JupyterHub

Du Mooc « Understanding Queues » à une solution Jupyterhub globalisée (ou presque)

Emmanuel BRAUX – IMT Atlantique / DISI

Cargoday #10 When the dev Meets Ops

Le besoin : Mooc « Understanding Queues »

- Environnement de TP pour MOOC « Understanding Queues »
- Plateforme de MOOC EDX.org
- Contenu sous forme de Notebook Jupyter
- Nombre de participants : inconnu, estimé entre 100 et 1 000
- Un serveur est en cours d'acquisition
- Si possible pas de ré-authentification des utilisateurs
- Début de la session dans 4 mois

Questions / Contraintes

- Comment fonctionne la plateforme EDX ?
  Gestion des utilisateurs / authentification / intégration de ressources externes …

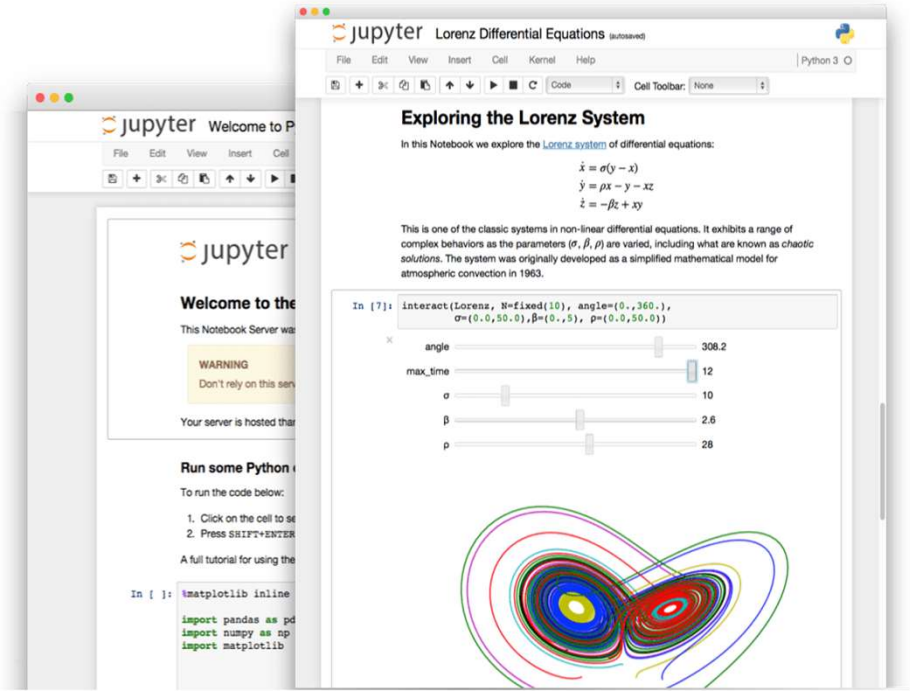
- C’est quoi les Notebooks Jupyter, comment ça marche ?

- Caractéristiques du serveur ? Possible d’intercepter la commande ?

- Il faut trouver une solution simple et scalable si possible
- Il va falloir être efficace et pratique

- Solution alternative ?

Notebook Jupyter

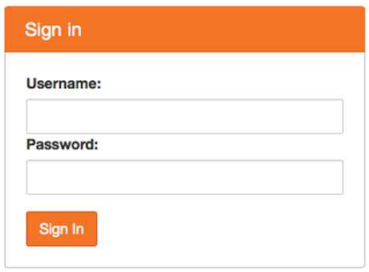


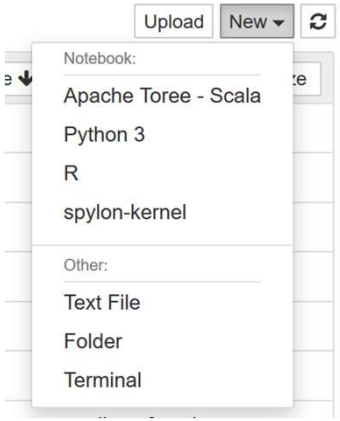
- Application web, permettant de créer des documents mêlant :
  - du texte (markown, html, latex...),
  - du code interactif (python, php, ...),
  - des outils de visualisation de données,
  - ...
- Installation simple (conda, pip, ...), export pdf,

https://jupyter.org/

http://mboileau.pages.math.cnrs.fr/notebook-mania2/notebook-mania2.html

Jupyterhub :
Partager des Notebook Jupyter





- En enseignement, ou en équipe de recherche
- Pour simplifier :
  - un portail web
  - qui permet de gérer des utilisateurs
  - qui lance des instances de notebook Jupyter
- Approche actuelle : "Zero to JupyterHub with Kubernetes", chez Google ou Amazon

https://jupyterhub.readthedocs.io/en/stable/

Notebook Jupyter, les usages

- Jupyter très utilisés
  - en enseignement
  - en traitement et analyse de données
  - …

« Les notebook Jupyter, c'est vraiment simple et portable »

- Jupyterhub assez répandu également:

« on est autonome, ça marche, en 20 minutes en suivant une vidéo sur youtube. »

Notebook Jupyter : les retours

Jupyter :

- On a souvent des problèmes avec les environnements des élèves (sous Windows)

Jupyterhub:

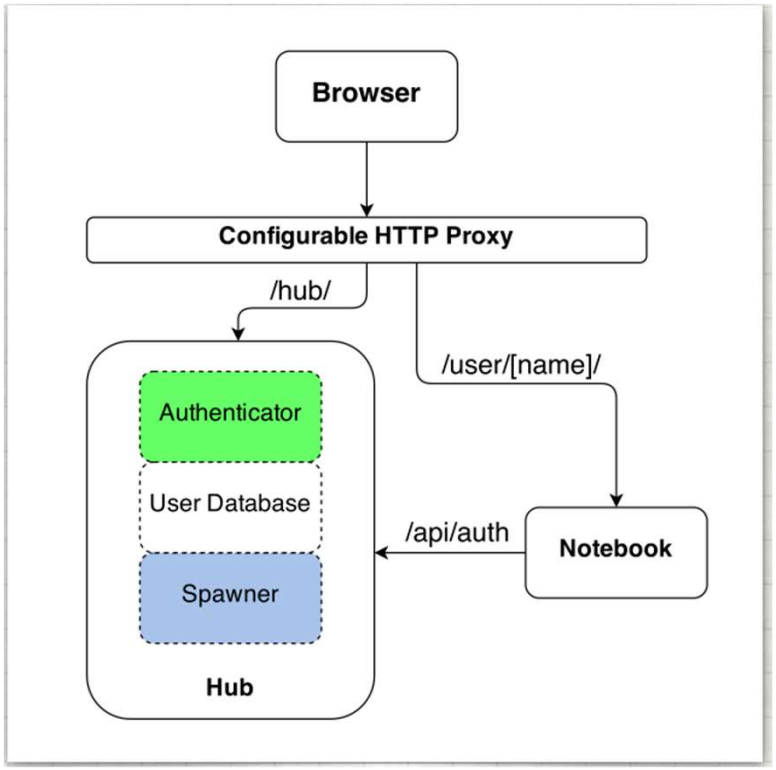
- Il faut le financer :
  - Acheter un serveur
  - En mode cloud : env 100€ pour une petite session (en « fonctionnement »)
- Il faut le gérer
  - créer les comptes des élèves
  - sauvegarder les données
  - ….
- « Faire la même chose pour les copains qui trouvent ça super, mais qui sont trop nuls pour le faire eux même »

Bilan du tour de table

- Jupyterhub semble être une bonne piste

- Notes pour plus tard:
  - Réfléchir à une solution globalisée

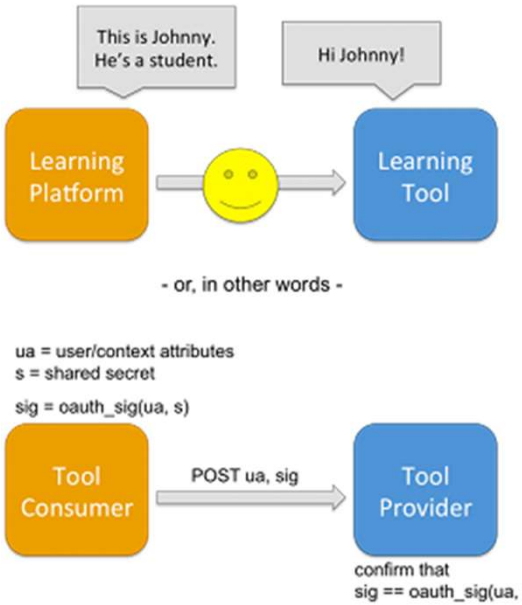
Principe de l'architecture Jupyterhub



- Portail web : point d'accès unique vers des notebook jupyter **[http-proxy]**
- Gestion des utilisateurs **[Authenticator]**
  - Dummy
  - PAM, Ldap, Kerberos, …
  - OAUTH (gitlab, ...)
- Lancement de notebook Jupyter **[Spawners]**
  - "Standard linux" : local system, batch, sudo, ...
  - "Cloud" : docker, Kuberntes, Openstack, ...

https://github.com/jupyterhub/jupyterhub

[Authenticator]: « OAuth », puis « LTI »



- EDX :
  - gère sa propre base d'utilisateurs
  - supporte Oauth, mais uniquement dans version communautaire
  - Supporte l'iIntégration de ressources externes via 'LTI' (Learning Tools Interoperability)
- Authenticator LTI : « bricolé », puis officiel.

[Spawner] : « System » puis « Docker »

System

- Installer Jupyter, et les outils utilisés par les notebooks
- Gérer les utilisateurs, et l'isolation des ressources
- Scalabilité complexe

Docker

- Images fournies par jupyterhub : simple et évolutif
- Isolation des espaces utilisateurs
- Persistance des données via l'utilisation de volumes.
- Scalabilité possible : orchestrateur swarm/kubernetes
- Bonus: autonomie des utilisateurs pour la mise au point des notebooks (si ils savent utiliser docker)

Bilan

- Utilisation de Jupyterhub,
  - en tant que provider LTI
  - Image Docker « base-notebook » personnalisée

- Utilisation
  - 600 instances créées sur les 4 mois du MOOC
  - Très bon retours des particpants

- Mais aussi :
  - 100 instances supplémentaires pour des enseignements (Moodle compatible LTI)
  - Puis utilisation pour des WorkShop (image Spark)
  - Puis …..

Solution Globalisée

Une instance unique ? … Besoins trop différents

Instance dédiée pour chaque besoin :

- Choix de l'image Docker pour le Notebook,
- Choix du mode d'authentification,
- Estimation des ressources nécessaires
- Déploiement automatisés via Ansible, sur Openstack

Limitations / contraintes

Sécurité des accès

- Accès à des données externes ?
- Accès direct à internet ?

Limitation des performances :

- Orchestration de containers
- Spawner Openstack

Choisir un [Authenticator]

Aspects à prendre en compte

- niveau de sécurité
- nombre de personnes et fréquence d'utilisation
- intégration à l'existant

Dépend du contexte :

- Enseignement/Recherche : ldap
- Workshop : "dummy" (x utilisateurs, toujours le même MDP)
- Intégration EDX, Moodle : LTI

Choisir un [Spawner]

Aspects à prendre en compte

- Déploiement
- Mise à jour,
- Autonomie des utilisateurs dans la gestion des Notebook
- Sécurité
- Interaction avec le reste du SI, et l'extérieur
- Performance / ressources
- Contraintes d'utilisation (temps de lancement, pérennité des données, ...)

Évolutions

- Gestion des performances :
  - déploiement en mode "cluster" Swarm, ou kubernetes
  - déploiement en cloud public ou hybride

- Gestion de la sécurité
  - Sécuriser les container (accès réseau, accès root, ...)
  - Sécuriser les images de notebook à déployer (intégration continue)

- Gestion du Déploiement
  - Améliorer la procédure : heat/Terraform
  - Rendre les utilisateurs plus autonomes

Évolutions

- Module d'auto-évaluation (NbGrader)
  https://nbgrader.readthedocs.io/en/stable/
- Jupyterhub As a service (binderhub)
  https://binderhub.readthedocs.io/en/latest/
- Environnement de travail complet (Jupyterlab)
  https://jupyterlab.readthedocs.io/en/latest/
- …

When the dev Meets Ops

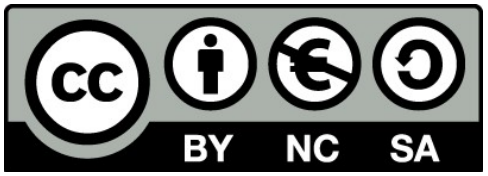
- Passer « de l'autre côté »
  - Accepter de passer un POC en Prod
  - Privilégier la réactivité : « quick and dirty »
  - S'adapter au besoin : personnalisation, pas de sur-qualité

- Sans pour autant vendre son âme
  - Capitalisation, Amélioration continue
  - Test/Validation du « contrat de service »
  - Automatisation !!!

Auteur : emmanuel.braux@imt-atlantique.fr

Cette présentation est sous License Créative Commons 3.0 France (CC BY-NC-SA 3.0 FR)

Selon les options : Attribution - Pas d'Utilisation Commerciale - Partage dans les Mêmes Conditions.



http://creativecommons.fr/licences