[Runbook] Add SFTP Folders

Example task: Pat O'Brien: Another ICSF Request | Korio Engineering > DevOpsTeam | Microsoft Teams

Components

- **User Assigned Managed Identity (UAMI)** - The security principal that will be used in both the ACLs and the Workload ID of the pod
- **Service Account** - A runtime identity that is assigned to a pod in Kubernetes
- **Federated Identity Credential** - used to map a Service Account in Kubernetes to a User Assigned Managed Identity (UAMI) in Azure

Creating the components will require the latest version of `korioctl` - which you can get from the releases page on github.com/korio-clinical/korioctl, or `git clone` the repo and run `mage build` on master to get the bleeding edge

A general runbook can be found here: presto-besto-manifesto: Add biostats identities to prod CLOSED

Note: the ACLs in this PR are incorrect - the leaf user ACLs need to have user with write access

Install latest korioctl

- `git clone github.com/korio-clinical/korioctl`
- `mage build`
- make sure to use the output of `mage build` as your korioctl (`./korioctl`)

Clone presto-besto-manifesto repo

- `git clone github.com/korio-clinical/presto-besto-manifesto`
- `cd presto-besto-manifesto`

Prerequisites

Note that you will need to have the following IAM roles assigned to you in the Blob Storage container in question (i.e. `stagingsftpmirror` for staging, `prodsftpmirror` for prod, etc.):

- Storage Blob Data Owner
- Security Admin

Setup environment

```
# Adjust environment (kenv) and study (study) as needed
export sub_id="1a6c6f27-521f-4e1a-81e6-d855dd0b464a"
export kenv="staging"
export sftp_rg="vozni-${kenv}-sftp-storage"
export study="int-icsf-node"

az login
```

User Assigned Managed Identity (UAMI)

- Check the list of currently provisioned UAMIs with `az identity list`
  - If you need to create any, proceed; else skip to Federated Identity Credential section
- Get the list of active sub-environments for an environment
  - `cat staging/subenvironments.yaml` ⇒ `configure validate preview`

```
# Adjust sub-environments below based on environment
cat "${kenv}/subenvironments.yaml"
for sbenv in configure validate preview; do
  korioctl azure uami create -g "${sftp_rg}" "${kenv}-${sbenv}-${study}"
done
```

- To check if these UAMIs are already present

```
az identity list -g ${sftp_rg} \
  --query "[?type == 'Microsoft.ManagedIdentity/userAssignedIdentities'].{Name:name, Location:location, ClientID:clientId, PrincipalID:principalId}" \
  --output table | grep "${study}"

> staging-configure-int-icsf-node       eastus      8c7d96db-b34a-4fbf-a18f-f6c02816b082  972068d8-f28d-4492-8112-c88fafa93943
> staging-validate-int-icsf-node        eastus      efea86e2-67e2-4428-8426-84a83970e754  5a4e0179-a715-4ff8-8ef7-83c818611aa6
> staging-preview-int-icsf-node         eastus      2dd246c8-689b-4dee-a7b3-e63ec3a855d6  169b86ca-6175-4d0b-ba31-1d503ea6e1b1

```

Federated Identity Credential (FIC)

- Get the target environment's Kubernetes OIDC Issuer URL

```
export oidc_url="$(basename $(az aks show -g "vozni-${kenv}-rg" \
                                --name "vozni-${kenv}-aks" \
                                --query "oidcIssuerProfile.issuerUrl" \
```

```
4                                   -otsv 2> /dev/null))"
```

- Create the Federated Identity Credentials (FICs)

```
1  # Adjust sub-environments below based on environment
2  cat "${kenv}/subenvironments.yaml"
3  for sbenv in configure validate preview; do
4    korioctl azure fic create \
5      -g "${sftp_rg}" \
6      --identity "${kenv}-${sbenv}-${study}" \
7      --issuer "https://eastus.oic.prod-
   aks.azure.com/${sub_id}/${oidc_url}/" \
8      --subject
   "system:serviceaccount:${sbenv}:${kenv}-${sbenv}-${study}" \
9      "${kenv}-${sbenv}-${study}"
10 done
```

- To check for FICs:

```
1  for name in $(az identity list -g "vozni-${kenv}-sftp-storage" \
2                  --query "[].name" -o tsv); do
3    count="$(az identity federated-credential list \
4              -g "vozni-${kenv}-sftp-storage" \
5              --identity-name "$name" --query 'length(@)' \
6              -o tsv 2>/dev/null || echo 0)"
7    if [ "$count" -gt 0 ]; then
8      az identity show -g "vozni-${kenv}-sftp-storage" -n "${name}" | \
9      jq -r '. | "\(.name) \(.location) \(.clientId)"'
10   fi
11 done | grep icsf
12
13 > staging-preview-moderna-icsf eastus 35b15f18-bc4b-44f5-975a-3fb6e4dd435f
14 > staging-accept-moderna-icsf eastus 3d126285-0e92-418d-a4ac-1672741630e6
15 > staging-configure-moderna-icsf eastus 35658d3b-9964-4b8a-a06e-b6cbc8046ab1
16 > staging-my-moderna-icsf eastus 6539b06d-b78a-4c6a-be0e-8f38a521952d
17 > staging-validate-moderna-icsf eastus fe4b8562-1496-4169-b4bb-8a80a0eda309
18
```

Data Lake Directory Paths (DLDP)

Options needed as input:

- `-a` Storage account name
  - Will always be `${kenv}sftpmirror`
- `-f` Filesystem name (also called “Storage Container name”)
  - Will always be `mirror` - static string, no env, nor subenv needed
- `-p` ParentACL
  - Must be specified individually - do NOT specify the default (the code automates making the default ACL the same as the Access ACL)
  - Always going to be the same for everything
    - `-p user::rwx -p group::r-x -p other::--x`
- `-l` LeafACL
  - ACL of the directory being created at the leaf - Where the service and `sftp-server` both write and read file data

    - Will always have the same logical structure
        - Principal IDs change in each subenvironment
        - Principal IDs needed for
            - Per sub-environment
                - `sftp-server` - The UAMI that the data sync container in the `sftp-server` pod runs under:
                    - ```
                      az identity list | \
                      jq -r '.[] | "\(.name) \(.principalId)"' | \
                      grep "^${kenv}\-${sbenv}\-" | column -t | sort | \
                      grep 'sftp\-server'

                      > staging-preview-sftp-server         91024095-0f28-4643-bb1a-b291263729c3

                      export sftp_srv_id="$(az identity list | \
                                           jq -r '.[] | "\(.name) \(.principalId)"' | \
                                           grep "^${kenv}\-${sbenv}\-" | column -t | \
                                           sort | grep 'sftp\-server' | awk '{ print $2 }')"
                      ```
                - `service` - The UAMI just created that the client service pod container runs under
                    - ```
                      az identity list | \
                        jq -r '.[] | "\(.name) \(.principalId)"' | \
                        grep "^${kenv}\-${sbenv}\-" | column -t | sort | \
                        grep "${study}"

                      > staging-preview-int-icsf-node       169b86ca-6175-4d0b-ba31-1d503ea6e1b1

                      export client_srv_id="$(az identity list | \
                                             jq -r '.[] | "\(.name) \(.principalId)"' | \
                                             grep "^${kenv}\-${sbenv}\-" | column -t | \
                                             sort | grep "${study}" | awk '{ print $2 }')"
                      ```
            - Per environment
                - `sftp-read` group - Entra AD security group of humans assigned read access:
                    - ```
                      az ad group show -g "sftp-read-${kenv}" | \
                      jq -r '. | "\(.displayName) \(.id)"'

                      > sftp-read-staging 1c22a08a-6ad2-4dd4-b15b-ff7f9c4315f7

                      export sftp_ro_id="$(az ad group show -g "sftp-read-${kenv}" | \
                                           jq -r '.id')"
                      ```
                - `sftp-read-write` group - Entra AD security group of humans assigned read-write access:
                    - ```
                      az ad group show -g "sftp-read-write-${kenv}" | \
                      jq -r '. | "\(.displayName) \(.id)"'

                      > sftp-read-write-staging 57c4dc8f-e68c-47ad-8b1b-17f80d8a0a9b

                      export sftp_rw_id="$(az ad group show -g "sftp-read-write-${kenv}" | \
                                           jq -r '.id')"
                      ```
- Create folders for environment and subenvironment (based on folder names requested):

```
korioctl azure dldp create \
  -a "${kenv}sftpmirror" \
  -f mirror \
  -p user::rwx -p group::r-x -p other::--x \
  -l user::r-x -l group::r-x -l other::--- \
  -l "user:${sftp_srv_id}:rwx" \
  -l "user:${client_srv_id}:rwx" \
  -l "group:${sftp_ro_id}:r-x" \
  -l "group:${sftp_rw_id}:rwx" \
  -l mask::rwx \
  "${kenv}/${sbenv}/moderna/icsf/Moderna/mRNA-4194-P101/Inventory"
```

```
  korioctl azure dldp create \
  -a "${kenv}sftpmirror" \
  -f mirror \
  -p user::rwx -p group::r-x -p other::--x \
  -l user::r-x -l group::r-x -l other::--- \
  -l "user:${sftp_srv_id}:rwx" \
  -l "user:${client_srv_id}:rwx" \
  -l "group:${sftp_ro_id}:r-x" \
  -l "group:${sftp_rw_id}:rwx" \
  -l mask::rwx \
  "${kenv}/${sbenv}/moderna/icsf/Moderna/mRNA-4194-P101/Patient"
```

```
  korioctl azure dldp create \
  -a "${kenv}sftpmirror" \
  -f mirror \
  -p user::rwx -p group::r-x -p other::--x \
  -l user::r-x -l group::r-x -l other::--- \
  -l "user:${sftp_srv_id}:rwx" \
  -l "user:${client_srv_id}:rwx" \
  -l "group:${sftp_ro_id}:r-x" \
  -l "group:${sftp_rw_id}:rwx" \
  -l mask::rwx \
  "${kenv}/${sbenv}/moderna/icsf/Moderna/mRNA-4194-P101/Site"
```

Validating you used the correct ID (principal, not client)

It is easy to use the client or `sftp-server` Client IDs rather than the Principal IDs. The Principal ID is what is required for the ACLs.

Using the CLI

You can view the ACL assignments using the CLI:

```
az storage fs access show \
  --account-name "${kenv}sftpmirror" \
  --file-system mirror \
  --path "/${kenv}/${sbenv}/moderna/icsf/Moderna/mRNA-4194-
P101/Patient" \
  --auth-mode login \
  --query acl -o tsv \
  | tr ',' '\n'

> user::r-x
> user:2dd246c8-689b-4dee-a7b3-e63ec3a855d6:rwx
> user:cb826355-5d62-40b8-9b66-4bd69c34a8e8:rwx
> group::r-x
> group:1c22a08a-6ad2-4dd4-b15b-ff7f9c4315f7:r-x
> group:57c4dc8f-e68c-47ad-8b1b-17f80d8a0a9b:rwx
> mask::rwx
> other::---
> default:user::r-x
> default:user:2dd246c8-689b-4dee-a7b3-e63ec3a855d6:rwx
> default:user:cb826355-5d62-40b8-9b66-4bd69c34a8e8:rwx
```

```
> default:group::r-x
> default:group:1c22a08a-6ad2-4dd4-b15b-ff7f9c4315f7:r-x
> default:group:57c4dc8f-e68c-47ad-8b1b-17f80d8a0a9b:rwx
> default:mask::rwx
> default:other::---

az storage fs access show \
  --account-name "${kenv}sftpmirror" \
  --file-system mirror \
  --path "/${kenv}/${sbenv}/moderna/icsf/Moderna/mRNA-4194-P101/Site" \
  --auth-mode login \
  --query acl -o tsv \
  | tr ',' '\n'

> user::r-x
> user:2dd246c8-689b-4dee-a7b3-e63ec3a855d6:rwx
> user:cb826355-5d62-40b8-9b66-4bd69c34a8e8:rwx
> group::r-x
> group:1c22a08a-6ad2-4dd4-b15b-ff7f9c4315f7:r-x
> group:57c4dc8f-e68c-47ad-8b1b-17f80d8a0a9b:rwx
> mask::rwx
> other::---
> default:user::r-x
> default:user:2dd246c8-689b-4dee-a7b3-e63ec3a855d6:rwx
> default:user:cb826355-5d62-40b8-9b66-4bd69c34a8e8:rwx
> default:group::r-x
> default:group:1c22a08a-6ad2-4dd4-b15b-ff7f9c4315f7:r-x
> default:group:57c4dc8f-e68c-47ad-8b1b-17f80d8a0a9b:rwx
> default:mask::rwx
> default:other::---

```

Note that in this case, the user IDs do not match the needed Principal IDs:

```
az identity list | \
jq -r '.[] | "\(.name) \(.principalId)"' | \
grep "^${kenv}\-${sbenv}\-" | column -t | \
sort | grep 'sftp\-server' | awk '{ print $2 }'

> 91024095-0f28-4643-bb1a-b291263729c3

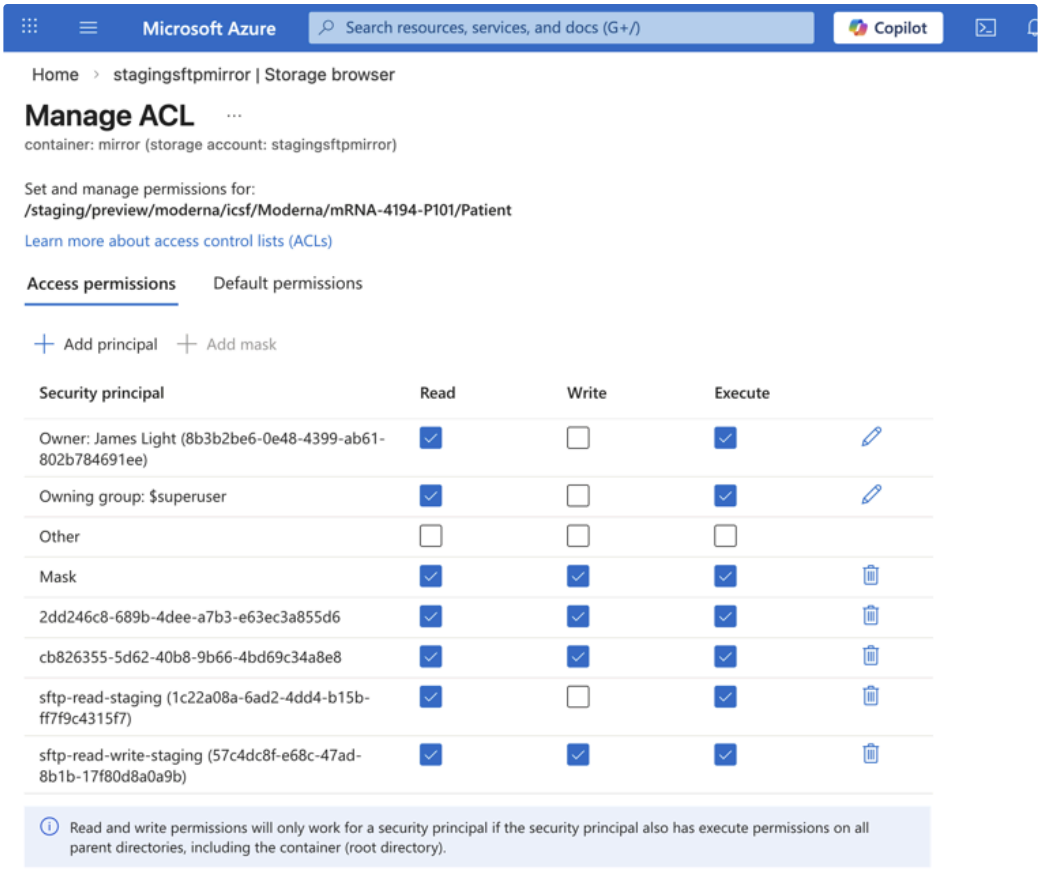
az identity list | \
jq -r '.[] | "\(.name) \(.principalId)"' | \
grep "^${kenv}\-${sbenv}\-" | column -t | \
sort | grep "${study}" | awk '{ print $2 }'

> 169b86ca-6175-4d0b-ba31-1d503ea6e1b1
```

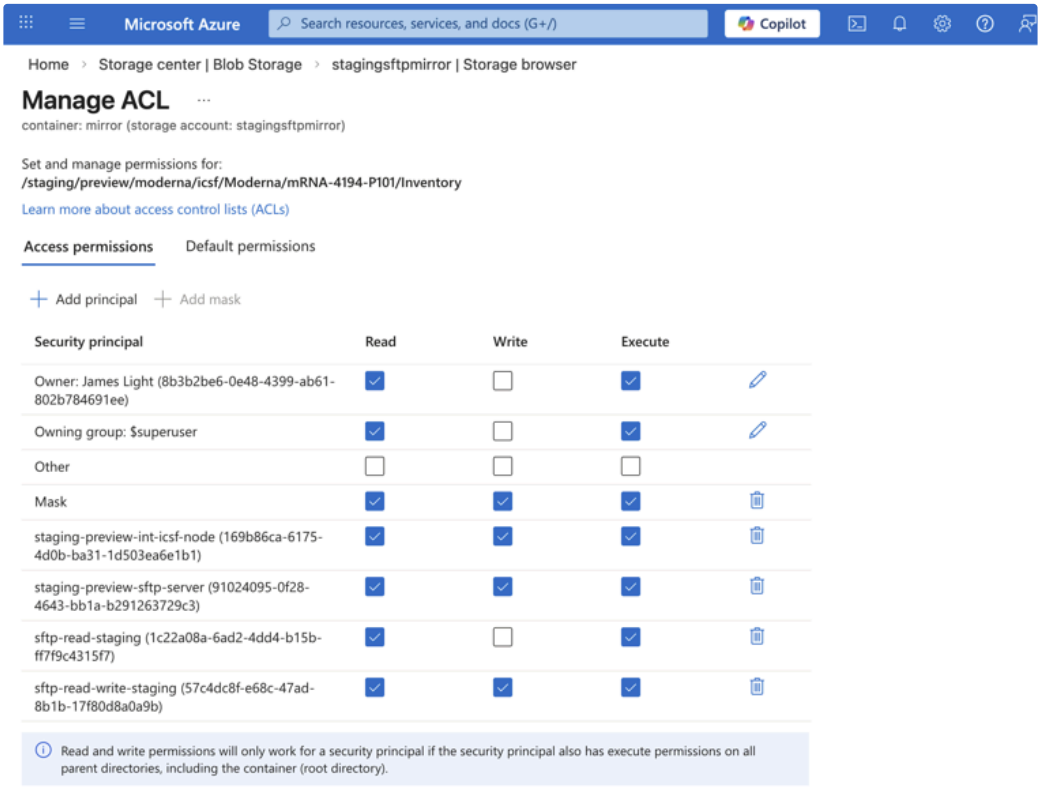
Using the Portal

The Azure Portal Storage Browser can help you verify that you used the correct ID. When you click on Manage ACL, if the ID is NOT a valid principal ID, it will show only the GUID. If the ID is a valid principal ID, it will show the name of the principal, and the GUID in parens.

Here is an example of a case where the incorrect ID, (a **Client ID**) was used in the ACL:



Here is what is looks like with a valid **Principal ID:**



Presto Identity Config

Tell presto to use the new service account and UAMI IDs

FUN FACT - For this use the UAMI's Client ID (not the principal ID).

in `presto-besto-manifesto/staging/presto_conf/.internal/preview/identities.yaml` add an entry to the `identityConfig` dictionary

- First get the UAMI client ID of the client
  - ```
    az identity list | \
    jq -r '.[] | "\(.name) \(.clientId)"' | \
    grep "^${kenv}\-${sbenv}\-" | column -t | sort | \
    grep "${study}"

    > staging-preview-int-icsf-node       2dd246c8-689b-4dee-a7b3-e63ec3a855d6
    ```
- Then update the `identityConfig` dictionary with the corresponding results

  ```
  identityConfig:
    int-icsf-node:
      serviceAccount: staging-preview-int-icsf-node
      workloadId: 2dd246c8-689b-4dee-a7b3-e63ec3a855d6
  ```
  ```