

# Université **IBM i**

## 7 novembre 2023

IBM Innovation Studio Paris

### S05 – Orchestration des travaux IBM i avec service-commander

11:15 / 12:15

**Gautier Dumas**

CFD-Innovation

*gdumas@cfd-innovation.fr*

 **infrasdufutur**

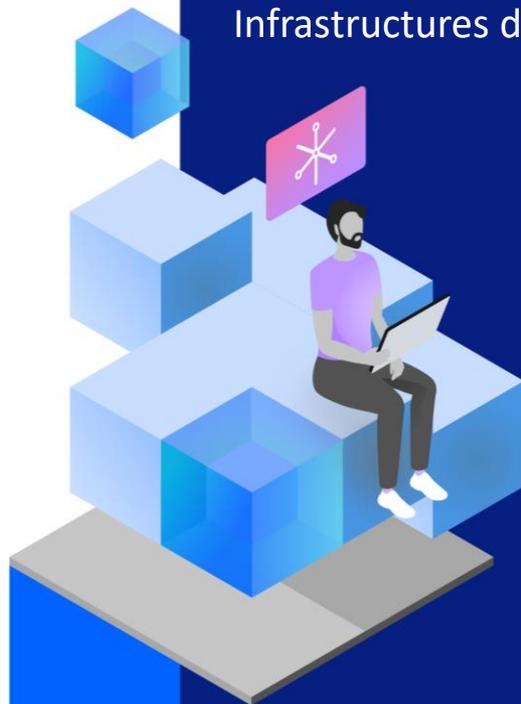
#ibmi

#uui2023

#infrastructuredufuturIBM23



Infrastructures du futur



7 et 8 novembre 2023

# Agenda



1. L'environnement Open Source
2. Présentation de service-commander
3. Démo
4. Conclusion



Université **IBM i**

7 novembre 2023

# 1. L'environnement Open Source



Let's  
Create

# Aperçu du catalogue rpm

## Langages de développements

- PHP
- Node.js
- Python
- R
- Perl

## Des outils

- chroot
- Curl
- Orion
- rsync
- openssl
- Gzip
- Wget
- ibm-iaccess
- Git
- Ansible
- Bash
- Vim
- chsh
- **service-commander**
- ca-certificates-mozilla
- siteadd

## Des serveurs

- Nginx
- Cronie
- MariaDB
- Sqlite
- Node

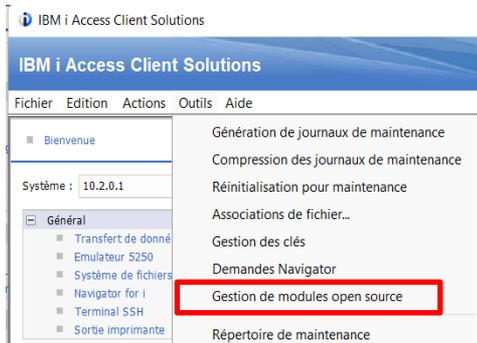
# Environnement Open Source – YUM et RPM



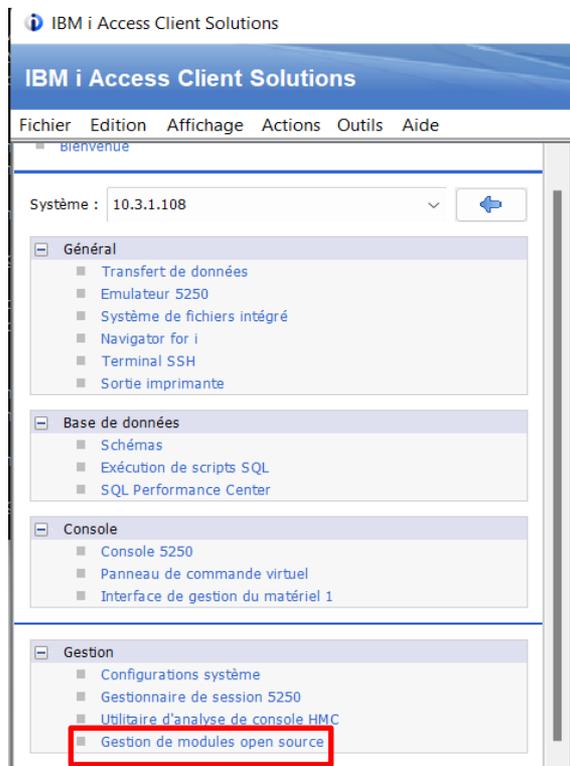
- Les solutions Open Source sous IBM i sont délivrées par des packages **RPM** et le gestionnaire de packages **YUM**
  - Accessible à partir de la V7R3
    - Encore possible de l'installer manuellement en V7R2
  - **YUM peut-être installé puis ensuite utilisé via le produit ACS (IBM Access Client Solutions) (version 1.1.8 minimum), en mode graphique**
  - Le produit 5733-OPS n'est plus nécessaire
- Avantages
  - Mise à disposition beaucoup plus rapide des nouveaux packages
  - Un standard dans le monde Linux
- Répertoire d'installation
  - Les packages sont installés sous /QOpenSys/pkg

# Gestion dans ACS

- A partir de la version **1.1.8** d'ACS, nous retrouvons l'option dans Outils-> Gestion des modules open source



- A partir de la version **1.1.9** d'ACS, nous retrouvons l'option sur l'écran d'accueil dans l'encart Gestion



# Gestion de modules open source

## ■ Connexion -> Passe par SSH

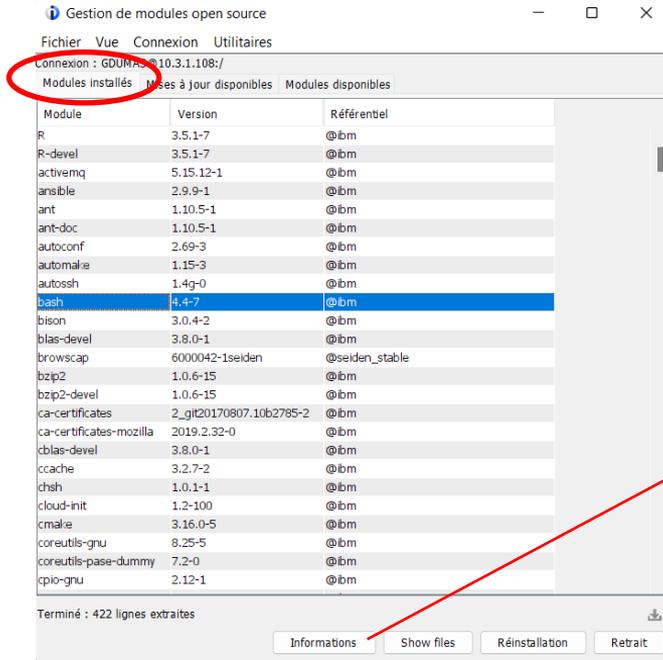
The screenshot shows a dialog box titled "Connexion à SSH" with a close button (X) in the top right corner. The dialog is divided into several sections:

- Système :** A dropdown menu showing "10.3.1.108". A red arrow points from this field to the text "Système IBM i cible".
- Utilisateur :** A text input field containing "GDUMAS". A red arrow points from this field to the text "Profil utilisateur".
- Mécanisme d'authentification :** A section containing:
  - Mot de passe :** A text input field with masked characters (dots). A red arrow points from this field to the text "Authentification".
  - Clé SSH (facultatif) :** A text input field next to a "Survolt" button.
- Conteneur :** A section containing:
  - Chargement du logiciel dans un conteneur Chroot spécifique :** A red arrow points from this checkbox to the text "Choix du scope (container ou scope global si décoché)".
  - A dropdown menu below the checkbox showing a slash "/" as the selected option.
- Mode Proxy :** A section containing:
  - Aucun.** Utilisez cette option lorsque le système ci-dessus est connecté à Internet.
  - Tunnellisation SSH.** Utilisez cette option pour vous connecter à Internet via cet ordinateur. A red arrow points from this option to the text "Pour utiliser la connexion internet du PC".

At the bottom of the dialog are two buttons: "OK" and "Annulation".

# Modules installés

- Liste des modules installés sur le système cible, accès aux informations des packages



```
Package Information
seiden_stable_74/primary_db | 41 kB 00:00
Installed Packages
Name      : bash
Arch     : ppc64
Version  : 4.4
Release  : 7
Size     : 8.0 M
Repo    : installed
From repo : ibm
Summary  : GNU Bourne Again Shell
URL      : http://www.gnu.org/software/bash/bash.html
License  : GPL-3.0-or-later
Description : Bash is the GNU Project's shell. Bash is the Bourne Again Shell.
           : Bash is an sh-compatible shell that incorporates useful features
           : from the Korn shell (ksh) and C shell (csh). It is intended to
           : conform to the IEEE POSIX P1003.2/ISO 9945.2 Shell and Tools
           : standard. It offers functional improvements over sh for both
           : programming and interactive use. In addition, most sh scripts can
           : be run by Bash without modification.

Available Packages
Name      : bash
Arch     : ppc64
Version  : 5.1
Release  : 2
Size     : 2.3 M
Repo    : ibm
Summary  : GNU Bourne Again Shell
URL      : http://www.gnu.org/software/bash/bash.html
License  : GPL-3.0-or-later
Description : Bash is the GNU Project's shell. Bash is the Bourne Again Shell.
           : Bash is an sh-compatible shell that incorporates useful features
           : from the Korn shell (ksh) and C shell (csh). It is intended to
           : conform to the IEEE POSIX P1003.2/ISO 9945.2 Shell and Tools
           : standard. It offers functional improvements over sh for both
           : programming and interactive use. In addition, most sh scripts can
           : be run by Bash without modification.
```

# Mises à jour disponibles

- Liste des mises à jour disponibles pour les packages déjà installés

Gestion de modules open source

Fichier Vue Connexion Utilitaires

Connexion : GDUMAS@193.51.108/

Modules installés Mises à jour disponibles Modules disponibles

Module	Version	Référentiel
libiconv2	1.14-5	ibm
libidn2-0	2.3.0-2	ibm
libiperf-devel	0.11.1-1	ibm
libiperf1	0.11.1-1	ibm
libncurses6	6.0-8	ibm
libopenssl1_1	1.1.1n-1	ibm
libp11-kit0	0.23.14-2	ibm
libpcre1	8.45-1	ibm
libpcreposix0	8.45-1	ibm
libreadline8	8.1-2	ibm
libsodium-devel	1.0.18-1	ibm
libsodium23	1.0.18-1	ibm
libutil-devel	0.11.1-1	ibm
libutil2	0.11.1-1	ibm
libuv-devel	1.42.0-1	ibm
libuv1	1.42.0-1	ibm
libzstd-devel	1.5.2-1	ibm
libzstd1	1.5.2-1	ibm
logrotate	3.15.1-3	ibm
man-db	2.9.1-5	ibm
maven	3.6.0-2	ibm
ncurses-devel	6.0-8	ibm
ncurses-terminfo	6.0-8	ibm
ncurses-tools	6.0-8	ibm
nginx	1.16.1-4	ibm
nodejs12	12.22.11-1	ibm
nodejs14	14.19.1-1	ibm
nspr	4.32-1	ibm

Terminé : 149 lignes extraites

Informations Mise à niveau

Package Upgrade

```

Setting up Upgrade Process
Resolving Dependencies
--> Running transaction check
--> Package man-db.ppc64 0:2.9.1-4 will be updated
--> Package man-db.ppc64 0:2.9.1-5 will be an update
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package          Arch          Version        Repository     Size
=====
Updating:
man-db           ppc64         2.9.1-5        ibm            1.9 M
=====

Transaction Summary
=====
Upgrade          1 Package

Total download size: 1.9 M
Is this ok [y/N]: y
Downloading Packages:
man-db-2.9.1-5.ibmi7.2.ppc64.rpm          | 1.9 MB 00:01
Running Transaction Check
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Updating   : man-db-2.9.1-5.ppc64 [#####] 1/2
    
```

# Modules disponibles

- Liste des modules disponibles (non installés) dans les différents référentiels de packages enregistrés sur le système cible

Gestion de modules open source

Fichier Vue Connexion Utilitaires

Connexion : GDUMAS@10.3.1.108:/

Modules installés Mises à jour disponibles **Modules disponibles**

Module	Version	Référentiel
ImageMagick	7.0.11.14-5seiden	seiden_stable
ImageMagick-data	7.0.11.14-5seiden	seiden_stable
ImageMagick-devel	7.0.11.14-5seiden	seiden_stable
ImageMagick-doc	7.0.11.14-5seiden	seiden_stable
R	3.6.3-1	ibm
R-devel	3.6.3-1	ibm
ansible	2.9.10-2	ibm
ant	1.10.10-1	ibm
ant-doc	1.10.10-1	ibm
argon2	20190702-2seiden	seiden_stable
bash	5.1-2	ibm
blas-devel	3.8.0-2	ibm
brotil	1.0.9-1	ibm
brotil-devel	1.0.9-1	ibm
browscap	600046-1seiden	seiden_stable
c-ares-devel	1.17.2-1	ibm
c-ares-utils	1.17.2-1	ibm
ca-certificates	2_git20170807.10b2785-3	ibm
ca-certificates-mozilla	2021.2.46-1	ibm
cairo-devel	1.17.4-2	ibm
cblas-devel	3.8.0-2	ibm
charset-alias	1.14-5	ibm
cloud-nit	1.3-1	ibm
cmake	3.16.9-1	ibm
coreutils-gnu	8.25-6	ibm
cronie-anacron	1.5.5-4	ibm
cups-devel	2.3.1-1	ibm
freetsds	1.2.18-3	ibm

Terminé : 372 lignes extraites

Informations Installation

Package Installation

```
Setting up Install Process
Resolving Dependencies
--> Running transaction check
-->> Package siteadd.ppc64 0:0.7-0seiden will be updated
-->> Package siteadd.ppc64 0:0.10-0seiden will be an update
--> Processing Dependency: bash >= 5.1 for package: siteadd-0.10-0seiden.ppc64
--> Running transaction check
-->> Package bash.ppc64 0:4.4-7 will be updated
-->> Package bash.ppc64 0:5.1-2 will be an update
--> Finished Dependency Resolution

Dependencies Resolved

-----
Package      Arch      Version      Repository      Size
-----
Updating:
siteadd      ppc64     0.10-0seiden seiden_stable   301 k
Updating for dependencies:
bash        ☐ ppc64   5.1-2        ibm              2.3 M
-----

Transaction Summary
-----
Upgrade      2 Packages

Total size: 2.6 M
Total download size: 301 k
Is this ok [y/N]: y
Downloading Packages:
siteadd-0.10-0seiden.ibm17.2.ppc64.rpm | 301 kB 00:00
Running Transaction Check
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Updating : bash-5.1-2.ppc64 [#####] 1/4
```

# Gestion des packages Open Source - ACS

- Informations complémentaires :

<http://www-01.ibm.com/support/docview.wss?uid=nas8N1022619>

➔ Getting started with Open Source Package Management in IBM i ACS

# Contrôle et maitrise

- De plus en plus de technologies différentes sur vos IBM i
- Il nous faut « un service pour les gouverner tous »



Université **IBM i**

7 novembre 2023

Let's  
Create

## 2. Présentation service-commander



# Présentation



- Projet GitHub accessible sur :
  - <https://github.com/ThePrez/ServiceCommander-IBMi>
- Par Jesse Gorzinski
  - alias ThePrez sur GitHub : <https://github.com/ThePrez>
- Avec des contributeurs reconnus de la communauté IBM i OSS

Contributors 13



# Introduction service-commander

- Gestion de divers services et applications exécutés sur IBM i
- Objectif :
  - Fournir une interface ligne de commande intuitive et facile à utiliser pour gérer des services ou des tâches
  - Intégration avec STRTCPSVR
- Exemples de services pouvant être gérés :
  - Travaux IBM i
  - Serveurs TCP de l'IBM i : \*FTP, \*SSHD, \*HTTP ...
  - Programmes utilisant des technologies Open Source (Node.js, Python, PHP etc...)
  - Instances Apache
  - Le démon Cron
  - Base de données Open Source (PostgreSQL, MariaDB)
  - ...

# Les fonctionnalités

- service-commander permet :
  - de gérer des services (arrêt / démarrage)
  - de spécifier des dépendances entre les services
    - Le démarrage d'un service peut lancer le démarrage de ses dépendances automatiquement
  - de soumettre un job en batch
  - de check le statut d'un service par le port / nom du job
  - de configurer les variables d'environnement de chaque job
  - de gérer les services par groupe de services
  - de consulter des statistiques de performance des services
  - de gérer les logs des services
  - de consulter la liste des ports actuellement ouvert

# Différence avec les autres solutions

- Service-commander est différent des autres outils qui accomplissent des tâches similaires (comme init.d, supervisord ...)
- Ne stocke pas l'id du processus qu'il lance (couplage faible) + pas d'accès spécial aux structures de données du noyau
  - service-commander émet des hypothèses fortes basées sur des vérifications d'un nom de tâche ou d'une utilisation de port particulière
  - Une limitation connue est donc que service commander peut confondre une autre tâche avec un service configuré en fonction de l'un de ces attributs (exemple utilisation du port 80)
- Configuration à l'aide de fichiers .yaml

# Installation

- Installation : `yum install service-commander`
- Prérequis (qui sont installés automatiquement lors de l'installation) :
  - Db2util (`yum install db2util`)
  - OpenJDK (`yum install openjdk-11`)
  - Bash (`yum install bash`)
  - GNU coreutils (`yum install coreutils-gnu`)
- Les informations de performance (`perfinfo`) ont des prérequis supplémentaires :
  - Python 3 avec `ibm_db connector` (`yum install python3-ibm_db`)
  - OS level :
    - IBM i 7.4: included with base OS
    - IBM i 7.3: Group PTF SF99703 Level 11
    - IBM i 7.2: Group PTF SF99702 Level 23

# Démarrage avec service-commander

- Pour commencer avec une configuration par défaut (pré-alimentée)
  - `sc_install_defaults`
  - Avec l'option `--cleanupv0` si vous aviez déjà la v0
- Cela va installer et définir les services :
  - Cron (si cronie est installé)
  - MariaDB (si mariadb est installé)
  - Les instances d'Apache de l'IBM i (sauf si option `--noapache` spécifiée)

- Après initialisation, vous pouvez lancer le premier **sc check**

```

NOT RUNNING | apache_ajsp (Apache server ajsp)
RUNNING     | apache_apachedft (Apache server apachedft)
RUNNING     | apache_artichaut (Apache server artichaut)
NOT RUNNING | apache_iwadft (Apache server iwadft)
NOT RUNNING | apache_serveur2 (Apache server serveur2)
RUNNING     | apache_sphinxdemo (Apache server sphinxdemo)
RUNNING     | apache_turlurette (Apache server turlurette)
RUNNING     | apache_wsdemo2 (Apache server wsdemo2)
NOT RUNNING | apache_wservice (Apache server wservice)
NOT RUNNING | crond (Cron daemon)
NOT RUNNING | mariadb (MariaDB Server)
NOT RUNNING | postgres (PostgreSQL Server)

-bash-5.1$ █

```

# Usage de base pour des services préconfigurés

- `sc start <service_name>`
  - Démarrage d'un service
- `sc stop <service_name>`
  - Arrêt d'un service
- `sc check group:all`
  - Vérifier l'état de tous les services configurés
- `sc start group:all`
  - Démarrage de tous les services
- `sc info group:all`
  - Affiche des informations sur tous les services configurés

```
-----  
crond (Cron daemon)  
  
Defined in: /QOpenSys/etc/sc/services/oss_common/crond.yaml  
  
Startup Command: /QOpenSys/pkg/bin/crond -n  
Startup Wait Time (s): 60  
  
Shutdown Wait Time (s): 45  
  
Check-alive conditions: JOBNAME:PGM-CROND  
Batch Mode: <submitted to batch>  
  Batch Job Name: CROND  
  SBMJOB options: JOBQ(QUSRNOMAX)  
  
Inherits environment variables?: true  
-----
```

- sc info crond

```
-----  
apache_wsdemo2 (Apache server wsdemo2)  
  
Defined in: /home/GDUMAS/.sc/services/apache_wsdemo2.yaml  
  
Startup Command: system "STRTCPSVR SERVER(*HTTP) HTTPSVR(wsdemo2)"  
Startup Wait Time (s): 60  
  
Shutdown Command: system "ENDTCPSVR SERVER(*HTTP) HTTPSVR(wsdemo2)"  
Shutdown Wait Time (s): 45  
  
Check-alive conditions: PORT:20224  
Batch Mode: <not running in batch>  
  
Inherits environment variables?: true  
-----
```

- sc info apache\_wsdemo2

# Usage de base

- `sc start group:host_servers`
  - Démarre tous les services du groupe « host\_servers »
- `sc list group:all`
  - Lister tous les services
- `sc jobinfo port:8080`
  - Liste les jobs tournant sur le port 8080
- `sc stop port:8080`
  - Arrête les travaux tournant sur le port 8080
- `sc check port:8080`
  - Vérifie si le port est utilisé
- `sc start myservice.yaml`
  - Démarre le service défini dans un fichier local myservice.yaml

# Usage de base

- sc groups
  - Liste tous les groupes configurés
- sc groups --ignore-globals
  - Liste seulement les groupes définis dans les fichiers de configurations yaml privés
- scopenports
  - Liste les ports utilisés (avec une description sur les services configurés dans service-commander)

```

Address      Port      'sc' service name (and friendly name)
-----
0.0.0.0     21        system_ftp (System FTP server)
::          21        system_ftp (System FTP server)
0.0.0.0     22        system_sshd (System Secure Shell server)
::          22        system_sshd (System Secure Shell server)
0.0.0.0     23        system_telnet (System Telnet server)
::          23        system_telnet (System Telnet server)
0.0.0.0     25        system_smtp (System SMTP server)
::          25        system_smtp (System SMTP server)
0.0.0.0     80        apache_apachedft (Apache server apachedft)
0.0.0.0     110       system_pop (System POP3 server)
::          110       system_pop (System POP3 server)
0.0.0.0     137       system_netserver (System NetServer)
0.0.0.0     139       system_netserver (System NetServer)
0.0.0.0     389       system_directoryserver (System Directory Server)
::          389       system_directoryserver (System Directory Server)
127.0.0.1   427       system_slp (System SLP server)
172.17.0.4  427       system_slp (System SLP server)
172.30.14.12 427       system_slp (System SLP server)
:           427       system_slp (System SLP server)
0.0.0.0     445       system_netserver (System NetServer)
::          445       system_netserver (System NetServer)
0.0.0.0     446       system_ddm (System DDM server)
::          446       system_ddm (System DDM server)
0.0.0.0     447       system_ddm (System DDM server)
::          447       system_ddm (System DDM server)
0.0.0.0     448       system_ddm (System DDM server)
::          448       system_ddm (System DDM server)
0.0.0.0     449       system_as-svrmap (System *SVRMAP Host Server)
::          449       system_as-svrmap (System *SVRMAP Host Server)
0.0.0.0     515       system_lpd (System Line Printer Daemon Server)
::          515       system_lpd (System Line Printer Daemon Server)
0.0.0.0     657       system_rmc (Resource Monitoring and Control [RMC])
0.0.0.0     992       system_telnet (System Telnet server)
::          992       system_telnet (System Telnet server)
0.0.0.0     2001      system_http_admin (HTTP Admin Server)
0.0.0.0     2002      system_admin1 (Navigator for i (latest generation))
0.0.0.0     2004      system_admin2 (IBM Navigator for i)
0.0.0.0     2006      system_admin3 (management (Digital Certificate Manager, Db2 Mirror))
0.0.0.0     2008      system_admin4 (HTTP Administration for i)
0.0.0.0     2012      port:2012
127.0.0.1   2015      port:2015
127.0.0.1   2016      system_admin2 (IBM Navigator for i)
127.0.0.1   2017      system_admin3 (management (Digital Certificate Manager, Db2 Mirror))
127.0.0.1   2018      system_admin4 (HTTP Administration for i)
127.0.0.1   2019      system_admin5 (HTTP Administration UI)

```

# Services « system » préconfigurés

- Service-commander propose des services préconfigurés associés au groupe « system »
  - IBM i Host servers
  - Services classiques de l'IBM i (ftp, ssh ...)
  - Interface d'administration (Navigator for i).
- **sc -a check all** ou **sc -a check group:system** ou **sc list group:system**
  - Pour afficher les services du groupe system qui sont cachés par défaut

```

PARTIAL (2/3) | system_admin1 (Navigator for i (latest generation)) [not running at -->PORT:2003]
PARTIAL (3/6) | system_admin2 (IBM Navigator for i) [not running at -->PORT:2005, PORT:16324, PORT:
RUNNING      | system_admin3 (management (Digital Certificate Manager, Db2 Mirror))
RUNNING      | system_admin4 (HTTP Administration for i)
PARTIAL (2/3) | system_admin5 (HTTP Administration UI) [not running at -->PORT:2011]
NOT RUNNING  | system_ajs (Advanced Job Scheduler)
RUNNING      | system_as-central (System *CENTRAL Host Server)
RUNNING      | system_as-database (System *DATABASE Host Server)
RUNNING      | system_as-dtaq (System *DTAQ Host Server)
RUNNING      | system_as-file (System *FILE Host Server)
RUNNING      | system_as-netprt (System *NETPRT Host Server)
RUNNING      | system_as-rmtcmd (System *RMTCMD Host Server)
RUNNING      | system_as-signon (System *SIGNON Host Server)
RUNNING      | system_as-sts (Service Tools Server)
RUNNING      | system_as-svrmap (System *SVRMAP Host Server)
NOT RUNNING  | system_cimom (System CIM Server)
NOT RUNNING  | system_dbgsvr (System Debug Server)
RUNNING      | system_ddm (System DDM server)
PARTIAL (1/2) | system_directoryserver (System Directory Server) [not running at -->PORT:3539]
RUNNING      | system_ftp (System FTP server)
PARTIAL (2/3) | system_http_admin (HTTP Admin Server) [not running at -->PORT:2010]
RUNNING      | system_lpd (System Line Printer Daemon Server)
RUNNING      | system_netserver (System NetServer)
RUNNING      | system_pop (System POP3 server)
RUNNING      | system_rmc (Resource Monitoring and Control [RMC])
RUNNING      | system_slp (System SLP server)
RUNNING      | system_smtp (System SMTP server)
NOT RUNNING  | system_sptprx (IBM Support Communication Service)
RUNNING      | system_sshd (System Secure Shell server)
RUNNING      | system_telnet (System Telnet server)
NOT RUNNING  | system_webquery (Db2 WebQuery)
NOT RUNNING  | system_ypsjsvr (IBM Management Central Server)

```

# Configurer des Services

- `sc_install_defaults` -> pour des services préconfigurés
- Pour en définir soit même :
  - Outil `scinit` -> création
  - Outil `scredit` -> édition
- Définition de service « à la volée » basée sur un port ou un nom de job (pour des checks en live sans besoin de créer un service dans service-commander)
- Création / Edition dans un fichier YAML

# Utilisation scinit

- Pour une assistance pas à pas à la création d'un fichier de configuration YAML
  - `scinit <program start command>`
- Le process à suivre est le suivant :
  1. `cd` vers le répertoire où on démarre normalement le service
  2. Exécuter la commande pour démarrer le service, préfixé par `scinit`
  3. Répondre à une série de questions sur le déploiement du service
- Cela va créer le fichier YAML de votre service dans
  - `/QOpenSys/etc/sc/services/myapp.yml` (où `myapp` est le nom du service)
- Puis l'afficher

# Exemple définition d'un service node

```

-bash-5.1$ cd /home/GDUMAS/myApp
-bash-5.1$ scinit node app.js
/QOpenSys/pkg/bin/scinit: line 13: warning: setlocale: LC_ALL: cannot change lo
cale (no_PV.UTF-8): No such file or directory
Would you like this service to be available to all users? [n] y
Short name: nodedemo
Friendly name:
Empty response. Asking again
Friendly name: node demo for UUI2023
Must the application be started in the current directory (/home/GDUMAS/myApp)? [
y] y
If your application runs under a unique job name, what is it? Leave blank for no
ne:
Which ports does your application run on? Separate with commas, or leave blank f
or none: 2204
Will your application need to be submitted to batch? [n] n
(Recommended) Will your application need to run with the PATH and JAVA_HOME valu
es of the current process? [y] y
What other environment variables from this current process should be used?
    (press <enter> after each entry, leave blank to finish entering values)
1> PATH
2>
What group(s) would this application be a part of?
    (press <enter> after each entry, leave blank to finish entering values)
1>
What service(s) does this application rely on?
    (press <enter> after each entry, leave blank to finish entering values)
1>
Written to file: /QOpenSys/etc/sc/services/nodedemo.yml

Printing information about the newly-defined service

```

Lancement de scinit

Série de questions pour configuration du service

```

-----
nodedemo (node demo for UUI2023)

Defined in: /QOpenSys/etc/sc/services/nodedemo.yml

Working Directory: /home/GDUMAS/myApp

Startup Command: node app.js
Startup Wait Time (s): 60

Shutdown Wait Time (s): 45

Check-alive conditions: PORT:2204
Batch Mode: <not running in batch>

Inherits environment variables?: true
Custom environment variables:
    PATH=/QOpenSys/pkg/bin:/QOpenSys/usr/bin:/usr/ccs/bin:/QOpenSys/usr/bin/X11
:/usr/sbin:./usr/bin
-----

```

# Utilisation scedit

- A utiliser pour modifier un fichier YAML existant
  - `scedit <service>`
- `scedit` va localiser le fichier YAML du service spécifié et l'ouvrir en fonction de l'éditeur configuré par défaut en suivant la liste suivante
  1. L'éditeur configuré dans la variable d'environnement **\$EDITOR**
  2. Si vide, **nano**
  3. Si nano non installé, **joe**
  4. Si joe non installé, **vim**
  5. Si vim non installé, **vi**

# Ad hoc service definition

- Possible de checker un service à la volée sans l'avoir défini dans service-commander
  - Par le nom du job.  
***sc check job:jobname*** (*jobname accepte aussi subsystem/jobname et le format PGM-\_\_\_\_\_ pour spécifier un certain programme*)
  - Par le port  
***sc jobinfo port:portname***

# Utilisation des fichiers yaml

- Sans passer par les outils scinit ou scedit, il est possible de créer et maintenir soit même ses services dans les fichiers YAML
- On les retrouve dans les emplacements suivants :
  - Un répertoire global : **/QOpenSys/etc/sc/services**.  
*Nécessite d'avoir les droits \*ALLOBJ.*
  - Des répertoires spécifiques par utilisateur : **\$HOME/.sc/services**
- Le nom du fichier doit être dans le format `service_name.yaml` (ou `.yml`), où
  - "service\_name" est le "short name" du service.
  - Le nom ne peut contenir que des lettres minuscules, des nombres, tirets et des underscores
- Le fichier peut aussi être n'importe où dans l'IFS, à partir du moment où le chemin complet est indiqué dans la commande
  - Ex: `sc start /my/specific/path/my_service.yaml`

# Format des fichiers yaml

- Des exemples de services préconfigurés sont disponibles sur :  
<https://github.com/ThePrez/ServiceCommander-IBMi/tree/main/samples>
- Les attributs suivants sont disponibles :
- Champs obligatoires
  - **start\_cmd**: La commande pour démarrer le service
  - **check\_alive**: Ce qui va déterminer si un service est alive ou non : le port, ou le nom du travail
- Champs facultatifs souvent nécessaires
  - **name**: Le nom user-friendly du service (avec espace, minuscules, majuscules etc...)
  - **dir**: Le répertoire de travail dans lequel switcher avant de lancer les commandes de démarrage / arrêt

# Format des fichiers yaml

## ▪ Autres champs

- **stop\_cmd**: La commande d'arrêt du service
- **startup\_wait\_time**: Le temps d'attente en secondes avant que le service soit démarré (60 par défaut)
- **stop\_wait\_time**: Le temps d'attente en secondes avant que le service soit arrêté (45 par défaut)
- **log\_dir**: Le répertoire de stockage des logs créés par Service Commander
- **cluster**: Active le mode cluster
- **batch\_mode**: Indique si le service doit être lancé en batch
  - **sbmjob\_jobname**: Si batch\_mode, le nom du job à utiliser
  - **sbmjob\_opts**: Si batch\_mode, les options du SBMJOB (pour l'instant, utilisation d'une JOBD)
- **environment\_is\_inheriting\_vars**: Active l'héritage des variables d'environnements de l'environnement courant (true par défaut)
- **only\_if\_executable**: Configuration ignorée sauf si le fichier spécifié existe et est exécutable
- **environment\_vars**: Définition de variables d'environnements custom au lancement du service (au format "KEY=VALUE")
- **service\_dependencies**: Un tableau de service duquel dépend ce service.
- **groups**: Regroupement du service dans des groupes custom (Pour démarrage/ arrêt en une seule opération).

# Exemple d'un fichier yaml

- Pour une application node.js utilisant le port 2204

```
name: My Node.js application
dir: /home/GDUMAS/myApp
start_cmd: node index.js
check_alive: '2204'
batch_mode: 'false'
environment_vars:
-PATH=/QOpenSys/pkgsrc/bin:/QOpenSys/usr/bin:/usr/ccs/bin:/QOpenSys/usr/bin/X11:/usr/sbin:./usr/bin
```

# Gestion des logs

- Les logs sont stockés par défaut dans
  - \$HOME/.sc/logs/date\_time.service\_name.log
  - Où :
    - date\_time est l'heure de démarrage du service
    - Service\_name est le nom du service
- Pour un service donné, directive « log\_dir » pour indiquer l'emplacement des logs
- sc loginfo <service name>
  - Va afficher l'emplacement des logs d'un service donné
  - Qu'on peut ensuite afficher (cat, tail ...)
- Possible de diriger les logs vers un \*SPLF
  - sc --splf start <service name>

# Usage avancé – Cluster Mode

- Permet le clustering automatique des applications (s'appuie sur NGINX)
- Quand cluster mode utilisé :
  - SC va démarrer n jobs, chacun tournant sur un port différent
  - SC va gérer les différents jobs lors des opérations sur le service
  - Le travail est load balancé sur les jobs
- Prérequis :
  - L'application doit s'appuyer sur la variable d'environnement **PORT**
  - Ou passer l'information sur la ligne de commande --port=\$PORT pour que l'information soit transmise et utilisée par l'application

# Usage avancé – Cluster Mode - Exemple

- Imaginons un service node simple

```
name: My Node App
dir: /home/GDUMAS/myApp
start_cmd: node ./app.js
check_alive: 2204
```

- Ce service va démarrer un serveur web sur le port 2204

- Activation d'un cluster

```
name: My Node App
dir: /home/GDUMAS/myApp
start_cmd: node ./app.js
cluster: 2206, 2208, 2210, 2212
check_alive: 2204
```

- L'application reste accessible sur le port 2204 pour les utilisateurs.
- 4 jobs supplémentaires seront démarrés respectivement sur les ports indiqués dans la directive cluster.

# Usage avancé – Restart automatique

- Fonction connue et utilisée par les utilisateurs de process manager (comme pm2 pour node)
- Actuellement non intégré dans SC mais...
- Possible de le scripter simplement :

```
SBMJOB CMD(CALL PGM(QP2SHELL2) PARM('/QOpenSys/usr/bin/sh' '-c' 'while ;; do
sleep 40 && /QOpenSys/pkgs/bin/sc start navigator >/dev/null 2>&1 ; done'))
JOB(NAVMON) JOBD(*USRPRF) JOBQ(QUSRNOMAX)
```

- Ici, le script va tenter de démarrer navigator toutes les 40 secondes
  - S'il est démarré, rien n'est fait
  - Sinon, le service est démarré

# Usage avancé – Intégration STRTCPSVR

- Service Commander est maintenant intégré aux commandes STRTCPSVR / ENDTCPSSVR
- Pour intégrer, jouer la commande suivante :
  - /QOpenSys/pkgs/lib/sc/tcpsvr/install\_sc\_tcpssvr
- Cela va créer la bibliothèque SCOMMANDER et compiler/installer les programmes TCP à l'intérieur
- Une fois installé, nous pouvons utiliser \*SC et le nom du service à gérer
- Exemple :
  - STRTCPSVR SERVER(\*SC) INSTANCE('kafka')
- Pratique pour intégrer dans des CL schedulés / QSTRUP



Infrastructures du  
futur

7 et 8 novembre 2023

## Université **IBMi**

7 novembre 2023



Let's  
Create

# 3. Démo

# Visionner la démo

# IBM i



```
-bash-5.1$ sc check -q  
  
NOT RUNNING      | crond (Cron daemon)  
RUNNING           | grafana (Backend Grafana)  
NOT RUNNING      | mariadb (MariaDB Server)  
NOT RUNNING      | nodered (node-red sur le scope global)  
NOT RUNNING      | postgres (PostgreSQL Server)  
RUNNING          | zenddbi (Zend DBi Server)
```



Infrastructures du  
futur

7 et 8 novembre 2023

Université **IBM i**

7 novembre 2023



Let's  
Create

## 4. Conclusion

# Conclusion

- A destination des administrateurs IBM i
  - Mais aussi des développeurs
- Centralise la gestion de divers services
  - IBM i traditionnel
  - Services Open Source
- Permet d'automatiser certaines tâches de monitoring
- Permet de gagner en maîtrise sur les services qui tournent sur IBM i
- Facile et rapide d'accès

