Karlsruher Institut für Technologie

Steinbuch Centre for Computing

Tokens Management in COMPUTE4PUNCH

Inter-TA Technical Meeting

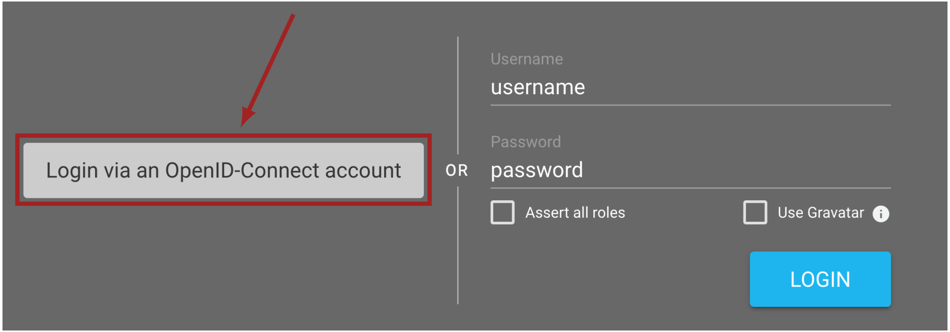
Benoit Roland, Manuel Giffels | 25 January 2023

Introduction

- Tokens management in COMPUTE4PUNCH
- Access to the STORAGE4PUNCH
  - ▸ via Web Interface
  - ▸ via Command Line Interface
  - ▸ in job submission via HTCondor
- Ongoing work to make the token handling in HTCondor transparent to the users

STORAGE4PUNCH Web Interface

- Access to the **STORAGE4PUNCH** granted via the use of **access tokens**
- On the STORAGE4PUNCH web interface, users log in via an **OpenID Provider**



Login via an OpenID-Connect account

OR

Username: username

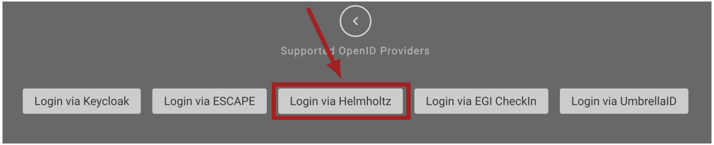
Password: password

□ Assert all roles

□ Use Gravatar

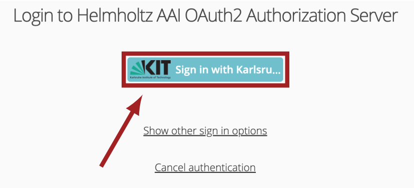
STORAGE4PUNCH Web Interface

- Users choose the **OpenID Provider** for which they own an account . . .



Supported OpenID Providers

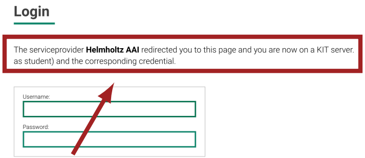
- . . . and are redirected to the corresponding **Authorization Server**



Login to Helmholtz AAI OAuth2 Authorization Server

STORAGE4PUNCH Web Interface

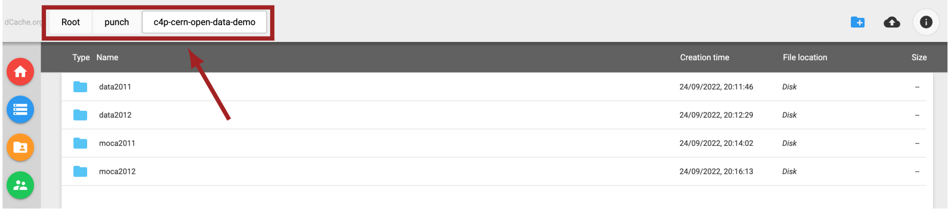
- Users are finally redirected to the **Identity Provider** of their choice . . .



Login

The serviceprovider **Helmholtz AAI** redirected you to this page and you are now on a KIT server.
as student) and the corresponding credential.

- . . . and are **granted access** to the STORAGE4PUNCH **during the lifetime of their access token**



Type	Name	Creation time	File location	Size
	data2011	24/09/2022, 20:11:46	*Disk*	–
	data2012	24/09/2022, 20:12:29	*Disk*	–
	moca2011	24/09/2022, 20:14:02	*Disk*	–
	moca2012	24/09/2022, 20:16:13	*Disk*	–

STORAGE4PUNCH Command Line Interface

- User can be granted access to the **STORAGE4PUNCH** using the **OIDC Command Line Interface**
- After having once **configured the OIDC agent** for the PUNCH AAI provider:

```
$ eval ‘oidc-agent‘
$ oidc-gen --pub punch-aai
```

- The user can **generate an access token and transfer files** to the STORAGE4PUNCH using:

```
$ eval ‘oidc-agent‘
$ oidc-add punch-aai
$ export TOKEN=‘oidc-token -f punch-aai‘
$ curl -L -X PUT -H ‘‘Authorization: Bearer \${TOKEN}’’ --upload-file FILE
     https://dcache-demo.desy.de:2443/punch/mydirectory/FILE
```

COMPUTE4PUNCH credentials and job submission

- COMPUTE4PUNCH uses the **HTCondor batch system** for job submission
  - **Token handling** to access the STORAGE4PUNCH should be **made transparent** to the users
- Once the user has generated a **first access token**, HTCondor should take over:
  - **Secure embedding** of the token into the user job sandbox
  - **Monitoring and refreshment** of the token when its lifetime is about to expire
- HTCondor uses **two components** to accomplish these tasks:
  - **Credentials Daemon credd**
  - **Credentials Monitoring credmon**

COMPUTE4PUNCH credentials management

Credentials Daemon credd

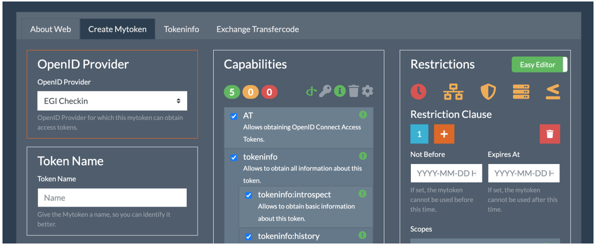
- **Fetches credentials** from secure storage and **pushes them** to the job sandbox
- **Does not care** about credential type, does not access credential content
- **Sends signal** to monitoring component when action is needed
- **Can be used directly** without need for modification

Credentials Monitoring credmon

- In charge of **obtaining** and **manipulating** tokens
- **Monitors** existing tokens and **refreshes** them when needed
- **Specific** to token type, **development needed**

COMPUTE4PUNCH credentials monitoring

- Development available **in personal GitHub repository**
- Introduce new component on top of OIDC provider: **The Mytoken service**
  - Service to obtain OIDC access tokens **for extended periods of time**
  - Users create **Mytokens** instead of access tokens
  - These are used by the credentials monitoring component **to create and refresh access tokens**



OpenID Provider

OpenID Provider: EGI Checkin

OpenID Provider for which this mytoken can obtain access tokens.

Token Name

Token Name: Name

Give the Mytoken a name, so you can identify it better.

Capabilities

☑ AT
Allows obtaining OpenID Connect Access Tokens.

☑ tokeninfo
Allows to obtain all information about this token.

☑ tokeninfo:introspect
Allows to obtain basic information about this token.

☑ tokeninfo:history

Restrictions

Restriction Clause

Not Before: YYYY-MM-DD H

If set, the mytoken cannot be used before this time.

Expires At: YYYY-MM-DD H

If set, the mytoken cannot be used after this time.

Scopes

COMPUTE4PUNCH credentials monitoring

- After having once **configured the OIDC agent** for the PUNCH AAI provider and the Mytoken service:

```
$ oidc-gen --pub --mytoken-url="https://mytoken.data.kit.edu" --issuer="
    https://login.helmholtz.de/oauth2/" --mytoken-profile=agent punch-aai
```

- The user generate a **Mytoken**:

```
$ oidc-token punch-aai --MT
```

- This Mytoken is handled by the credentials monitoring component **without any further user intervention**
  - To obtain access tokens
  - To refresh them when needed

Motivation behind the use of the Mytoken service

- Recommanded by the **Base4NFDI**
- Developed and maintained by **Marcus Hardt** and **Gabriel Zachmann** from the SCC
- Developed in particular to provide OIDC access tokens to long-running compute jobs
- Extensive and friendly **support**
- Extensive **documentation**
- **Command Line** and **Web interfaces** to create new Mytokens and get information about existing ones
- Powerful object allowing **capabilities**, **restrictions** and **rotation**

Mytoken capabilities

- Similar to **scopes** of an **OIDC access token**

- Define **allowed actions** for the Mytoken

  - Capability to **obtain access tokens**: AT

  - Capability to **create new Mytokens**: create_mytoken

  - Capability to **access Mytoken history**: tokeninfo:history

  - Capability to **revoke any Mytoken**: revoke_any_token

  - . . .

Mytoken restrictions

- Limit the power of a Mytoken to the **necessary and sufficient privileges**
  - **Timespan** within which the Mytoken can be used
  - **Scopes** for the requested access tokens: compute, storage.read, storage.write
  - **Audience** defining the accessible resources (https://dcache-demo.desy.de, . . .)
  - **Hosts** from which the Mytoken can be used
  - **IP geolocalisation** to allow or reject access token requests
  - . . .

Mytoken rotation

- **To prevent** illegitimate access to resources
- Every time a Mytoken is used to request a new access token, a **new Mytoken** can also be returned
- Mytokens continuously **exchanged** and **invalidated**
- **Reduce** the possibility to compromise a Mytoken

Summary

- **Access tokens handling** for HTCondor job submission under development
- **Mytoken service** used for access tokens creation and refreshment
- Mytoken service is a **flexible and powerful tool**
- **No further user intervention** after initial Mytoken creation

Special thanks to Gabriel Zachmann and Marcus Hardt for the discussion about the Mytoken service!

Thanks for your attention!