Git and HTCondor

Integrating Configuration with Version Control

Erin Grasmick

May 17, 2016

Center for High Throughput Computing

Why?

- version control
- close eye on your contributors
- collaboration
- branches to test changes
- removes layer of complexity
- efficiency (diffs)

- puppet
- distributed filesystem
- http
- ????
- …and you're probably managing these with git anyway

What problems does this solve?

- condor_reconfig integration
- monitoring
- configuration version analysis
- ensures homogeneity
- allows configuration version tracking (git HEAD)

For something like HTCondor, we want to see our changes immediately.

We need a smart, secure, access-controlled repository.

gitolite.

A repo accessible using ssh with authorized keys, which has its own configuration.

You could use Github or any git server with access control for this too!

```
git clone gitolite@gitserver:htcondor-config
```

You're halfway done!

Lots of documentation - included in official git docs.

Repository Security

group of repos:

```
repo @oss_repos
        RW int$              = @interns
        RW eng-              = @engineers
        RW refs/tags/rc[0-9] = @engineers
        RW+                  = @admins
```

singular repo:

```
repo foo_bar
        RW+                     = alice @teamleads
        -  master               = bob @devteam
        -  refs/tags/v[0-9]      = bob @devteam
        RW+ dev/                 = bob @devteam
        RW                       = bob @devteam
        R                        = @managers
```

No execute nodes are allowed to push configuration changes.

HTCondor Integration

- executes command and evaluates stdout in config
- published master ad attribute
- MASTER_ATTRS = $(MASTER_ATTRS) GITHEAD
- run as master to check GITHEAD otherwise the nature of the macro language will cause the script to run every time you use condor_config_val, which IsTool rather than IsMaster

```
condor_config_val -name mynode.chtc.wisc.edu GITHEAD
```

HTCondor configuration:

```
ETC   = /etc/condor
# Host-specific LOCAL file
LOCAL = $(ETC)/hosts/$(FULL_HOSTNAME).local

if $(IsMaster)
    include command : /usr/bin/sudo $(ETC)/bin/htcondorconfig_pull.sh -d $(
        ETC)
endif

REQUIRE_LOCAL_CONFIG_FILE = false
LOCAL_CONFIG_FILE = $(HOSTS)    \
                    $(GLOBAL)   \
                    $(POLICY)   \
                    $(PLATFORM) \
                    $(SECURITY) \
                    $(LOCAL)
```

Host-specific "LOCAL" file
/etc/condor/hosts/mynode.chtc.wisc.edu.local:

```
include : $(ETC)/include/startd.conf
include : $(ETC)/include/ckpt_server.conf
include : $(ETC)/include/facility_cs_3370a.conf
include : $(ETC)/include/owned_by_chtc.conf
```

Checking for compliance:

```
$ condor_status -master -const '! IsUndefined(GitHead)' &&
    GitHead =!= "bcfd89a199ddbd050ce6f9fbe63760fe2136488c"' -af
    githead name

7853086710d87dbbb945d379d8a05f3c69090d80 a004.chtc.wisc.edu
7853086710d87dbbb945d379d8a05f3c69090d80 a005.chtc.wisc.edu
7853086710d87dbbb945d379d8a05f3c69090d80 a006.chtc.wisc.edu
7853086710d87dbbb945d379d8a05f3c69090d80 a011.chtc.wisc.edu
```

Sudo Configuration

Allow condor user to run only only the configuration-pulling script as root!

```
user@host ~ $ cat /etc/sudoers.d/htcondor_config_pull
condor ALL=(root) NOPASSWD: /etc/condor/bin/htcondor_config_pull.sh
Defaults!/etc/condor/bin/htcondor_config_pull.sh !requiretty
```

HTCondor config permissions should be root:root, and writeable only by owner.

- logs to syslog
- very basic: performs clean, reset, pull
- staggers reconfig to avoid overload
- staggering will be built into condor_reconfig in the future

Questions?

http://cs.wisc.edu/~egrasmic/condor_config

http://cs.wisc.edu/~egrasmic/htcondor_config_pull.sh

Contact:

egrasmick@wisc.edu