CSAN, a Comprehensive Software Archive Network

Richard Randriatoamanana
Laboratoire LS2N UMR 6004 - CNRS
1 Rue de la Noë
44321 Nantes

Martin Souchal
Laboratoire APC - CNRS
10 rue Alice Domont et Léonie Duquet
75013 Paris

Rémy Dernat
Laboratoire ISEM UMR 5554 – CNRS
Campus Triolet, Bat22, Université de Montpellier
Place E. Bataillon
34095 Montpellier

Alexandre Dehne Garcia
Direction pour la Science Ouverte - INRAE
UMR MISTEA, 2 place Pierre Viala
34000 Montpellier

Jérôme Pansanel
Institut Pluridisciplinaire Hubert Curien
23, rue du Loess – BP28
67037 Strasbourg Cedex 2

Tovo Rabemanantsoa
Direction pour la Science Ouverte - INRAE
UMR ISPA, 71 avenue Édouard Bourlaux
33140 Villenave d'Ornon

Pavel Zakharov
Laboratoire APC - CNRS
10 rue Alice Domont et Léonie Duquet
75013 Paris

Résumé

L'initiative du projet CSAN (Comprehensive Software Archive Network) est de proposer à la communauté ESRI un catalogue d'applications scientifiques open-source prêtes à l'emploi et optimisées pour les centres de calcul nationaux, les mésocentres et les infrastructures distribuées de calcul. Cette proposition part du constat que les applications scientifiques dans les différents mésocentres sont souvent les mêmes selon les secteurs (bio-informatique, physique, etc). La plateforme CSAN permettra aux auteurs de logiciels d'y déposer les versions de leurs sources (ou de pointer vers des dépôts versionnés). Ensuite, le groupe d'experts de CSAN traitera ce code et analysera sa configuration de compilation et de déploiement, via des méthodes d'intégration et de

déploiement continu, afin de le rendre accessible et installable sans effort sur différents systèmes d'exploitation.

Mots-clefs

archive, conteneur, dépôt, hub, versioning, mésocentre, intégration, vérification, reproductibilité, OpenData

1 Introduction

Les conteneurs sont, depuis quelques années, devenus une nouvelle manière de distribuer des logiciels dans un environnement système spécifique. L'intérêt est la facilité avec laquelle il est possible de produire un environnement logiciel complexe puis de le réutiliser ailleurs.

De nombreuses critiques fondées sont apparues sur cette nouvelle méthode de distribution. En effet, le nombre de vulnérabilités a augmenté surtout dans les strates du conteneur, contrairement à un package classique. Par ailleurs, l'environnement de construction et d'exécution du conteneur est déjà une source de vulnérabilité intrinsèquement augmentant ainsi la surface d'attaque.

La communauté de l'enseignement supérieur, de la Recherche et de l'innovation (ESRI) dispose, aujourd'hui, de nombreuses ressources de calcul (centres informatiques des établissements, mésocentres, datacentres régionaux, centres de calcul nationaux) et d'entrepôts de données, répartis dans plusieurs centres nationaux et régionaux, des laboratoires, ou bien encore des plateformes type grille et cloud. L'idée d'utiliser les conteneurs dans ces environnements hétérogènes a fait son chemin, de part leur portabilité, mais la plupart de ces centres sont encore en réflexion pour une généralisation de l'utilisation des conteneurs à la demande.

Dans la chaîne de distribution des conteneurs, un point central concerne le « hub ». En l'occurrence la plateforme docker-hub[1] permet à n'importe qui de déposer une image et de la publier à l'ensemble de la communauté. Une personne expérimentée préférera toujours choisir une image dite « certifiée », offrant plus de garanties en matière de qualité (sécurité, stabilité, à jour, ...), car souvent construite par un éditeur. Sur un autre type de hub, quay.io[2], une analyse des images est régulièrement effectué, permettant de chercher des vulnérabilités (*Common Vulnerabilities and Exposures* - CVE) dans des bases de données avec un rapport qui permet de voir les risques (ou pas) de déployer les images scannées. Côté calcul, singularity-hub[3], qui utilise la technologie de conteneurs Singularity[4], est disponible, mais il n'est plus maintenu officiellement.

Tous ces modèles ont des faiblesses communes : ils sont tous centralisés (source d'attaque potentielle de type DDoS), ne proposent pas tous les mêmes fonctionnalités et services, ont des niveaux de sécurité plus ou moins forts et des périmètres d'utilisation différents. Afin de palier à ces dépendances, des structures institutionnelles et académiques proposent, à leur niveau, des aides à la construction, au déploiement ou à l'utilisation de conteneurs sur leurs ressources hébergées.

Aujourd'hui, l'initiative du projet CSAN (*Comprehensive Software Archive Network*)[5] est de proposer à la communauté ESRI un catalogue d'applications scientifiques open-source à jour, prêtes à l'emploi et optimisées pour les centres de calcul nationaux, les mésocentres et les infrastructures distribuées de calcul. Le projet est composé d'une vingtaine de personnes issues de divers instituts de l'ESRI (CNRS, INRAE, INRIA, IFB, Université de Montpellier et Centrale Nantes) et de mésocentres (ceux de Montpellier et de l'institut de calcul intensif).

2 Le projet CSAN

La proposition du projet CSAN part du constat que les applications scientifiques dans les différents mésocentres sont souvent les mêmes selon les secteurs (bioinformatique, physique, etc). La plateforme CSAN permettra aux auteurs de logiciels d'y déposer les versions de leurs sources (ou de pointer vers des dépôts versionnés). Ensuite, le groupe d'experts de CSAN traitera ce code et analysera la configuration, via des méthodes d'intégration et de développement continu (CI/CD), afin de le rendre accessible et installable sans effort sur différents systèmes d'exploitation.

En effet, dans le cadre de leurs analyses/simulations, les chercheurs et ingénieurs sont souvent amenés à installer. Les sources de ces programmes se trouvent dans une « forêt » de sites web ou des dépôts de code publics et privés. Certains programmes sont empaquetés avec des outils dédiés (comme Brew, Conda, Spack, Guix, ou EasyBuild) facilitant généralement leur installation au niveau local. Mais cette diversité d'outils rajoute finalement des difficultés pour les utilisateurs et du travail supplémentaire pour les équipes de développement et de l'infrastructure, qui doivent se soumettre aux exigences spécifiques de ces outils et au suivi sans fin de leurs dernières versions au risque de laisser des brèches de sécurité et de négliger l'optimisation des codes. Cette étape chronophage fait que chaque programme supporte rarement plus de deux outils de packaging. Bien entendu, ce packaging peut être fait par de tierces personnes. Les programmes peuvent également être mis à disposition via des conteneurs sur une sorte de "Marketplace". Se pose ensuite le problème de qualité et de sécurité de ces solutions.

Ces problèmes sont bien connus depuis des années par la communauté informatique. Des modèles de distributions éprouvés plus sécurisés et mieux contrôlés (Quality of Service - QoS) existent ; c'est le cas par exemple des dépôts apt pour l'OS Debian ou encore ceux des *Comprehensive Archive Network*, tels que le *Comprehensive Perl Archive Network* (CPAN) [6] ou le *The Comprehensive R Archive Network* (CRAN)[7] ou encore le *Comprehensive TeX Archive Network* (CTAN)[8]. De nombreux « miroirs » permettent de s'assurer que les ressources à télécharger sont toujours disponibles et accessibles rapidement. De nombreuses vérifications complètent ce système (sommes de contrôle, compatibilité avec les dépendances et le système cible, ...).

Pour construire cette nouvelle base d'archives, celle-ci doit se baser sur un système de conteneurisation (type Singularity ou Docker) et éventuellement via des outils tiers de packaging (type Guix ou Nix). En effet, la conteneurisation à elle seule ne peut apporter le principe de reproductibilité indispensable pour une science ouverte : si elle garantit de reproduire l'environnement d'exécution logicielle des codes, elle ne permet pas de garantir la reproductibilité de facto, il faut respecter un ensemble de bonnes pratiques et s'exécuter sur un environnement matériel similaire. Aussi le choix d'un outil de packaging comme Guix pourrait être motivé par le fait qu'il est l'un des rares à répondre au besoin de reproductibilité. Concernant Singularity, il est à ce jour la technologie de conteneurisation la plus répandue dans le monde du HPC.

L'archive CSAN pourra ainsi se présenter sous la forme d'un portail de catalogue d'applications déployables dont chaque version renseigne des spécifications matérielles et logicielles (options de compilation, type de processeur, etc.) ainsi que les coordonnées d'un ou des mainteneur·s et toutes les sources et recettes pour reproduire le package à partir de zéro. Les paquets pourront être consultés et évalués via une plateforme web sous forme d'un hub. Chaque paquet publié du catalogue devra être testé, optimisé et validé par un groupe d'experts composé d'un ou plusieurs référents scientifiques et techniques de manière à proposer à la communauté des applications vérifiées et validées dans des environnements de calcul scientifique identifiés. Un premier démonstrateur basé sur le logiciel libre Harbor[9] a été mis en place, il propose d'ores et déjà le partage de conteneurs entre utilisateurs et équipes informatiques. Il permet de connecter plusieurs registres d'images (n'importe quel registre respectant les standards Container Runtime Interface), de

répliquer l'image sur un autre registre (respectant lui aussi les standards), d'accéder de manière authentifiée au travers d'un annuaire LDAP ou d'un service d'authentification ouvert (type OpenID ou fédération d'identité), de définir des accès basés sur les rôles (RBAC – pour différencier les contributeurs des administrateurs ou des développeurs), de scanner et signer les images (pour garantir la sécurité), de rajouter des étiquettes et d'y accéder par une API Restful.

3 Dépôt d'un programme dans CSAN

Prenons un utilisateur lambda qui aimerait partager un programme ou un code de son projet avec d'autres utilisateurs, ou tout simplement pouvoir simplement accéder à ce programme dans plusieurs mésocentres, sans avoir à se préoccuper de le maintenir à jour partout. Cet utilisateur doit, à minima :

- fournir un fichier décrivant le conteneur (par exemple un fichier Dockerfile si c'est sous Docker ou GUIX si c'est sous guix…) respectant les bonnes pratiques (explicitées sur le portail),
- remplir les métadonnées obligatoires (cf Annexe).
- Une fois la soumission effectuée, un ensemble de procédés automatiques va construire l'image, scanner le code pour déceler d'éventuelles failles de sécurité et faire tourner les tests unitaires.
- Une fois les tests passés et les failles de sécurités identifiés, le conteneur est publié sur le portail CSAN avec des informations sur l'auteur et les métadonnées.
- A posteriori, un sysadmin de mésocentre va s'assurer que le programme tourne de manière optimisée sur le système de son mésocentre, et le cas échéant valider ce conteneur pour son mésocentre.
- Les utilisateurs du portail pourront ainsi vérifier que le conteneur est recommandé et validé pour tourner dans le mésocentre de son choix.
- Une fois connecté dans un environnement de calcul, notre utilisateur (ou n'importe quel autre utilisateur) pourra récupérer le conteneur simplement (via un pull ; docker pull conteneur:latest par exemple) et faire tourner le programme sur ses données locales.

4 Conclusion

Un premier PoC (Proof Of Concept) CSAN a été rendu public pour une phase de bêta test en 2020, aujourd'hui une 1re version de production est en cours de déploiement. Un premier partenariat a été conclu avec la coopération du Mésocentre de Montpellier et l'infrastructure de recherche France Grilles pour l'hébergement de ce service. Ainsi, il est d'ores et déjà possible, aussi bien pour un centre de calcul ou toute personne de confiance identifiée, de partager ses images de conteneurs avec les labels désirés, en indiquant, par exemple, une compatibilité avec un type d'architecture matérielle et logicielle, tout en publiant sur son propre registre sécurisé. Le registre étant ensuite orchestré par Harbor, permettant ainsi un partage et une capitalisation des pratiques. Par ailleurs, l'outil Harbor étant décentralisé, il sera possible de faire communiquer plusieurs instances de Harbor, permettant ainsi un maillage bien plus large et une résilience plus forte de l'architecture. Afin d'en assurer la visibilité au niveau national et européen, ce service sera enregistré dans le méta-catalogue *European Open Science Cloud* - (EOSC) [10]

Le portail CSAN offrira donc un nouveau service stratégique aux développeurs, aux utilisateurs finaux ainsi qu'aux administrateurs système. Ce service permettra d'aboutir à un gain de temps substantiel pour tous. Il permettra aussi une meilleure diffusion des logiciels produits tout en

favorisant la notion de reproductibilité et de partage dans une communauté promouvant de plus en plus l’open-source et la science ouverte.

Annexe

Liste des métadonnées caractérisant un conteneur déposé sur CSAN

métadonnée	obligatoire	Description
version build	x	Version du soft
version conteneur	x	Version du conteneur
auteur	x	Auteur ou liste d'auteurs du conteneur
mainteneur	x	Mainteneur·s de cette version du conteneur
date	x	Date à laquelle la recette a été publiée (format YYYYMMDDHHMM)
licence		licence, si existante
mésocentre		Nom du mésocentre identifié
recette	x	Fichier de recette (ou à inclure dans l'image, ou URL)
spécificités matérielles		Compatibilité avec des spécifications matérielles de l'hôte (nvidia, infiniband...)
spécificités logicielles		Compatibilité avec noyau / pilotes de l'hôte
liste des dépendances, librairies et logiciels inclus		Dépendance inclue dans l'image (lister les logiciels, librairies et leurs versions)
liste des dépendances inverses		Si vous avez connaissance d'autres images dépendant de votre image, merci de les lister
tags	x	étiquettes supplémentaires pour l'image (ex: champs et sous-champs disciplinaires, ...)
Readme, utilisation		Détails ou documentation sur l'utilisation de l'image (ou URL)

Bibliographie

[1] https://hub.docker.com

[2] https://quay.io

[3] https://singularityhub.github.io

[4] https://sylabs.io/singularity

[5] http://csanhub.org/

[6] https://www.cpan.org

[7] https://cran.r-project.org

[8] https://ctan.org

[9] https://goharbor.io/

[10] https://marketplace.eosc-portal.eu/