

# Authentication par ID Token

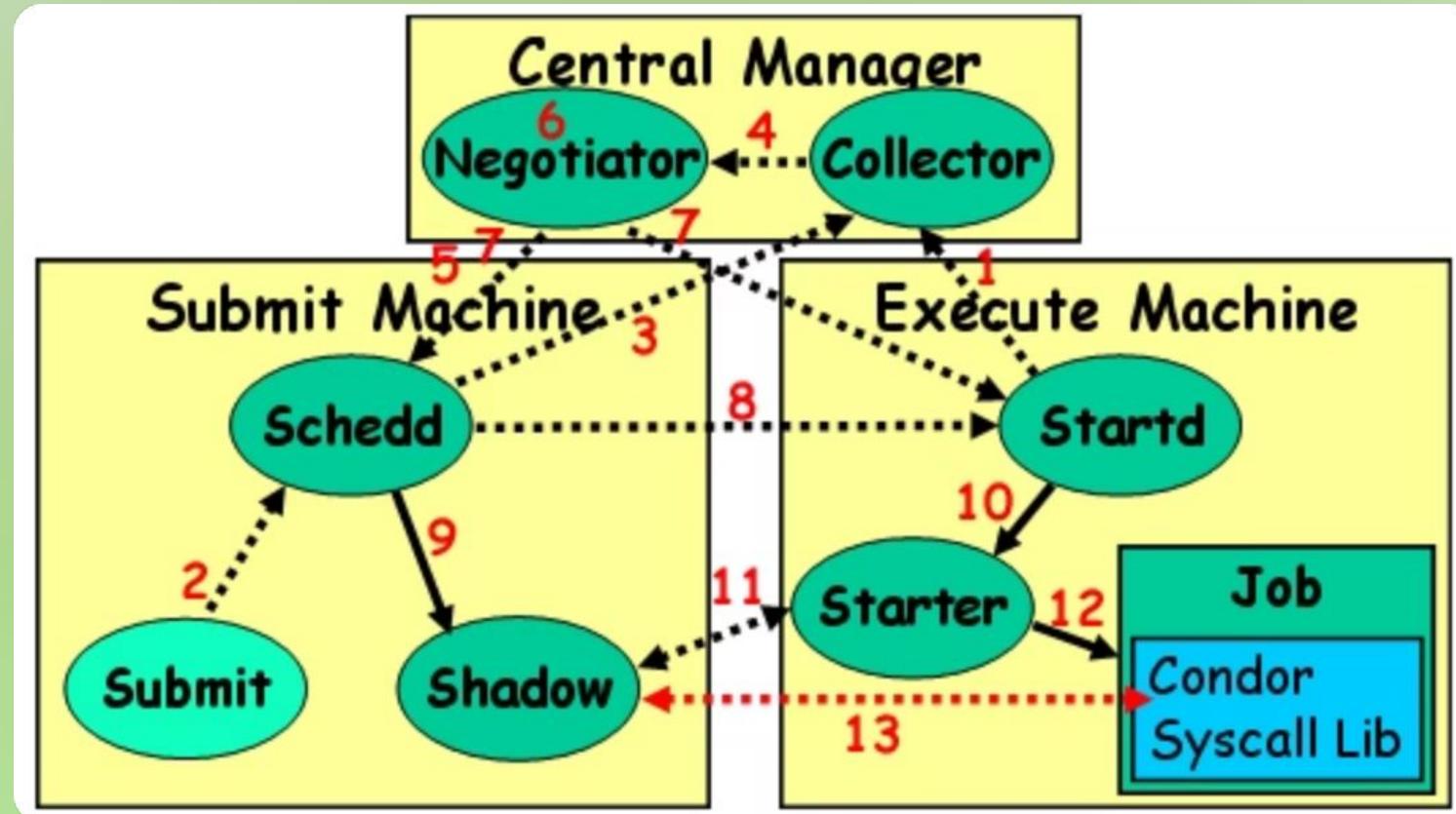
Pool HTCondor

Nouvelle méthode d'authentification pour exécuter des commandes distantes sur le pool HTCondor



# Rappel sur votre environnement de calcul

Des machines avec différents rôles dans le pool HTCondor



# Pourquoi l'authentification par ID Token ?

## Avant : Méthode FS

- Communication limitée aux démons locaux uniquement
- Impossible d'interroger d'autres machines du pool
- Commandes globales non fonctionnelles

## Maintenant : ID Token

- Authentification distante vers tous les services
- Exécution de commandes nécessitant l'accès à plusieurs machines
- Vue d'ensemble de vos jobs sur tout le pool



# Le problème concret résolu

Avant l'implémentation des ID tokens, visualiser l'ensemble de vos jobs soumis depuis différents Access Points était impossible.



## Soumission

Jobs lancés depuis lappui9a et lappui9b



## Blocage

'**condor\_q -global**' échouait systématiquement



## Solution

ID token permet l'accès aux services distants

# Message d'erreur typique

Voici l'erreur rencontrée avant la mise en place des ID tokens lors d'une tentative de commande globale :

```
Error: Couldn't contact the condor_collector
on cm01.tests.local.fr?sock=collector, cm02.tests.local.fr?sock=collector

The condor_collector might not be running, it might be refusing to communicate with you.
```

Cette erreur survenait car les processus ne pouvaient pas interroger les services HTCondor résidant sur d'autres machines du pool.



# Génération d'un ID Token

Un **utilisateur authentifié** est **autonome** pour créer son ID token depuis un serveur l'appui.

01

---

## Commande de création

Lancez : `condor_token_fetch`  
`-token <tokenname>`

02

---

## Fichier généré

Le token est créé avec le nom spécifié

03

---

## Stockage automatique

Placé dans `~/ .condor/tokens.d/`

L'identité renfermée dans le token correspond à celle de l'utilisateur connecté sur la machine où la commande a été exécutée.

# Visualiser vos tokens

## Lister vos tokens

Utilisez la commande :

```
condor_token_list
```

Elle affiche tous vos tokens avec leurs métadonnées, notamment : identité et emplacement du fichier.

## Exemple de sortie

```
Header: {"alg":"HS256","kid":"POOL"}Payload:
{"iat":1758549431, "iss":"tests.local.fr",
"sub":"pierrot@tests.local.fr"}File:
/home1/pierrot/.condor/tokens.d/idtoken
```



# Commande globale : `condor_q -global`

Avec les ID tokens, vous pouvez maintenant visualiser **tous vos jobs**, quel que soit l'Access Point depuis lequel ils ont été soumis.

## Commande

`condor_q -global` depuis n'importe quelle machine UI

## Résultat

Liste complète des jobs sur tous les Schedds du pool : ui03, ce04, ce06, lappui9a, etc.

## Bénéfice

Vision unifiée de l'état de vos soumissions à travers tout le cluster

# Le résultat de commande

```
[pierrot@lappui9b pierrot]$ condor_q -global
-- Schedd: lappthui9d.tests.local.fr : <192.168.10.188:9618?... @ 11/12/25 10:20:31
OWNER BATCH_NAME          SUBMITTED   DONE    RUN    IDLE   HOLD   TOTAL JOB_IDS
Total for query: 0 jobs; 0 completed, 0 removed, 0 idle, 0 running, 0 held, 0 suspended
Total for pierrot: 2 jobs; 0 completed, 0 removed, 0 idle, 2 running, 0 held, 0 suspended
Total for all users: 5 jobs; 0 completed, 0 removed, 0 idle, 2 running, 0 held, 0 suspended

-- Schedd: lappui9a.tests.local.fr : <192.168.10.162:9618?... @ 11/12/25 10:20:31
OWNER  BATCH_NAME      SUBMITTED   DONE    RUN    IDLE   HOLD   TOTAL JOB_IDS
pierrot ID: 13906      10:16      -         -         1         -         1 13906.0

Total for query: 1 jobs; 0 completed, 0 removed, 1 idle, 0 running, 0 held, 0 suspended
Total for pierrot: 1 jobs; 0 completed, 0 removed, 1 idle, 0 running, 0 held, 0 suspended
Total for all users: 1 jobs; 0 completed, 0 removed, 1 idle, 0 running, 0 held, 0 suspended
```

# Analyse détaillée d'un job

Pour diagnostiquer pourquoi un job n'est pas pris en charge ou échoue, utilisez l'option `-better-analyze` :

```
condor_q -global -better-analyze <job_id>
```

## Requirements

Expression des exigences du job  
(GPU, mémoire, CPU)

## Matching

Analyse des machines  
correspondantes et raisons de  
rejet

## Diagnostic

Explication précise des échecs de  
correspondance

L'option `-global` permet d'analyser un job même s'il a été lancé depuis une autre machine.

## Type de sortie obtenue pour un job



```
-- Schedd: lappui9f.tests.local.fr : <192.168.10.175:9618?...
The Requirements expression for job 41579.000 is
  ((regex(".*k80.*",TARGET.GPU_MODEL,"i")) ||
   (regex(".*p6000.*",TARGET.GPU_MODEL,"i")) ||
   (regex(".*1g.5gb.*",TARGET.GPU_MODEL,"i"))) && GpuUsageRequirements

Job 41579.000 defines the following attributes:
  FileSystemDomain = "condor.must.tests.local.fr"
  GpuUsageRequirements = (Machine != LastRemoteHost) && (TARGET.Arch == "X86_64") && (TARGET.OpSys == "LINUX") && (TARGET.Disk >=
RequestDisk) && (TARGET.Memory >= RequestMemory) && (TARGET.Cpus >= RequestCpus) && (TARGET.GPUs >= RequestGPUs)
  RequestCpus = 8
  RequestDisk = 102400 (kb)
  RequestGPUs = 1
  RequestMemory = 8192 (mb)

The Requirements expression for job 41579.000 reduces to these conditions:
      Slots
Step   Matched  Condition
-----
[1]      14  regex(".*p6000.*",TARGET.GPU_MODEL,"i")
[5]       3  GpuUsageRequirements
[6]       0  [1] && [5]

No successful match recorded.
Last failed match: Fri Sep 19 16:50:31 2025

Reason for last match failure: no match found
```

# Obtenir des informations sur le statut du pool HTCondor

Grâce aux ID Tokens, la commande `condor_status` offre désormais une visibilité complète sur l'état du pool, incluant les noms des machines GPU, leurs caractéristiques et leur taux d'occupation.

Par exemple, pour lister tous les serveurs équipés de cartes graphiques (GPU) et le détail de leur disponibilité, utilisez la commande suivante :



```
[pierrot@lappui9a ~]$ condor_status -gpu
```

Name	ST	User	GPUs	GPU-Memory	GPU-Name
slot1@wngpu004.local.fr	Ui	_	1	15.8 GB	Tesla V100-PCIE-16GB
slot1@wngpu005.local.fr	Ui	_	1	23.9 GB	Quadro P6000
slot1@wngpu006.local.fr	Ui	_	4	14.6 GB	Tesla T4
slot1@wngpu008.local.fr	Ui	_	0	39.5 GB	NVIDIA A100-PCIE-40GB
slot1_1@wngpu008.local.fr	Cb	thiboup@local.fr	1	39.5 GB	NVIDIA A100-PCIE-40GB
slot1_2@wngpu008.local.fr	Cb	lafloud@local.fr	1	39.5 GB	NVIDIA A100-PCIE-40GB
slot1_3@wngpu008.local.fr	Cb	lafloud@local.fr	1	39.5 GB	NVIDIA A100-PCIE-40GB

Pour des informations détaillées sur une machine spécifique, utilisez `condor_status -l <nom_machine>`.

# Sécurité des ID Tokens

Un ID token équivaut à votre mot de passe : il permet d'exécuter des commandes sous votre identité.

1

## Protection par défaut

Stockage sécurisé dans votre répertoire personnel avec permissions restrictives

`(-rw-----)`

2

## Accès limité

Vous êtes propriétaire et seul lecteur autorisé du fichier token

3

## Ne pas modifier

Conservez les autorisations par défaut pour maintenir la sécurité



# Bonnes pratiques

1

## Régénération régulière des clés

La clé de signature des tokens est régénérée périodiquement, invalidant les anciens tokens pour renforcer la sécurité.

2

## Suppression en fin de session

Supprimez vos ID tokens lorsque vous terminez votre travail pour éviter les problèmes d'authentification.

3

## Création en début de session

Générez un nouveau token au début de chaque session pour garantir sa validité et éviter les messages d'erreur.