

# HTCondor 9.0 For Admins

# HTCondor 9.0 For Admins

Why users will be begging you to upgrade...

# 9.0 is big release!

Biggest NEWS: Secure by default!

Upgrade probably breaks existing config

You have 3 options:

# 3 options

1. Return to existing host-based security
2. Reinstall from scratch  
Todd 2-minute drill!
3. Configure IDToken based security  
Not as hard as you might think

# Release notes has details

- Miscellaneous Concepts
  - Grid Computing
  - Cloud Computing
- Application Programming Interfaces (APIs)
- Platform-Specific Information
- Frequently Asked Questions (FAQ)
- Contrib and Source Modules
- Version History and Release Notes
  - Introduction to HTCondor Versions
  - Upgrading from the 8.8 series to the 9.0 series of HTCondor
    - Step 1
    - Step 2
    - Step 3
    - Step 4
    - New Features (other than security)
    - Development Release Series 9.1
    - Stable Release Series 9.0
    - Development Release Series 8.9
    - Stable Release Series 8.8
- Command Reference Manual (man pages)
- APPENDIX
  - ClassAd Attributes
  - Codes and Other Needed Values
  - Index

Read the Docs  
<https://htcondor.readthedocs.io/en/latest/apis/index.html>

```
# condor_store_cred -c add  
# umask 0077; condor_token_create -identity condor@mypool > /etc/condor/tokens.d/condor@mypool
```

Continue with [Step 4](#) below.

**Option C.** Revert to the previous host-based security configuration that was the default before v9.0. This is the most expedient way to get your pool running again as it did before upgrading, but realize that a host-based security model is not recommended. If you go for this option, please consider it a temporary measure. To configure HTCondor to function as it did before the upgrade, see the instructions in [/etc/condor/config.d/00-htcondor-9.0.config](#).

## Step 2

If you chose **option A** or **option B** in step 1, skip this step.

If you did not previously set `ALLOW_DAEMON` explicitly, you will now need to do so. To duplicate the 8.8 behavior, set `ALLOW_DAEMON = $CONDOR_MACHINE`.

## Step 3

If you chose **option A** or **option B** in step 1, skip this step.

The deprecated configuration settings beginning with `HOSTALLOW` and `HOSTDENY` have been removed. If your 8.8 configuration was still using either, add their entries to the corresponding `ALLOW` or `DENY` list.

If you run the `condor_check_config` tool it will detect a couple of the most common configuration values that should be changed after an upgrade.

## Step 4

The following changes may affect but your pool, but are not security improvements. You may have to take action to continue using certain features of HTCondor. If you don't use the feature, you may

<http://htcondor.readthedocs.io>

<https://htcondor.readthedocs.io/en/latest/version-history/upgrading-from-88-to-90-series.html>

# Install from scratch is easy!

New script/service: "get\_htcondor"

Double check with Todd's talk!

still have rpms, docker images, python,  
etc.

# IDTokens?

New security method for HTCondor

Future of HTCondor security (we hope)

capability based

fine-grained

no 3<sup>rd</sup> party service needed

# SCITOKENS

For security to non-HTCondor daemons

e.g authenticating to xrootd or other storage

*Jobs can get files w/o sending password*

Emerging standard for scientific data access

# OAUTH2

Allows job to get/put data directly from  
box, google drive, etc

**WITHOUT PASSWORD IN CLEAR!**

Storage becomes someone else's problem!\*

\*(networking may be another story)

# Admins need to configure the credd

1. USE FEATURE: oauth
2. Setup web app
3. Register the schedd on the service

<http://htcondor.readthedocs.io>

# Oauth submit file

```
use_oauth_services = cloudboxdrive  
cloudboxdrive_oauth_permissions = read:/public
```

condor\_submit will wait and print:

Please to go to [http://cloudboxdrive.io/big\\_long\\_url](http://cloudboxdrive.io/big_long_url)

After that, condor has your credentials, and sends with job

# Yet more security

AES on by default, all wire writes encrypted (!)

- Hopefully, little performance impact

FIPS mode by default

- (no more MD5)
- No more separate FIPS release

# Submitter ceilings

```
$ condor_userprio -setceiling toddt 5
```

Sets a per-user limit of 5 cores per pool for toddt

Don't need to create groups just for limits

# THIS IS BIG!

# Why submitter ceilings?

Separate *priority* from *share*

e.g new user gets first dibs on few machines

If over ceiling, no preemption/eviction by default

# DAGMan throttles

Can now condor\_qedit a running dagman

And change:

MaxJobs,

MaxIdle

MaxPreScripts

MaxPostScripts

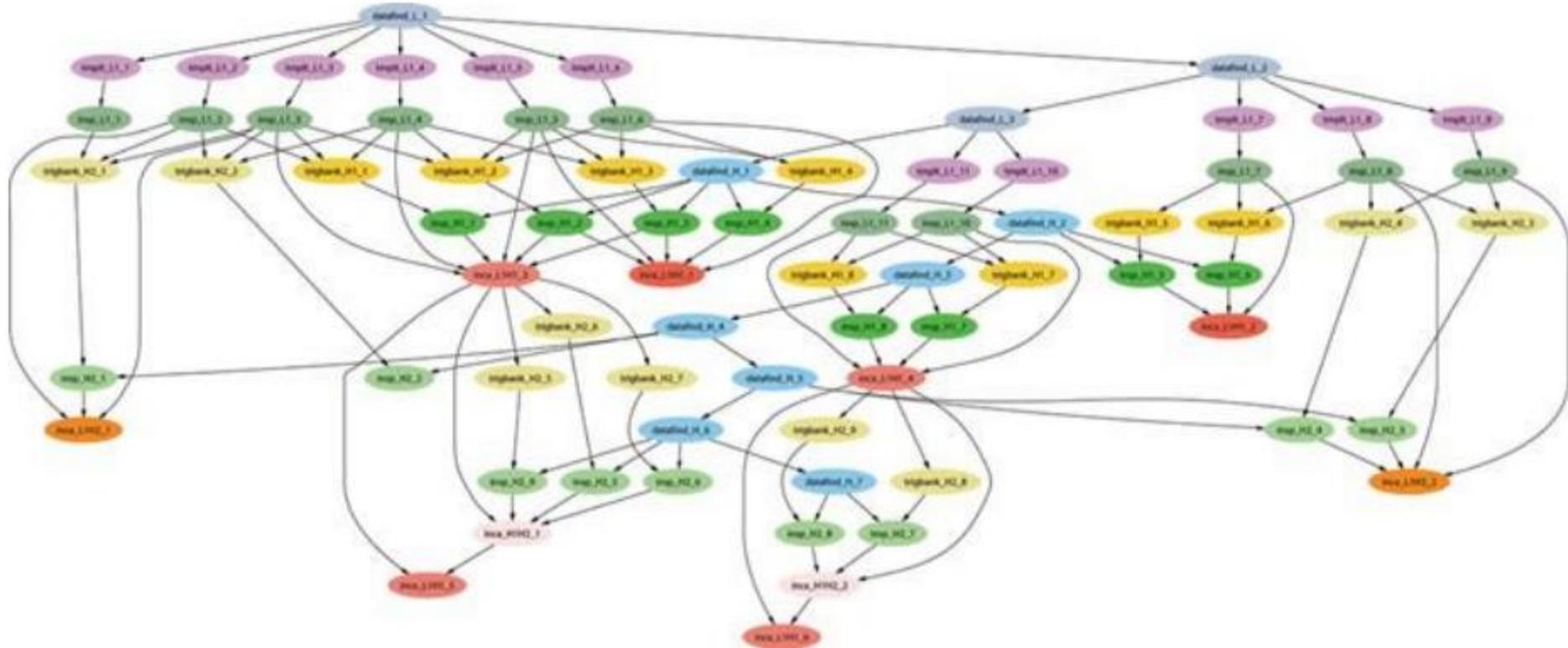
# More dagman

DAGMan provisioner nodes for cloud

condor\_dagman -dot just prints the dag

dot command line tool turns into an image

# Example dot output



# The little things...

More performance improvements  
especially in negotiator

More static/dynamic analysis of code  
fewer memory leaks, security issues

Python3 support in the bindings  
sic transit gloria python2

# Tools for debugging

`classad_eval`

You can test classads before deployment

```
$ classad_eval '9 * 8'  
[ ]  
72
```

```
$ classad_eval 's = "foo"' 'regexp(s, "foo")'  
[ ]  
true
```

# The not-so-little things

- › New platforms:
  - Enterprise Linux 8, Ubuntu 20
- › Old platforms removed
  - Enterprise Linux 6

# Packaging

- › All kinds of places to get HTCondor
  - New native repos for devel, stable and nightly
  - Docker hub has images
  - conda

# condor\_adstash - Intro

Pushes job history ads (from schedds and/or startds) to Elasticsearch

Requires Python 3.6+ with HTCondor Python Bindings and Python Elasticsearch Client (e.g. `pip install htcondor elasticsearch`)

Either run under `condor_master` with `use feature: adstash` in config or run standalone with `condor_adstash --standalone [...]`

Example config in `condor_config.local.adstash` (see `/usr/share/doc/condor-NNN/examples/` or GitHub)

Defaults to pushing ads to `htcondor-000001` index in ES, up to you how to configure and manage index lifecycle

# condor\_adstash – Example

“What’s causing jobs to run long?”  
Query finished job ads using Kibana

The screenshot shows the Kibana search interface. At the top, there's a search bar with the query: `RemoteWallClockTime > 86400 and CompletionDate > 1621054800 and ScheddName: login04.osgcor`. Below the search bar, there are several filters and a table of results. The table has 5 columns: `RemoteWallClockTime`, `MachineAttrMips0`, `MachineAttrName0`, `RequestCpus`, and `RequestMemory`. There are 2,181 hits in total.

Discover

New Save Open Share Inspect

RemoteWallClockTime > 86400 and CompletionDate > 1621054800 and ScheddName: login04.osgcor KQL Refresh

+ Add filter

osg-schedd-\*

Search field name

Filter by ... 0

Selected fields

- # Mach...
- # Mach...
- # Remo...
- # Requ...

| RemoteWallClockTime | MachineAttrMips0 | MachineAttrName0                                       | RequestCpus | RequestMemory |
|---------------------|------------------|--|-------------|---------------|
| > 609,185           | 25,123           | slot1@NDSU-Lancium-Backfill-7a1d94a63aa8               | 1           | 500           |
| > 457,951           | 20,518           | slot1_2@glidein_11261_289360996@mole087.beocat.ksu.edu | 1           | 250           |
| > 593,797           | 25,205           | slot1@NDSU-Lancium-Backfill-fbaecfee39c9               | 1           | 500           |
| > 146,040           | 25,327           | slot1_4@glidein_16951_77405811@uct2-c549.mwt2.org      | 1           | 512           |

# condor\_adstash – Example

“How did the pool look last week?”

Python script that queries ES and sends email

OSPool per Schedd usage for jobs completed from 2021-05-10 to 2021-05-17

|   | Schedd                 | All CPU Hours | % Good CPU Hours | CPU Hours / Bad Exec Att | Num Uniq Job Ids | Num DAG Node Jobs | Num Jobs w/>1 Exec Att | Shadw Starts / Job Id | Exec Atts / Shadw Start | Num Rm'd Jobs | Avg MB Sent | Max MB Sent | Avg MB Recv |
|---|------------------------|---------------|------------------|--------------------------|------------------|-------------------|------------------------|-----------------------|-------------------------|---------------|-------------|-------------|-------------|
| 9 | TOTAL                  | 2,405,352     | 58.2             | 3.7                      | 1,303,506        | 617,103           | 109,100                | 1.21                  | 0.897                   | 92,842        | 9           | 7,429       | 8           |
| 1 | login04.osgconnect.net | 1,109,376     | 65.8             | 2.6                      | 613,177          | 274,786           | 50,667                 | 1.18                  | 0.957                   | 47,602        | 3           | 286         | 8           |
| 2 | login05.osgconnect.net | 740,728       | 43.1             | 5.7                      | 386,423          | 160,155           | 32,780                 | 1.22                  | 0.806                   | 39,379        | 0           | 3,097       | 10          |

# New transform language

- › Schedd Transforms: An Admin's Chainsaw
- › Schedd job xforms easier to write,
  - Now with defined order of transforms!

<http://htcondor.readthedocs.io>

# Why Schedd transforms?

- › Admin wants to enforce policy at APs
  - e.g every RayTrace job requests disk
- › Admin wants to enforce job attributes
  - e.g. we will assign some group to every job

# Testing transforms

- `condor_transform_ads`

- stand alone tool for testing a single transform
- `-verbose` mode logs the transform steps

```
condor_transform_ads -rules <xfm> -in <in>  
condor_transform_ads -help rules
```

# Drain -reason

- › Condor\_drain takes a -reason
- › Defrag passes -reason defrag
- › Defrag ignores drain for other reasons

# Dataflow mode for jobs

Jobs are like Make:

When `SHADOW_SKIP_DATAFLOW_JOBS = true`

1. If pre-declared output files already exist AND
2. output files are more recent than input files

**JOB IS SKIPPED!**

# Container improvements

- › Containers now honor `request_gpus`
- › Port forwarding in a docker container
- › `docker_network = host // none`
- › Docker & singularity test before starting
- › Cgroups now sets both soft & hard mem limits

# Free as in Beer

[github.com/htcondor/htcondor/docs](https://github.com/htcondor/htcondor/docs)

Free beverage to first reasonable doc PR\*

(\*PATH employees not eligible. Some restrictions apply)

# Conclusion

Thank you, this is just some of the big things..