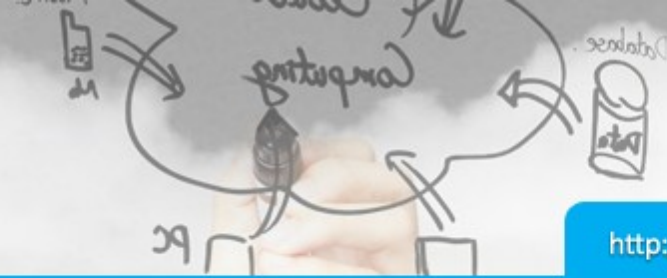




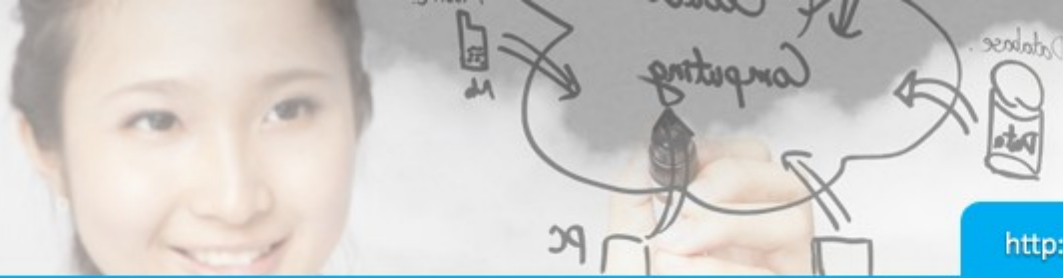
# Déploiement d'une infrastructure iRODS

Jérôme Pansanel et Emmanuel Medernach

15 mars 2021



# Introduction



## Gestion de données

### Données

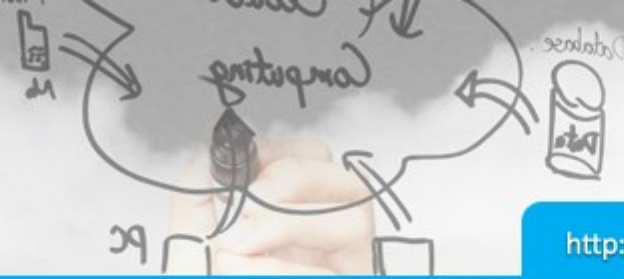
- Où sont stockées les données ?
- Faut-il les centraliser ou les distribuer (avantage / inconvénient) ?
- Faut-il stocker plusieurs copies des données ?
- Quelle quantité de données doit être stockée ? Quelle évolution à prévoir ?
- Quelle est le format des fichiers (ouvert / propriétaire) ?
- Est-ce que les données sont confidentielles ?

### Réseau

- Quel est le débit des réseaux ? Est-il stable dans le temps ?
- Quelle confiance pouvons-nous avoir ?

### Points liés au logiciel

- Est-ce que le projet de déploiement intègre un budget pour la partie logicielle ?
- Quelle est la position des décideurs par rapport à l'Open Source ?



## Gestion de données

### Organisation

- Quelle est l'organisation / gouvernance du projet
- Quelles sont les libertés décisionnelles et opérationnelles des participants du projet ?
- Qui est responsable ? Qui pilote ?

### Utilisateurs

- Quels types d'utilisateurs accèdent aux données ? Quels sont les privilèges à leur accorder ?
- Combien d'utilisateurs utiliseront le service ? Est-ce que ce nombre va croître, et dans quelle proportion ?
- Où sont situés les utilisateurs ? Quel support devons-nous leur apporter ?

## Points à considérer pour le dimensionnement

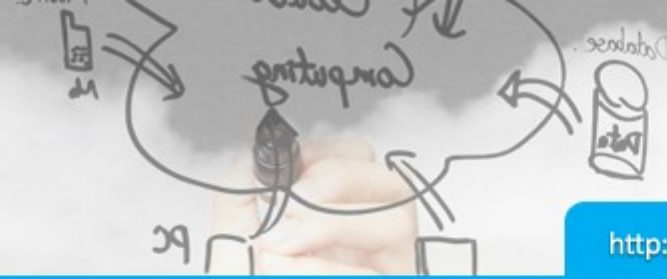
### Plusieurs éléments

- Nombre d'utilisateurs et nombre de connexions simultanées
- Taux d'ingestion
- Taille des fichiers
- Évaluation du nombre de petits fichiers (*overhead*)
- Rapport entre les accès en lecture / écriture partielle et l'accès à une copie complète
- Réplication pour la durabilité
- Réplication pour la localisation (par ex. près d'un cluster HPC)
- Répartition de charge vs haute disponibilité (théorème CAP)

## Configuration matérielle

### Une configuration légère ...

- iRODS occupent moins de 100 Mo (mais les dépendances ont besoin de plus d'espace)
- Serveur iCAT avec minimum 2 Go de RAM
- La taille de la base de données est dimensionnante pour le choix du serveur
- Dans un environnement de production, il est conseillé de séparer la base de données (sous forme cluster) en dehors du service de catalogue
- Pour une ressource de 40 To, un bi-pro AMD avec 32 Go de RAM permet de saturer deux liens 1 Gb/s
- Sur du matériel récent, il est possible de saturer des liens à 10 Gb/s



# Pré-requis

## Avant de commencer

### Connexion aux machines virtuelles

- Les identifiants sont sur le *pad* partagé
- `ssh -i <chemin_vers_votre_clé> centos@134.158.151.xxx`

### Documentation

- Documentation officielle : <https://docs.irods.org/master/>
- Support communautaire : <https://groups.google.com/g/irod-chat>
- Groupe francophone : <https://groups.google.com/g/french-irods-user-group>

### En cas de problème

- [https://docs.irods.org/master/system\\_overview/troubleshooting/](https://docs.irods.org/master/system_overview/troubleshooting/)
- Le fichier de log : `/var/lib/irods/log`



## L'importance de la configuration du réseau

### Le réseau est crucial

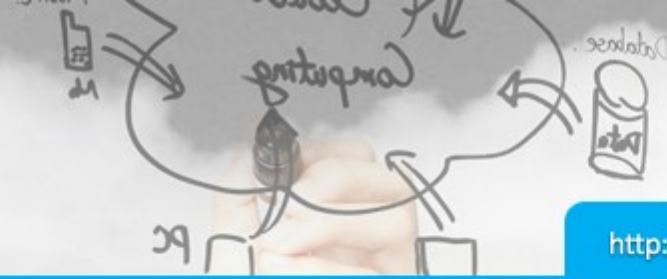
- Éviter le NAT devant un serveur iRODS
- En cas de pare-feu en amont du serveur, vérifiez-en la configuration et les valeurs de *timeout*.
- Effectuer des tests de transfert de gros fichiers (== le plus gros fichier que vous serez amené à transférer sur votre infrastructure)
- Dans une zone, il est conseillé de rajouter les paires noms / IP des serveurs dans `/etc/hosts` en cas de ralentissement dû à une surcharge DNS ou à cause d'un problème de résolution. Il faut effectuer cette modification sur tous les serveurs (icat et ressource). Il faut que ``hostname`` soit résolu.

## Spécificité du réseau Cloud

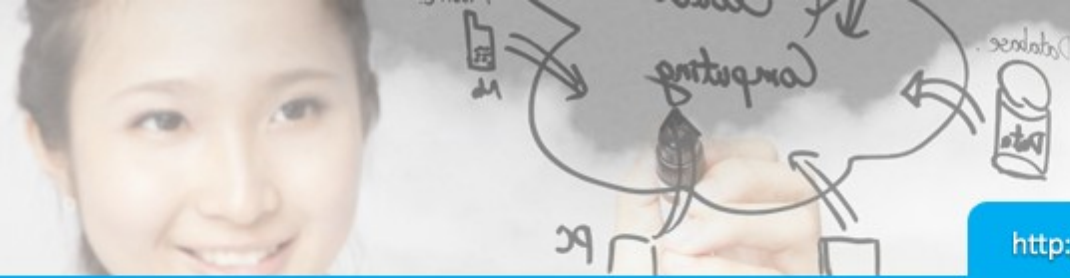
### Infrastructure de formation

- Hébergée sur le Cloud France Grilles (Université de Lille et IPHC)
- Système de NAT :
  - IP locale : 192.168.aaa.xxx
  - IP publique : 193.50.94.zzz
  - One-to-one NAT
- Modifier le fichier `/etc/hosts` sur vos trois serveurs :

```
192.168.aaa.xxx icat-X icat-X.novalocal  
192.168.aaa.yyy resourcel1-X resourcel1-X.novalocal  
192.168.aaa.zzz resource2-x resource2-x.novalocal
```



# Installation d'un serveur iCAT



## Le serveur iCAT

### Un composant essentiel

- Stockage des informations :
  - À propos de la zone
  - Les données et les méta-données associées
  - Le système de fichier virtuel
  - La configuration des ressources
  - La base des utilisateurs
- Bases PostgreSQL, MySQL et Oracle
- Possible de le configurer en HA

## Installation de la base de données

### En tant qu'utilisateur centos

```
$ sudo yum install -y wget epel-release lsof telnet yajl
$ sudo yum install -y postgresql-server
$ sudo yum update -y
$ sudo /usr/bin/postgresql-setup initdb
$ sudo systemctl start postgresql
$ sudo systemctl enable postgresql
$ sudo shutdown -r now
$ sudo systemctl status postgresql
$ sudo su - postgres
```

### En tant qu'utilisateur postgres

```
$ psql
CREATE DATABASE "ICAT";
CREATE USER irods WITH PASSWORD 'astrongpassword';
GRANT ALL PRIVILEGES ON DATABASE "ICAT" to irods;
\q
$ exit
```

## Installation d'iRODS

### En tant qu'utilisateur centos, installation du dépôt iRODS

```
$ sudo rpm --import \  
https://packages.irods.org/irods-signing-key.asc  
$ wget -qO - https://packages.irods.org/renci-irods.yum.repo |  
sudo tee /etc/yum.repos.d/renci-irods.yum.repo  
# Verification  
$ cat /etc/yum.repos.d/renci-irods.yum.repo
```

### Puis installation de iRODS

```
$ sudo yum install -y irods-server irods-database-plugin-postgres
```

## Spécificité CentOS 7

**En tant qu'utilisateur root, modifier le fichier  
/var/lib/pgsql/data/pg\_hba.conf :**

```
# IPv4 local connections:  
host      all             all             127.0.0.1/32      md5  
# IPv6 local connections:  
host      all             all             ::1/128           md5
```

**Puis redémarrer le service postgresql**

```
$ sudo systemctl restart postgresql
```

## Configuration d'iRODS

### Modifier le fichier `localhost_setup_postgres.input` :

```
$ sudo cp \
  /var/lib/irods/packaging/localhost_setup_postgres.input \
  /home/centos/
# valeur a modifier :
# testpassword (mot de passe de de l'utilisateur iRODS pour pgsq1
# (sel pour le stockage des mots de passe dans la base iCAT
# tmpZone (nom de la zone)
# TEMPORARY_ZONE_KEY (zone key)
# 32_byte_server_negotiation_key__ (negotiation key)
# 32_byte_server_control_plane_key (Control Plane Key)
# rods (iRODS Administrator Password)
```

### Trois clés importantes

- Control plane key : 32 caractères
- Negotiation key : 32 caractères
- Zone key : pas de restriction (sauf qu'il ne faut pas de trait d'union)
- Doivent être identiques dans toute la zone



## Configuration d'iRODS

### Configurer le serveur iCAT :

```
$ sudo python /var/lib/irods/scripts/setup_irods.py <
/home/centos/localhost_setup_postgres.input
```

**Note :** le programme de configuration peut planter

### Test du service

```
$ sudo /etc/init.d/irods status
irodsServer :
  Process 14910
  Process 14911
irodsReServer :
  Process 14913
$ exit
$ sudo su - irods
$ ils
/tempZone/home/rods:
```

# Anatomie d'une installation iRODS

## Plusieurs fichiers

```

/etc/irods/core.* - iRODS Rule Language
/etc/irods/host_access_control_config.json - hostname filtering
/etc/irods/hosts_config.json - local /etc/hosts style
configuration
/etc/irods/server_config.json - primary server configuration
/etc/irods/service_account.config - service account information
/usr/bin/* - iCommands

/usr/sbin/irodsAgent
/usr/sbin/irodsPamAuthCheck
/usr/sbin/irodsReServer
/usr/sbin/irodsServer

/var/lib/irods - service account home directory
/usr/lib/irods/plugins - plugins location

```

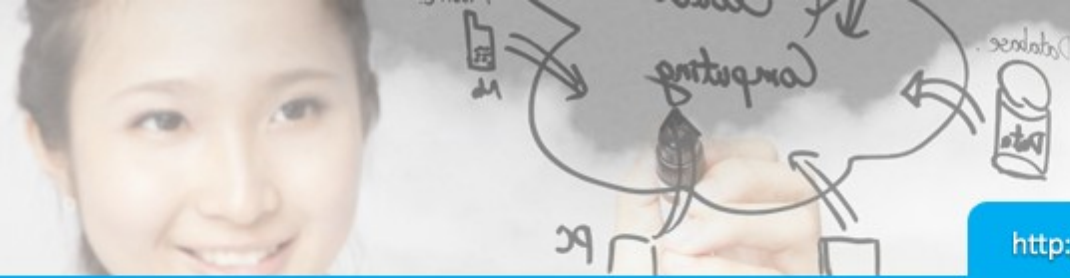
## iCommands

### Les commandes utilisateurs

- Proches des commandes unix :
  - `ils`
  - `icd`
  - `ipwd`
  - `iput`
  - `iget`
  - `irepl`
- Option `-h` pour avoir de l'aide pour chaque commande
- **`ihelp`** affiche toutes les commandes disponibles

### Les commandes administrateurs

- Avec la commande **`iadmin`**



## Bonnes pratiques

### Quelques bonnes pratiques

- [https://docs.irods.org/master/system\\_overview/best\\_practices/](https://docs.irods.org/master/system_overview/best_practices/)
- Ne pas utiliser la ressource demoResc (créée par défaut en production, /var/lib/irods/Vault)
- Utiliser un alias (type *passthru*) pour la ressource racine (permet de modifier les ressources sans modifier les configurations des utilisateurs) :

```
$ iadmin mkresc rootResc passthru
$ iadmin addchildtoresc rootResc ...
```

- Utiliser un fichier de règles locales au lieu de modifier core.re (qui risque d'être écrasé à la prochaine mise à jour), en modifiant /etc/irods/server\_config.json :

```
"re_rulebase_set": [
  "rules",
  "core"
],
```



# Installation des serveurs de ressource

## Installation de la base de données

### En tant qu'utilisateur centos

```
$ sudo yum install -y wget epel-release lsof telnet yaql  
$ sudo yum update -y  
$ sudo shutdown -r now
```

## Installation d'iRODS

### En tant qu'utilisateur centos, installation du dépôt iRODS

```
$ sudo rpm \
  --import https://packages.irods.org/irods-signing-key.asc
$ wget -qO - https://packages.irods.org/renci-irods.yum.repo |
  sudo tee /etc/yum.repos.d/renci-irods.yum.repo
# Verification
$ cat /etc/yum.repos.d/renci-irods.yum.repo
```

### Puis installation de iRODS

```
$ sudo yum install -y irods-server
```

## Configuration d'iRODS

### Quelques étapes

- Configurer le réseau
- Créer le fichier server.config (→ diapo suivante)
- Configurer le service :

```
$ sudo python /var/lib/irods/scripts/setup_irods.py <  
server.config
```

- Redémarrer le service irods
- Créer un répertoire /storage :

```
$ sudo mkdir /storage  
$ sudo chown -R irods.irods /storage
```



## Fichier server.config

```
irods
irods
2
tempZone
icat.novalocal
1247
20000
20199
1248

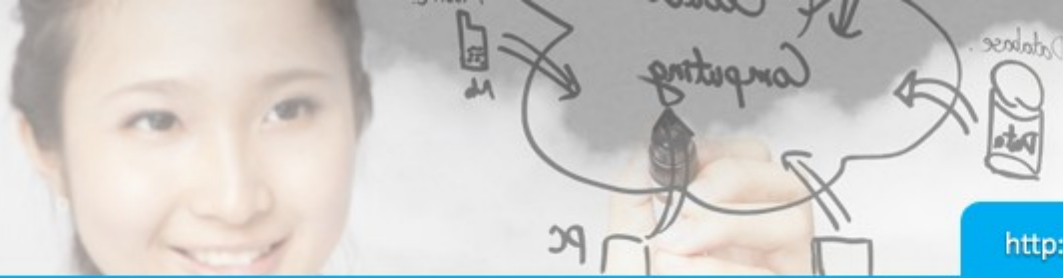
rods
yes
TEMPORARY_ZONE_KEY
32_byte_server_negotiation_key__
32_byte_server_control_plane_key
rods
```

## Créer la ressource sur le serveur iCAT

### Quelques étapes

- Sur le serveur iCAT, en tant qu'utilisateur **irods** :

```
$ iadmin mkresc resourcel unixfilesystem \
  resourcel-X.novalocal:/storage
$ ilsresc
demoResc:unixfilesystem
resourcel:unixfilesystem
resourcel-1Resource:unixfilesystem
rootResc:passthru
$ iadmin rmresc demoResc
$ iadmin rmresc resourcel-XResource
$ iadmin addchildtoresc rootResc resourcel
$ ilsresc
rootResc:passthru
└─ resourcel:unixfilesystem
```



## Test

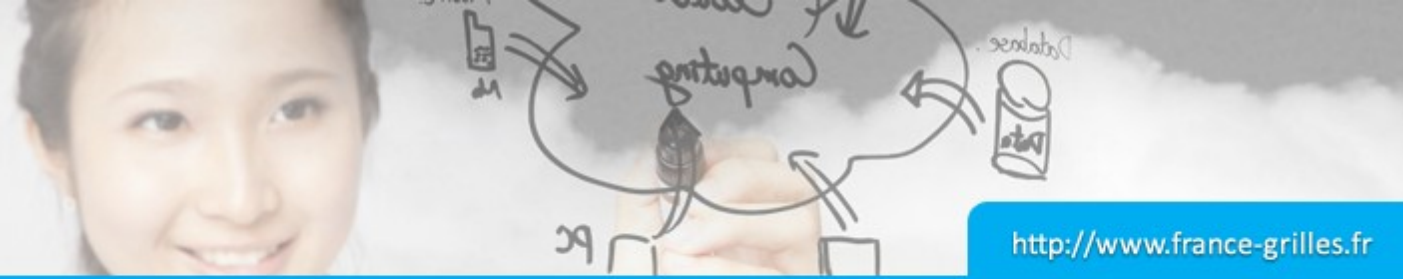
### Copie de fichiers

- Pour les utilisateurs, définir `rootResc` pour la variable `irods_default_resource` dans tous les fichiers `~/.irods/irods_environment.json`
- Puis effectuer

```
$ iexit
$ iinit
$ iput VERSION.json
$ ils -l VERSION.json
  rods                0 rootResc;resource1          224 2021-
01-05.10:56 & VERSION.json
```

- Vérification sur le serveur resource1-X

```
$ ls /storage/home/rods
VERSION.json
```



## Déploiement

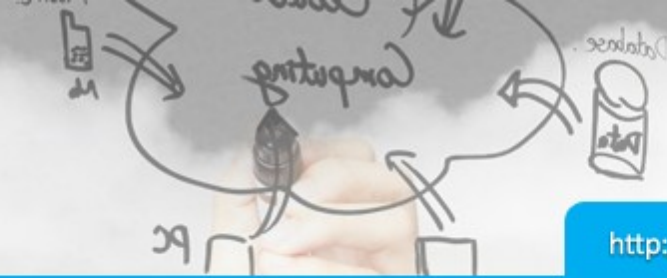
### Ansible

- <https://github.com/0x3bfc/iRODS>
- <https://github.com/CyVerse-Ansible/ansible-irods-cfg>
- [https://git.cines.fr/poc\\_irods/poc-irods](https://git.cines.fr/poc_irods/poc-irods)

### Puppet

- <https://github.com/VEuPathDB/puppet-irods>
- <https://github.com/EUDAT-B2SAFE/B2SAFE-puppet>

### Quattor



# Questions ?