avec singularity.



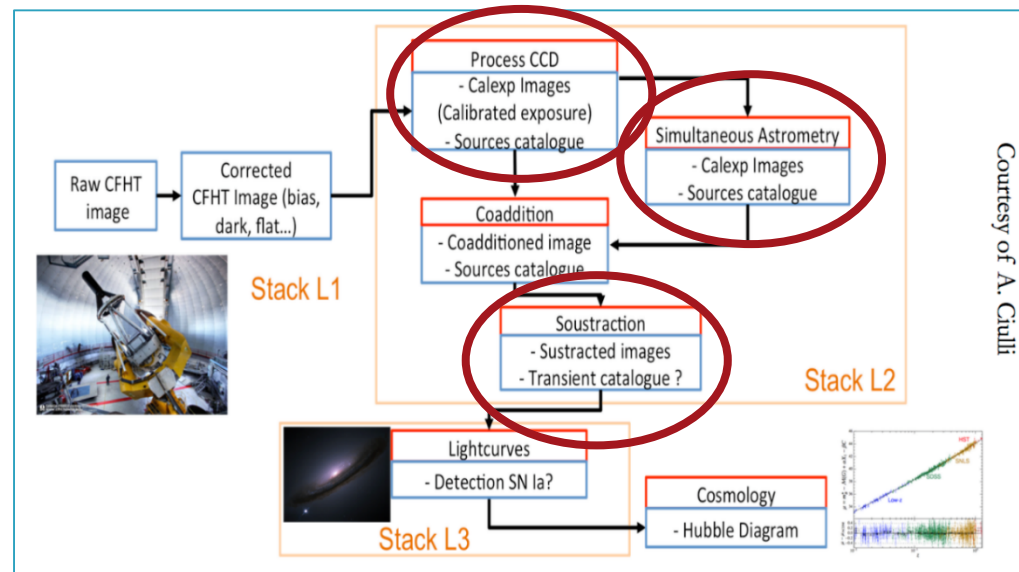
- ▶ Activité au CC de l'expériences CMS, utilisant singularity.
- ▶ Test par l'expérience CMS de la disponibilité de singularity
 - Ce test est jugé critique du côté applicatif.

Detailed output of Metric Result

Field	Value
Hostname	cccrcamceli04.in2p3.fr
Metric	org.cms.WN-isolation
VOFQAN	/cms/Role=pilot
Service Flavour	CREAM-CE
Timestamp	2018-05-30T14:26:02Z
Status	OK
Summary	ccwsgel102: OK: OK
Details	<pre>/usr/bin/singularity exec --home /scratch/43018761.1.mc_long/tmp.YVSMhODsp4:/srv --bind /cvmfs --pwd /srv --contain --ipc --pid /cvmfs/singularity.opensciencegrid.org/bbockelm/cms:rhel6 echo Hello World OR grep Hello World WARNING: Container does not have an exec helper script, calling 'echo' directly Hello World</pre>

- ▶ Singularity est également utiliser au CC pour tester valider des workflows applicatifs ou bien réaliser des challenges.
- ▶ LSST – Deep Learning Challenge
 - Ici c'est l'aspect maitrise de l'environnement d'exécution qu'a permis singularity.
 - Les frameworks nécessaires au code machines learning sont très dépendants des librairies GPU. Singularity à permit de décorréler les aspects implémentation des librairies GPU des applicatifs (tensor flow)

- ▶ LSST : validation du workflow et impact sur les performance de stockage.



- ▶ Un outil intéressant.
- ▶ Potentiellement il simplifie les aspects opérationnels.
- ▶ De plus en plus exploitable en production mais
 - Evolue vite.
 - Encore des bugs.
- ▶ Nécessite parfois tout de même un travail au niveau applicatifs pour être exploité.
 - Les difficultés avec le binding de certains espaces de stockage.
 - Certaines limitations dans le nommage de variables d'environnement.
- ▶ Des formats d'images variés (c'est un bon point) mais qui ont des comportements différents.
 - La façon dont les images sont distribuées peut avoir un impact sur le format de l'image.

- ▶ Des fonctionnalités de singularity ne sont couvertes que par des kernel (ou option de kernel) particulier.
- ▶ Pas de compatibilité assurée entre les différentes versions.
- ▶ Changement ou apport de fonctionnalité d'une version à l'autre.
- ▶ Des éléments doivent encore être clarifiés.
- ▶ Avenir du produit ?
 - Émancipation de Singularity et création de SingularityWare LLC (société à but commercial).