Berechtigungsmanagement für Compute Ordner in SAS Viya

Stefan Park – SAS Institute GmbH

Copyright © SAS Institute Inc. All rights reserved.

- Automatisierung
  - Anpassungen der Viya Konfiguration
  - CAS-Bibliotheken anlegen und berechtigen
  - Ordnerstrukturen in SAS Content anlegen und berechtigen
  - …
- Storage-Berechtigung
  - NFS-Ordnerstrukturen anlegen und Vergabe von POSIX oder extended ACL’s
- SAS Bibliotheken
  - Zuweisung von SAS Base Bibliotheken

- Aufbau eines Container Images
  - Bereitstellung SAS Viya CLI und Installation der Plugins
  - Zusätzliche shell tools: jq, sed, curl, acl, nfs4-acl-tools,
- Automatisierung durch K8s Cronjobs oder CI/CD Pipelines
- Template basierte Konfiguration
  - Erstellung möglichst einfacher aber ausreichender Strukturen für die jeweiligen Zwecke
- Gemeinsame Nutzung der Templates wenn möglich
  - Reduzierung und Vermeidung von Fehlern durch widersprüchliche Konfigurationen

Copyright © SAS Institute Inc. All rights reserved.

Container-Image

Enthält die benötigten Linux tools und eigene Shell-Skritpe

Dockerfile

```
1  FROM docker.io/ubuntu:latest

2  RUN  apt-get update && \
        apt-get upgrade -y && \
        apt-get install curl jq acl nfs4-acl-tools -y

3  COPY tools /sastools

4  RUN /sastools/sas-viya plugins install --repo SAS all

5  RUN chmod +x /sastools/*.sh
```

1. Auswahl des Basis-Images
2. Installation zusätzlicher tools
3. Kopieren der erstellten Skripte in das Image im Unterordner “/sastools”
4. Installation aller aktuellen SAS CLI Plugins
5. Die Shell-Skripte Ausführbar machen

Copyright © SAS Institute Inc. All rights reserved.

Skript Aufbau

- Übersichtlich Skript-Steuerung erstellen
  - Anzeige eines Flags zeigt die übergebenen Parameter an
- Usernamen und Passwörter
  - Verwendung von Secrets in k8s
- Die meisten der benötigten Templates können als einfach Textfiles bereitgestellt werden
  - Ermöglicht ein einfaches Einlesen in Bash Arrays und übergabe der Werte an die SAS Viya CLI und dem jeweiligen Plugin

Shell Skript Ausführung

```
Usage: run_configuration.sh <[options]>
        Options:
           ------------------ Required Options ------------------------------------------
             -u   --user                    Provide user for Viya Login (true)
             -p   --password                Provide Password for Viya Login (true)
             -e   --endpoint                Provide Url to Viya Environment (true)
             -l   --cfg-location            Base Path for config files (true)
             -i   --viya-cli                Location of Viya CLI (true)
           ------------------------------------------------------------------------------
             -f   --folder-cfg              Location and name of folders.json (true)
             -c   --cas                     Set Cas Setup to true    (true)
             -g   --groups                  Set Group Setup to true    (true)
             -m    --group-membership       Set Group Membership to true (true)
             -n   --nfs                     Set Group NFS Setup to true (true)
                  --launcher                Create Launches if not available (true)
                  --context                 Create Context if not available (true)
                  --viya-config-patch       Apply patchfile for Viya (true)
                  --sas-content-folder      Create Folders and apply permissions (true)
                  --create-domains          Create AuthDomains (true)
             -h   --help                    Show this message
```

Copyright © SAS Institute Inc. All rights reserved.

NFS für SAS Compute und CAS

NFSv3

- Performance: NFSv3 vs NFSv4.x
  - (Hinweis: locks and leases)
- NFS 16 Gruppen Limit bei Nutzung von AUTH_SYS bzw. 32 Gruppen mit Kerberos (RFC 5531)
- Standard Linux Berechtigungen:
  - RWX auf Owner, Group oder Everyone Ebene

NFSv4.x

- NFSv4.x benötigt nur einen Port
  - (vorteilhaft bei benötigten Firewall-Freischaltungen)
- NFS 16 Gruppen Limit bei Nutzung von AUTH_SYS bzw. 32 Gruppen mit Kerberos (RFC 5531)
  - Je nach NFS Provider gibt es Möglichkeiten das Limit zu erhöhen , z.B. die Optionen `-extended-groups-limit` und `-auth-sys-extended-groups` bei NetApp ONTAP 9
- Per Default nur die Standard Linux Berechtigungen:
  - RWX auf Owner, Group oder Everyone Ebene
- NFSv4.x ACLs bieten die Möglichkeit mehreren Gruppen und Usern Berechtigungen auf einem File oder Ordner zu erteilen, ähnlich den Berechtigungen in Windows

Copyright © SAS Institute Inc. All rights reserved.

Template

Template Erstellung für die ACL-Vergabe

- Pro Ordner können n Berechtigungen gesetzt werden
- ACL’s setzen sich aus 4 Teilen zusammen:
  - Type, Flag, Principal, Permission
  - NFS4_ACL man page
  - Setzen von Default Linux Berechtigungen
- JSON bietet sich bei dieser 1:n Zuordnung und der (einfachen) Verarbeitung an.
  - Das Format könnte auch gut über eine WebApp eingelesen und verarbeitet werden

JSON Template zur ACL Vergabe

```
{
  "nfs": [
    {
       "Folder": "/nfs/content_backup",
       "Permissions": [
          {
            "Type": "A",
            "Group": "gdf",
            "Id": 20000001,
            "Permission": "RWX"
          },
          {
            "Type": "A",
            "Group": "gdf",
            "Id": 20000002,
            "Permission": "RX"
          }
       ]
    }
   ]
}
```

Copyright © SAS Institute Inc. All rights reserved.

Shell Funktion zur Verarbeitung des Templates

Enthalten im Admin Container Image

Shell Function

```
set_nfsPermissions() {
 local jin=$1

 for k in $(cat ${jin} | jq '.nfs' |jq -r 'keys[]|@text’)
   do

   Folder=$(jq -r ".nfs[$k] .Folder" ${jin})
   mkdir -p $Folder

   for j in $(cat ${jin}|jq .nfs[$k]|jq .Permissions|jq -r 'keys[]|@text’)
     do
       Type=$(jq -r ".nfs[$k].Permissions[$j] .Type" ${jin})
       Group=$(jq -r ".nfs[$k].Permissions[$j] .Group" ${jin})
       Id=$(jq -r ".nfs[$k].Permissions[$j] .Id" ${jsoninput})
       Permission=$(jq -r ".nfs[$k].Permissions[$j] .Permission" ${jin})
     if [[ $j = 0 ]]; then
        nfs4_setfacl -s ${Type}:${Group}:$Id:${Permission} $Folder
     else
        nfs4_setfacl -a ${Type}:${Group}:$Id:${Permission} $Folder
     fi
   done
 done
}
```

1. Durchlaufen der einzelnen Objekte unterhalb vom nfs Array
2. Anlegen der Ordner inklusive der benötigten Struktur wenn nicht vorhanden
3. Parameter mit dem ACL Werten belegen (Type, Flag, Principal, Permission)
4. Beim Setzen der ersten Berechtigungen, wird über das ***-s*** Flag dem nfs4_setfacl mitgeteilt, das alle bestehenden Berechtigungen vorab gelöscht werden

Copyright © SAS Institute Inc. All rights reserved.

Template

Erweiterung - Template Erstellung für die ACL-Vergabe

- In Viya (Stand: 2024.09) sind SAS Base Bibliotheken keine Metadatenobjekte, die wie unter SAS 9 über Berechtigungen gesteuert zugeordnet werden können.
  - Das JSON Templates kann durch eine minimale Anpassung Informationen zu ***Server Contexten*** und der zu verwendenden ***Libref*** erweitert werden.

Erweitertes JSON Template zur ACL Vergabe

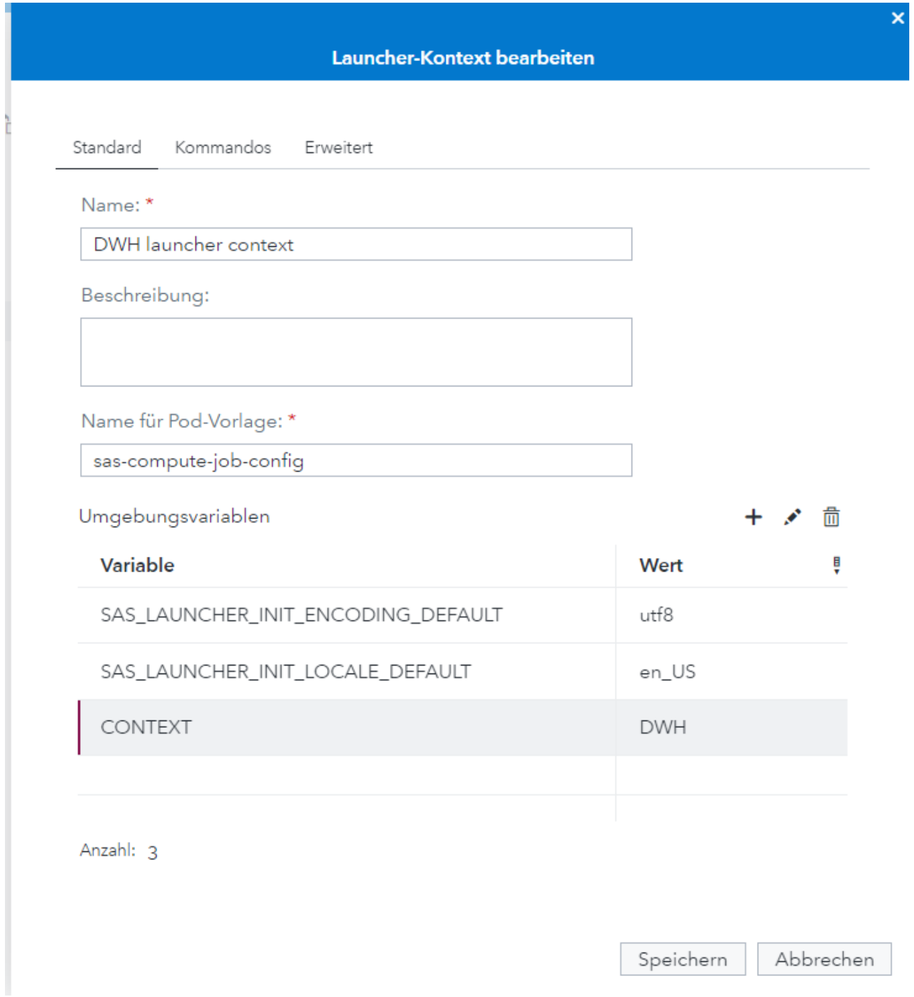
```
{
  "nfs": [
    {
       "Folder": "/nfs/content_backup",
       "Permissions": [
          {
            "Type": "A",
            "Group": "gdf",
            "Id": 20000001,
            "Permission": "RWX"
          },
          {
            "Type": "A",
            "Group": "gdf",
            "Id": 20000002,
            "Permission": "RX"
          }
       ],
       "Libref": "cbak",
       "Context": [
         {"Name": "DWH"}
       ]
    }
   ]
}
```

Copyright © SAS Institute Inc. All rights reserved.

SAS Viya Konfiguration

Anwendung des JSON Templates zur Bibliothekszuordnung

- Bei Verwendung mehrerer Launcher und Server Contexte, wird eine zusätzliche Variable definiert.



Copyright © SAS Institute Inc. All rights reserved.

SAS Viya Konfiguration

Anwendung des JSON Templates zur Bibliothekszuordnung

sas.compute.server: startup_commands

```
# Variable containes user GID’s in comma separated list
GIDS=$(id -G|sed 's/ /,/g’)
# Set base path
SDATA=/nfs

# Extract libraries by context and group permission and add to baselibs.sas
jq -r '.nfs[] |
  select((.Permissions[].Id == ('$GIDS')) and (.Libref !="")) |
  select(.Context[].Name == ("'$CONTEXT'"))|
  [.Folder, .Libref] | "libname \(.[1]) \"\(.[0])\";"' $SDATA/folder.json > /tmp/baselibs.sas
```

sas.compute.server: autoexec_code

```
%include "/tmp/baselibs.sas“;
```

Copyright © SAS Institute Inc. All rights reserved.

Automatisierung

Am Beispiel K8s Cronjob

Ergebnis

- Mit Hilfe eines Triggers kann der Job bei Änderung der folder.json automatisch alle Anpassungen im NFS vornehemen
- Jeder User bekommt entsprechend seiner Berechtigungen SAS Base Bibliotheken zugewiesen
  - Zusätzlich Unterscheidung je Context möglich
- Ausweitung auf den sas.batch.server kann identisch umgesetzt werden

K8s Cronjob Definition (YAML gekürzt)

```
kind: CronJob
metadata:
 name: viya-config-job
spec:
 suspend: true
 schedule: "0 0 * * *"
jobTemplate:
   spec:
     template:
       spec:
         containers:
         - name: viya-config-job
           image: <registry>/viya-admin:latest
           env:
           - name: VIYA_ENDPOINT
             value: "https://..."
           - name: VIYA_PASSWORD
             valueFrom:
               secretKeyRef:
                 name: saslogin
                 key: password
           ...
           command: ["/bin/bash"]
           args: ["-c", '/sastools/run_configuration.sh
                          -u ${VIYA_USER}
                          -p ${VIYA_PASSWORD}
                          -e ${VIYA_ENDPOINT}
                          --folder-cfg /nfs/folder.json
                          --nfs
                 ']
           ...
```

Copyright © SAS Institute Inc. All rights reserved.

Danke

Copyright © SAS Institute Inc. All rights reserved.