

HTCondor deployment using puppet

Alessandro Italiano

Agenda

- *HTCondor configuration structure*
- *Puppet introduction*
- *Puppet module to deploy HTCondor*

installation

- *In a linux environment it is a quick step, just install the repo for your distribution and then install htcondor*

Configuration

- *Each HTCondor program configure itself parsing the various configuration files that might be used. Environment variables may also contribute to the configuration.*
- *The order in which attributes are defined is important, as later definitions override existing definitions*

Configuration order

The order in which items are parsed is:

- 1. Condor global configuration file*
- 2. Condor local files*
- 3. ENVIRONMENTS variables*

Configuration order: global configuration file

- *CONDOR_CONFIG = /etc/condor/condor_config*
- *if not exported in the bash environment it is defined in the init script*
- *general configuration*
- *can be shared between all the servers belong to the cluster*

Configuration order: local configuration file

- *It can be enabled in the CONFIG_FILE*
- *LOCAL_CONFIG_FILE*
 - *lists one or more configuration files. The leftmost (first) in the list is parsed first.*
- *LOCAL_CONFIG_DIR*
 - *lists one or more directories. The leftmost (first) in the list is parsed first*
 - *lexicographical ordering by file name determines the ordering of file consideration*
 - *“computer” is lexicographical first than “computing”*
- *LOCAL_CONFIG_DIR_EXCLUDE_REGEXP* *in order to exclude some files*

Configuration order: environment variables

- *prefixed with **"_CONDOR_"***
- *Once the condor program find them, the prefix is striped off and what remains is used as configuration*
- *For security reason not any environment variables with the prefix will be considered*

Configuration macros

<macro_name> = <macro_definition>

- *macro_name is case insensitive*
- *white space can be omitted*
- *macro_definition is a string which can be a macro substitution*
- *more than 5000 macros well know as KNOBS*

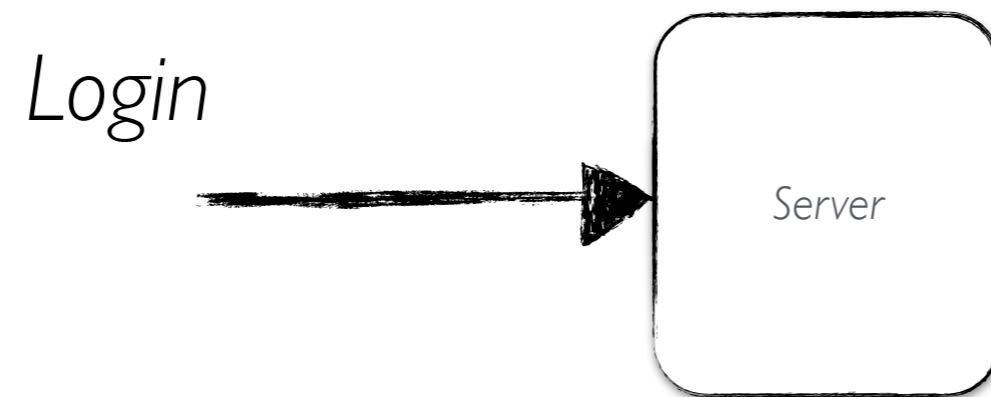
Configuration macros example

- `HAD_PORT = 51450`
- `DAEMON_LIST = MASTER, COLLECTOR, NEGOTIATOR, HAD, REPLICATION`
- `COLLECTOR_HOST = $(CENTRAL_MANAGER1), $(CENTRAL_MANAGER2)`
- `START = (KeyboardIdle > 15 * $(MINUTE)) && \`
`((LoadAvg - CondorLoadAvg) <= 0.3)`

configuration conclusion

- *HTCondor is a highly distributed framework. Several components running on different hosts*
- *At the same time we have all the features to configure HTCondor in a flexible way:*
 - *files order, macro substitution, SUBSYSTEM macro*
- *Flexibility can lead to complexity*
- *can we really use a storage file system to deploy our configuration ?*

Manual deployment



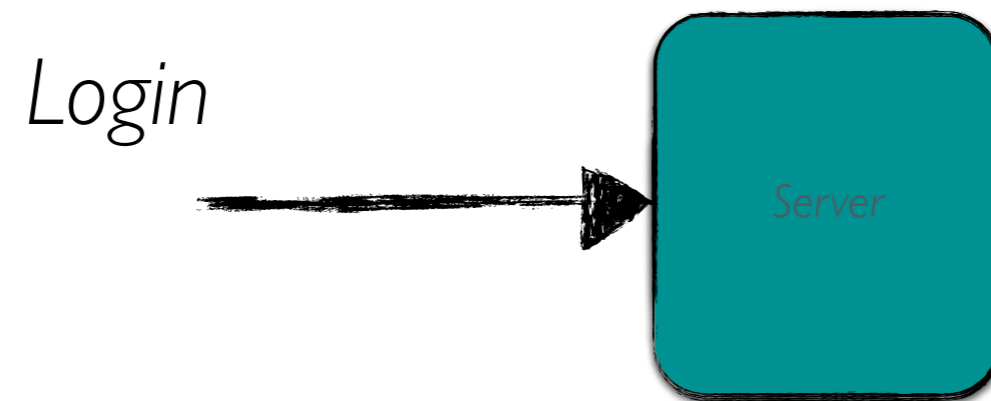
Manual deployment

Login



- install
- conf2
- conf3
- restart

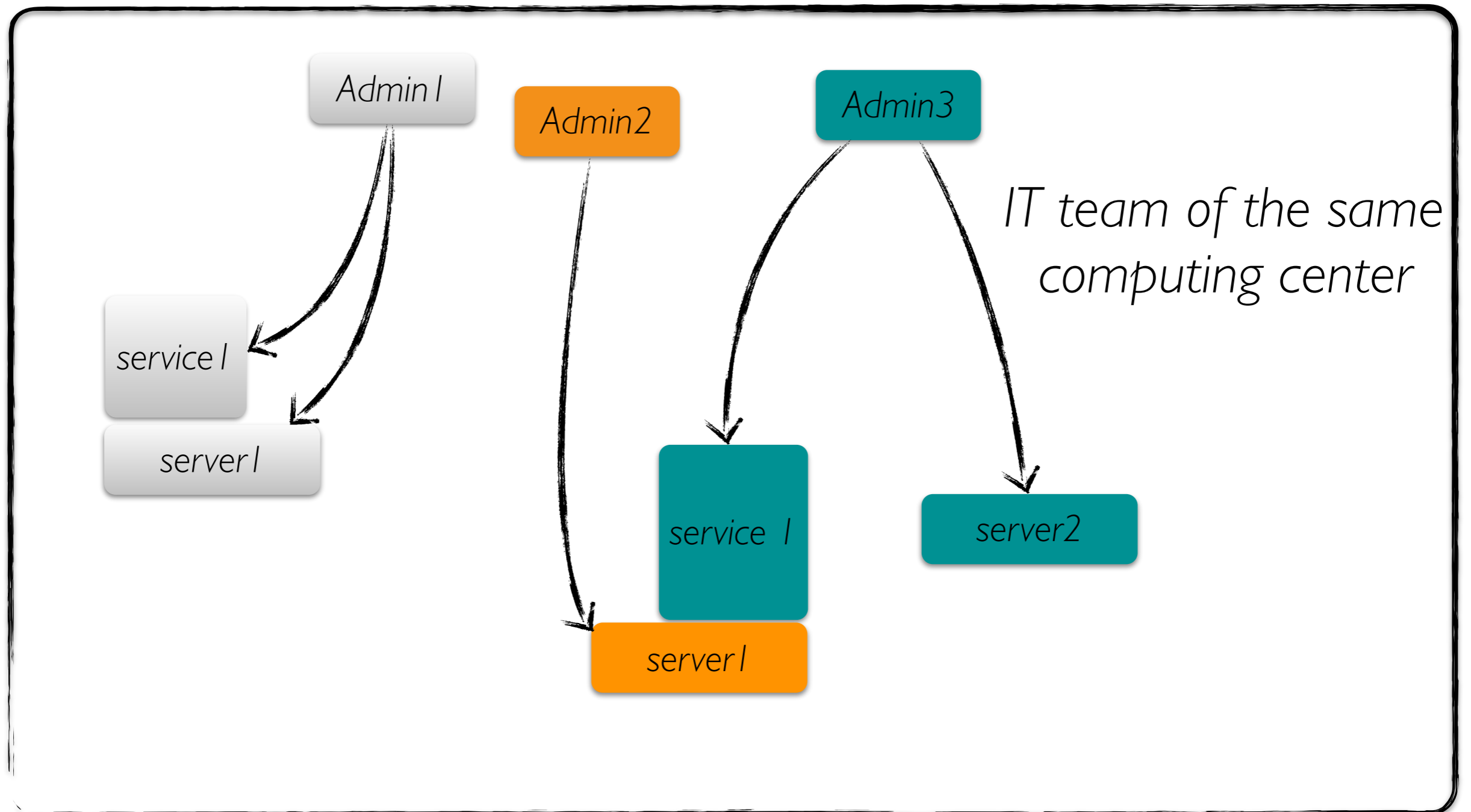
Manual deployment



Manual deployment

- *Two main issues*
 - *Repetitive tasks: waste time*
 - *Host can have inconsistent state due to*
 - *reinstallation*
 - *manual change*

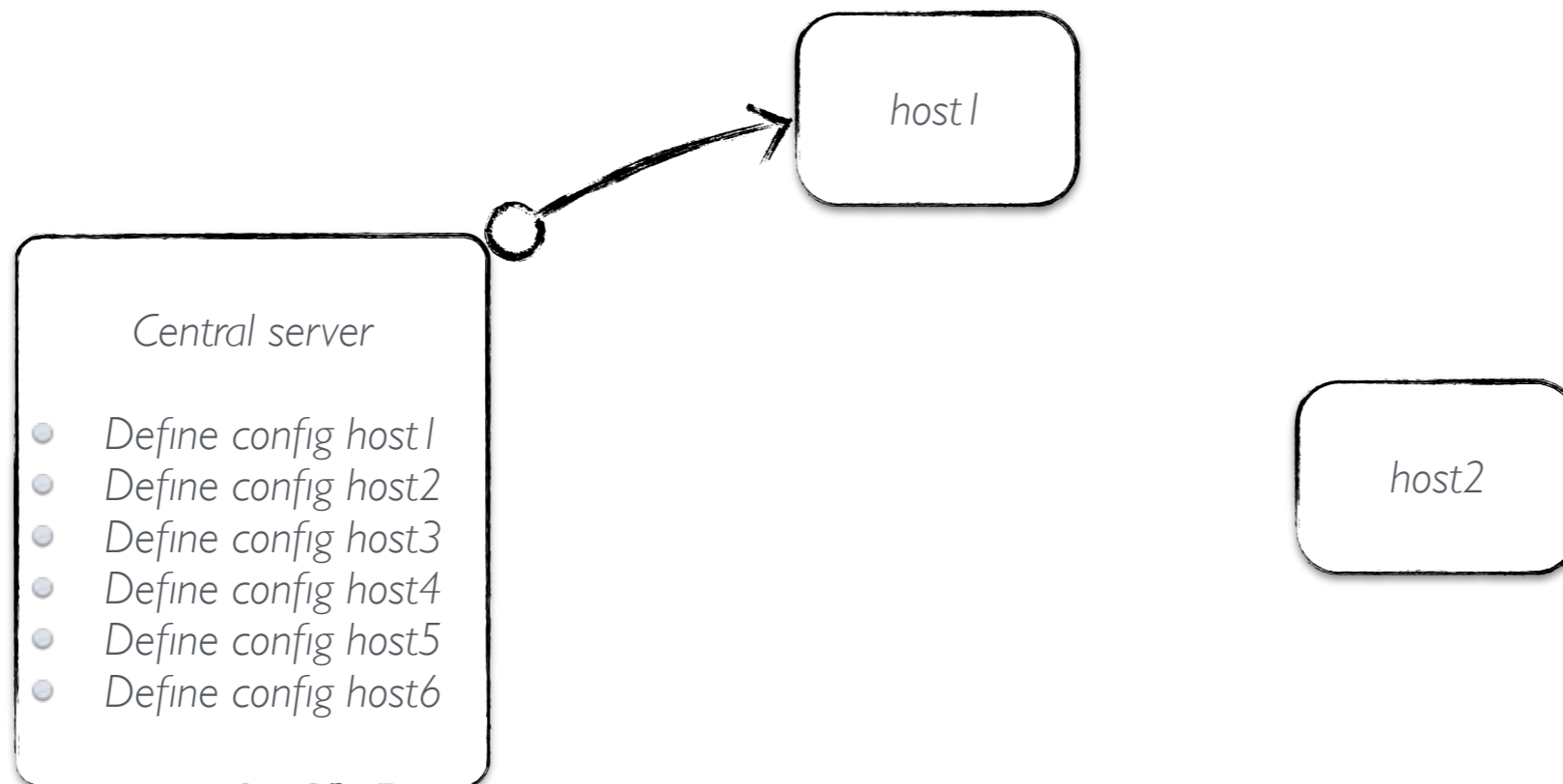
Manual deployment, the common habits



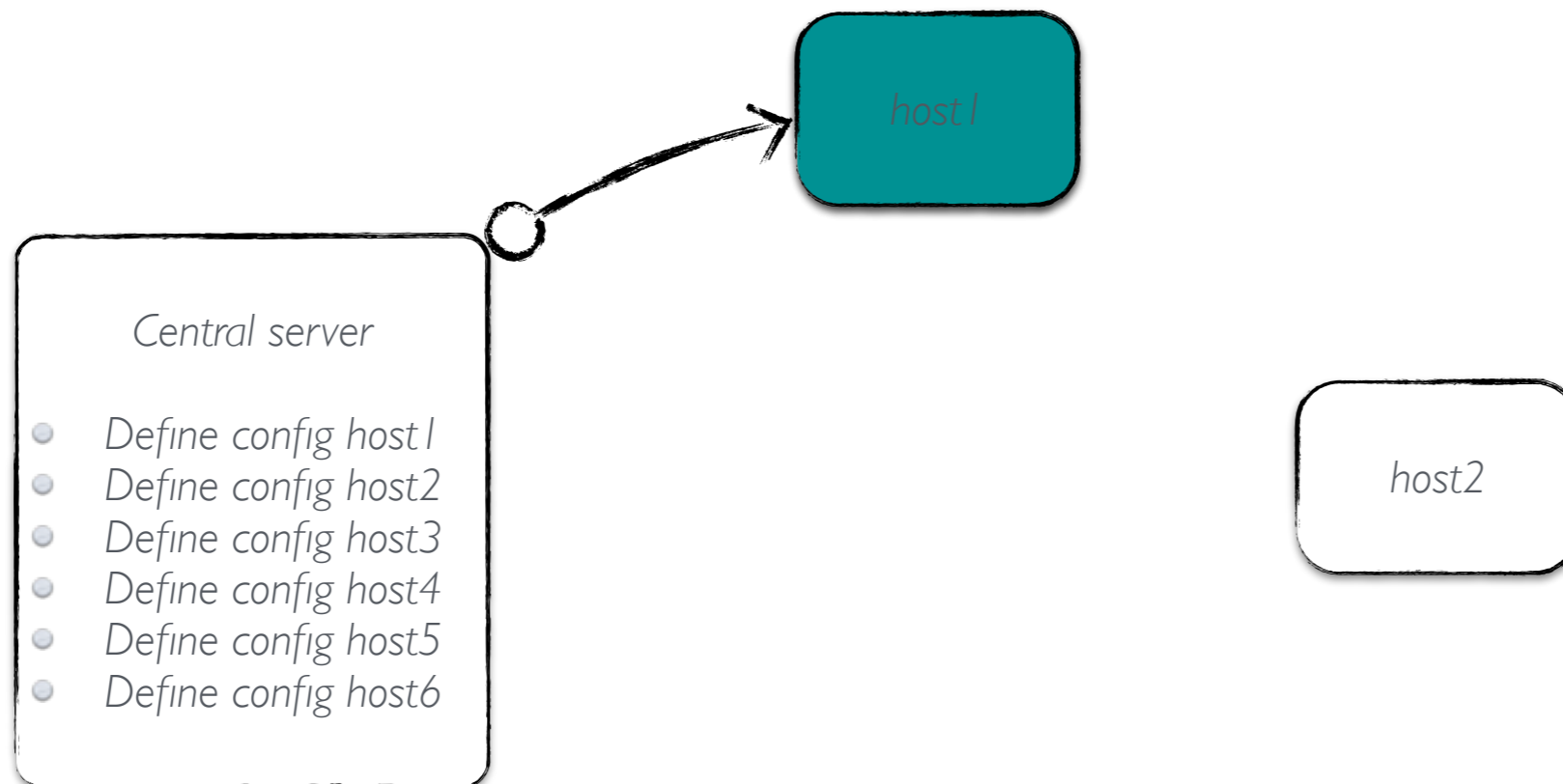
Automate deployment



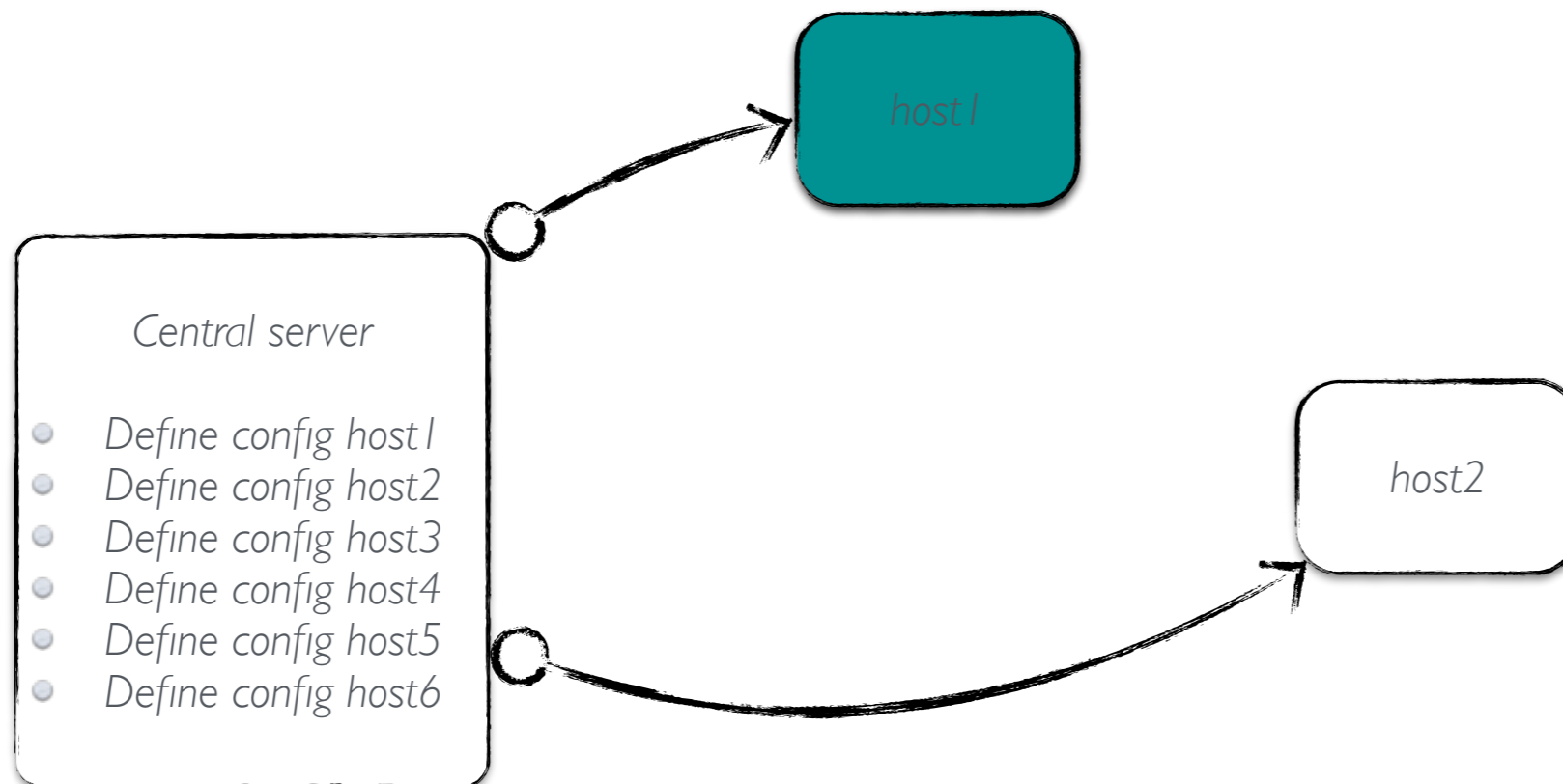
Automate deployment



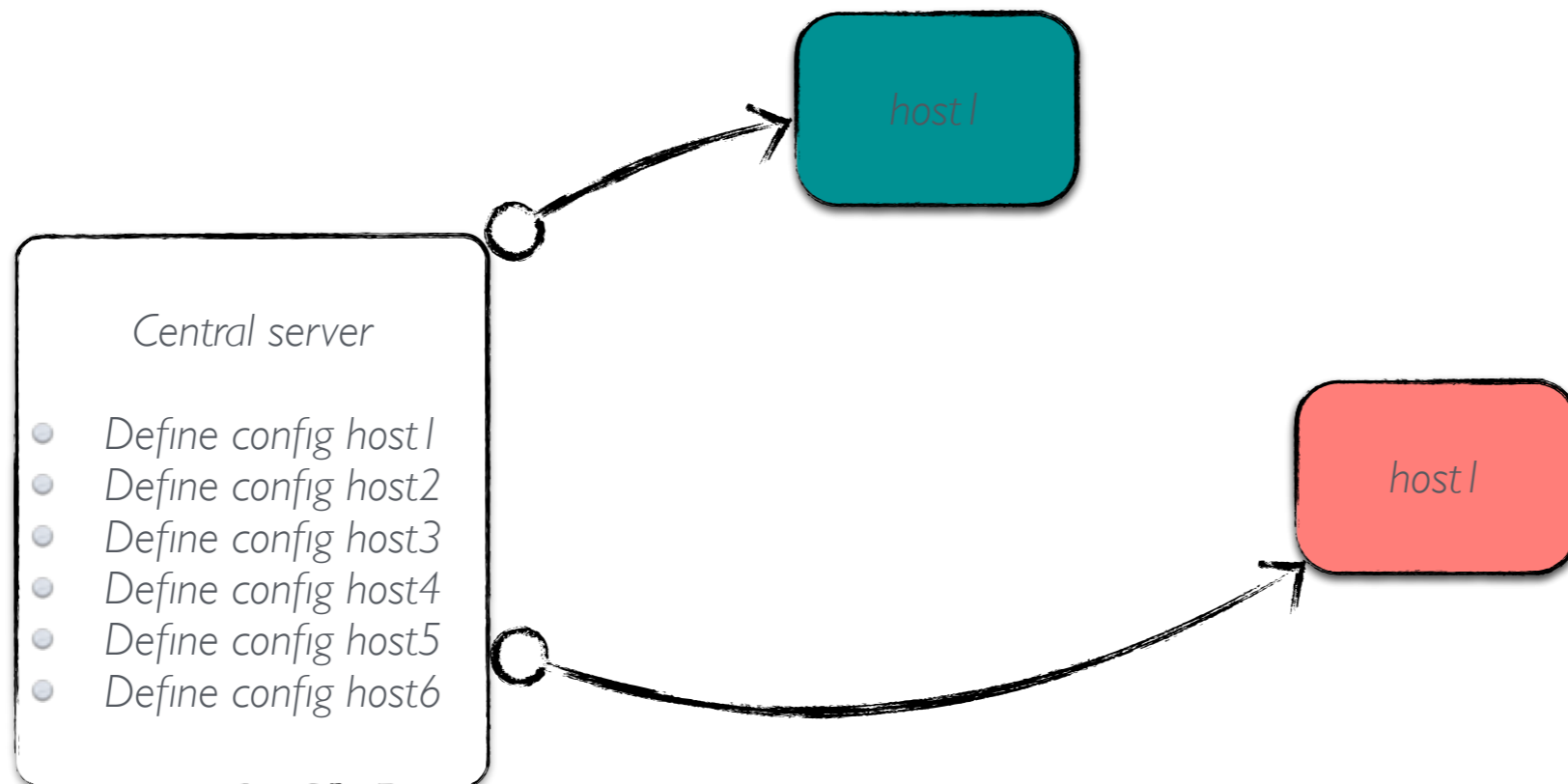
Automate deployment



Automate deployment



Automate deployment



Automate, why it is really important

Save time avoiding repetitive action

Synchronisation

Replication

Optimisation

IT infrastructure under control

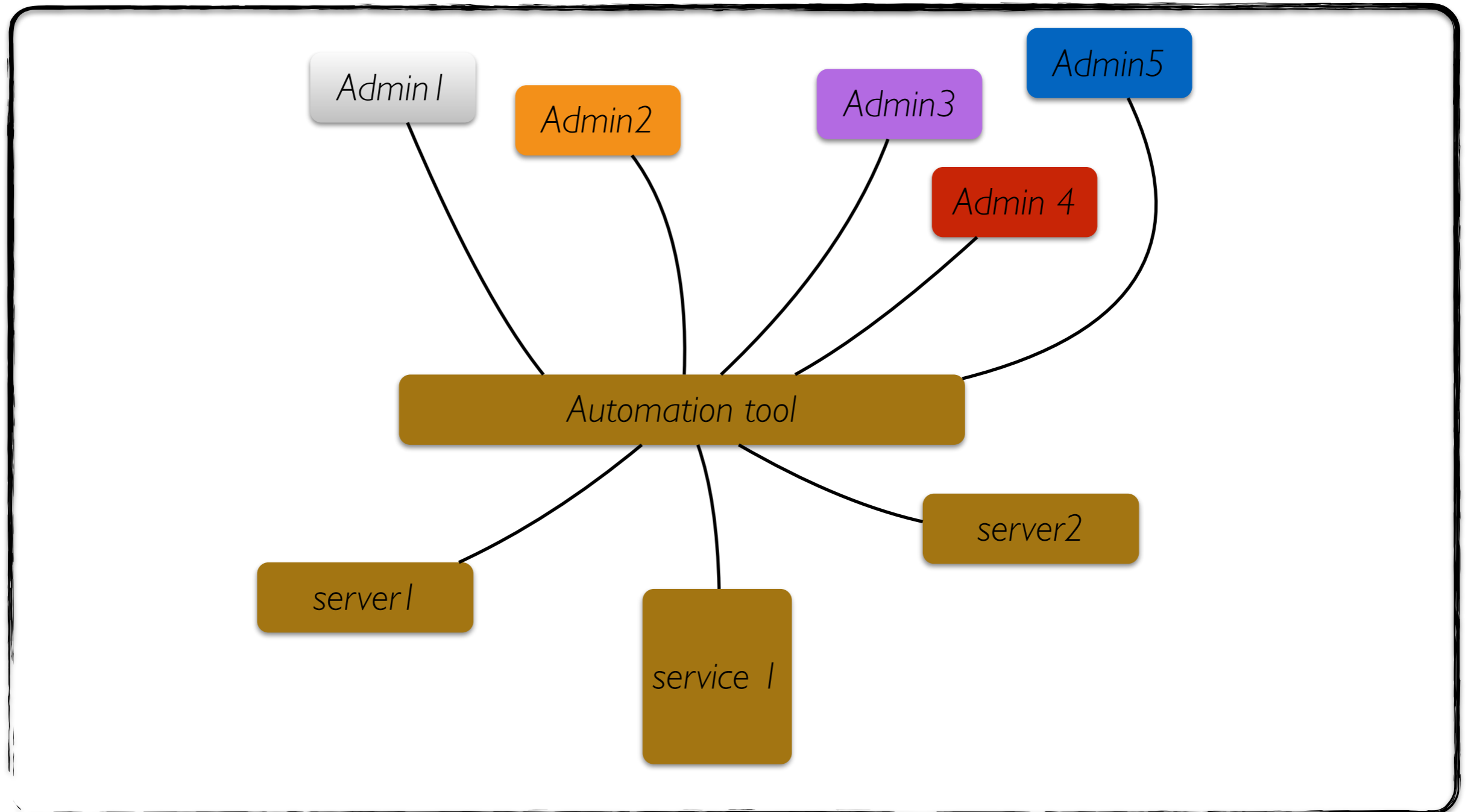
Contextualisation

Recovery

High level definition

Configuration language

Automate, why it is really important



Puppet, one solution !

Available free of charge

Server configuration defined at high level

Resources, classes and modules

Master/Agent, Masterless deployment

Node definition, manifest/site.pp

puppetAgent as daemon, cronJob or onDemand

Catalog compilation

Reports

Puppet basic

- *Define host configuration using the puppet language syntax*
- *Puppet will try to apply it on the host once it has a valid catalog which is the result of the compilation of the node definition*

Puppet basics: resource

*Define the single action to take on the target host.
There are a list of builtin resource as well as user
defined resource*

```
file { '/etc/passwd':  
  owner => root,  
  group => root,  
  mode  => 644  
}
```

Puppet basics: provider

Providers implement the same resource type on different kinds of systems. They usually do this by calling out to external commands.

*For example, **package** resources on Red Hat systems default to the **yum** provider*

Puppet basics: class

Classes are named blocks of Puppet code, which are not applied until they are invoked by name. They can be added to a node's catalog by either declaring them in your manifests

```
# A class with parameters
class apache ($version = 'latest') {
  package {'httpd':
    ensure => $version, # Using the class parameter from above
    before => File['/etc/httpd.conf'],
  }
  file {'/etc/httpd.conf':
    ensure => file,
    owner  => 'httpd',
    content => template('apache/httpd.conf.erb'), # Template from a
  }
  service {'httpd':
    ensure  => running,
    enable  => true,
    subscribe => File['/etc/httpd.conf'],
  }
}
```

Puppet basics: module

*Modules are self-contained bundles of code and data. You can write your own modules or you can download pre-built modules from the Puppet Forge
You will never instantiate a module*

Puppet basics: site.pp

The place [a file] where we declared all the classes or resources we wanted to apply

```
# Append this at the bottom of /etc/puppetlabs/puppet/manifests/site.pp

node 'agent1.localdomain' {

    # Note the quotes around the name! Node names can have characters that
    # aren't legal for class names, so you can't always use bare, unquoted
    # strings like we do with classes.

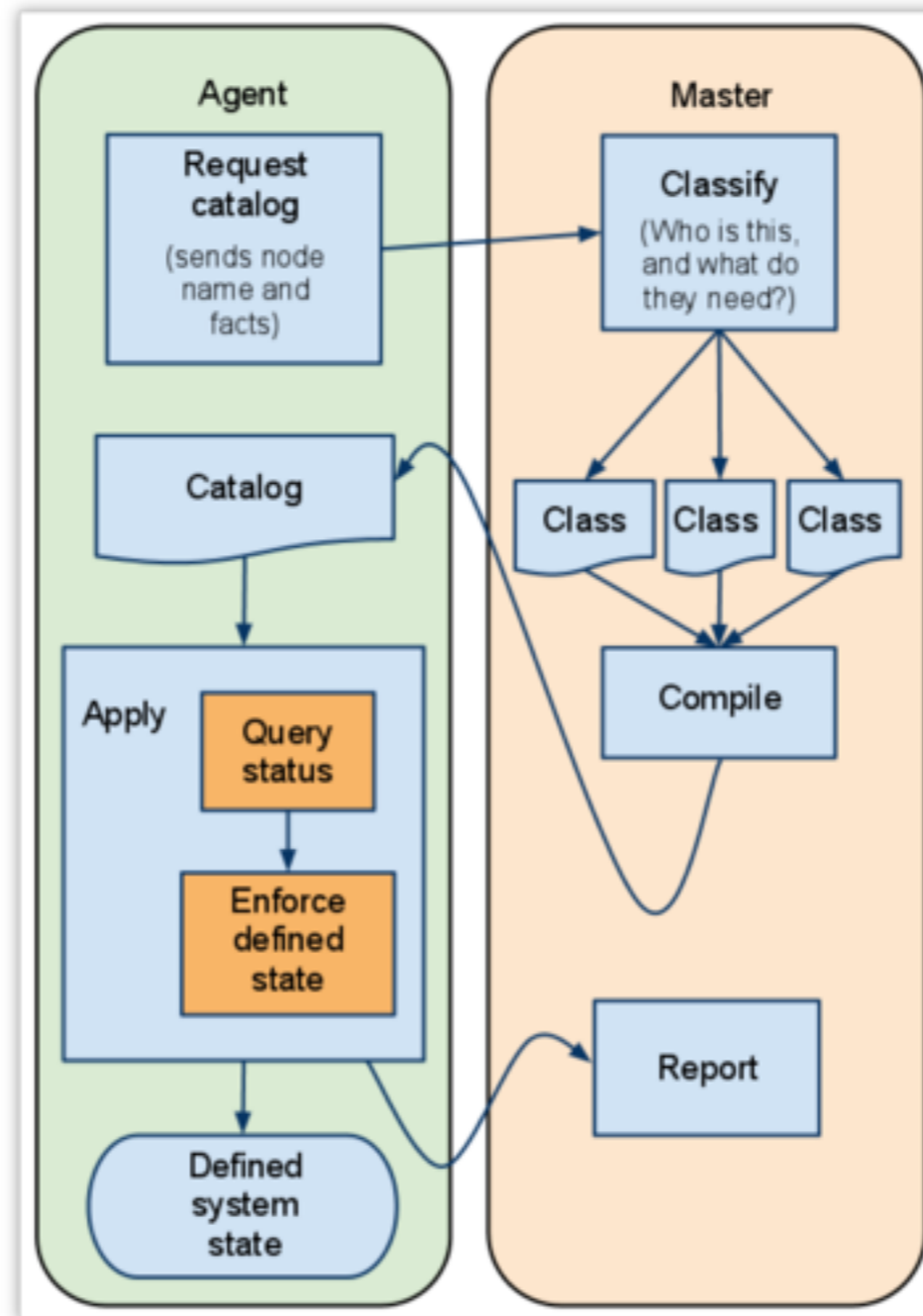
    # Any resource or class declaration can go inside here. For now:

    include apache

    class {'ntp':
        servers => [ "ntp1.example.com dynamic", "ntp2.example.com dynamic", ],
    }

}
```

Puppet Agent/Master workflow



Organise configuration using “hiera”

- Hiera is a key/value lookup tool
- Split configuration from puppet logic
- Hierarchical config
- Fine or macro

Hierarchy definition

```
[centos@puppetmaster-1 ~]$ cat /etc/puppet/hiera.yaml
# managed by puppet
---
:backends:
  - eyaml
  - yaml
:logger: console
:hierarchy:
  - secure
  - "nodes/%{::fqdn}"
  - "%{::environment}"
  - "%{::osfamily}"
  - common

:yaml:
  :datadir: /etc/puppet/environments/%{::environment}/hieradata

:eyaml:
  :datadir: /etc/puppet/environments/%{::environment}/hieradata
  :pkcs7_private_key: /etc/puppet/keys/private_key.pkcs7.pem
  :pkcs7_public_key: /etc/puppet/keys/public_key.pkcs7.pem

:merge_behavior: deeper

[centos@puppetmaster-1 ~]$
```

Puppet example

```
#cat site.pp
```

```
class { 'puppet::agent':  
  puppet_server => puppet.ba.infn.it,  
  environment   => production,  
  splay         => true,  
  puppet_run_interval => 15,  
}
```

Puppet example

```
#cat site.pp
```

```
class { 'puppet::agent':  
  puppet_server => puppet.ba.infn.it,  
  environment   => production,  
  splay         => true,  
  puppet_run_interval => 15,  
}
```

```
node 'myserver.ba.infn.it' {
```

```
  class { 'puppet::agent':  
    puppet_server => puppet.ba.infn.it,  
    environment   => production,  
    splay         => true,  
    puppet_run_interval => 30,  
    version       => '3.8.1-1puppetlabs1',  
  }  
}
```

It doesn't work, duplication declaration

Puppet example

```
#cat site.pp
```

```
node 'myserver.ba.infn.it' {
```

```
  class { 'puppet::agent':
```

```
    puppet_server => puppet.ba.infn.it,
```

```
    environment   => production,
```

```
    splay         => true,
```

```
    puppet_run_interval => 30,
```

```
    version       => '3.8.1-1puppetlabs1',
```

```
  }
```

```
}
```

```
node default {
```

```
  class { 'puppet::agent':
```

```
    puppet_server => puppet.ba.infn.it,
```

```
    environment   => production,
```

```
    splay         => true,
```

```
    puppet_run_interval => 15,
```

```
  }
```

```
}
```

Puppet example

```
#cat site.pp
```

```
hiera_include('default')
```

```
#cat common.yaml
```

```
---
```

```
default:
```

```
  - puppet::agent
```

```
puppet::agent::puppet_server: puppet.ba.infn.it
```

```
puppet::agent::environment: production
```

```
puppet::agent::splay: true
```

```
puppet::agent::puppet_run_interval: 15
```

```
#cat debian.yaml
```

```
puppet::agent::version: '3.8.1-1puppetlabs1'
```

```
#cat nodes/myserver.ba.infn.it
```

```
puppet::agent::puppet_run_interval: 15
```

syntax comparison

```
#cat site.pp
hiera_include('default')

#cat nodes/myserver.ba.infn.it
puppet::agent::puppet_run_interval: 30

#cat debian.yaml
puppet::agent::version: '3.8.1-1puppetlabs1'

#cat common.yaml
__
default:
  - puppet::agent

puppet::agent::puppet_server: puppet.ba.infn.it
puppet::agent::environment: production
puppet::agent::splay: true
puppet::agent::puppet_run_interval: 15
```

```
#cat site.pp

node 'myserver.ba.infn.it' {

class { 'puppet::agent':
  puppet_server      => puppet.ba.infn.it,
  environment        => production,
  splay              => true,
  puppet_run_interval => 30,
  version            => '3.8.1-1puppetlabs1',
}
}

node default {

class { 'puppet::agent':
  puppet_server      => puppet.ba.infn.it,
  environment        => production,
  splay              => true,
  puppet_run_interval => 15,
}
}
```

A puppet module to deploy HTCondor

- *already available a puppet module developed at Bristol University*
- *It uses classes to deploy the configuration*
 - *static approach*
 - *can it really support 5000 knobs ?*
 - *pre defined configuration files structure*

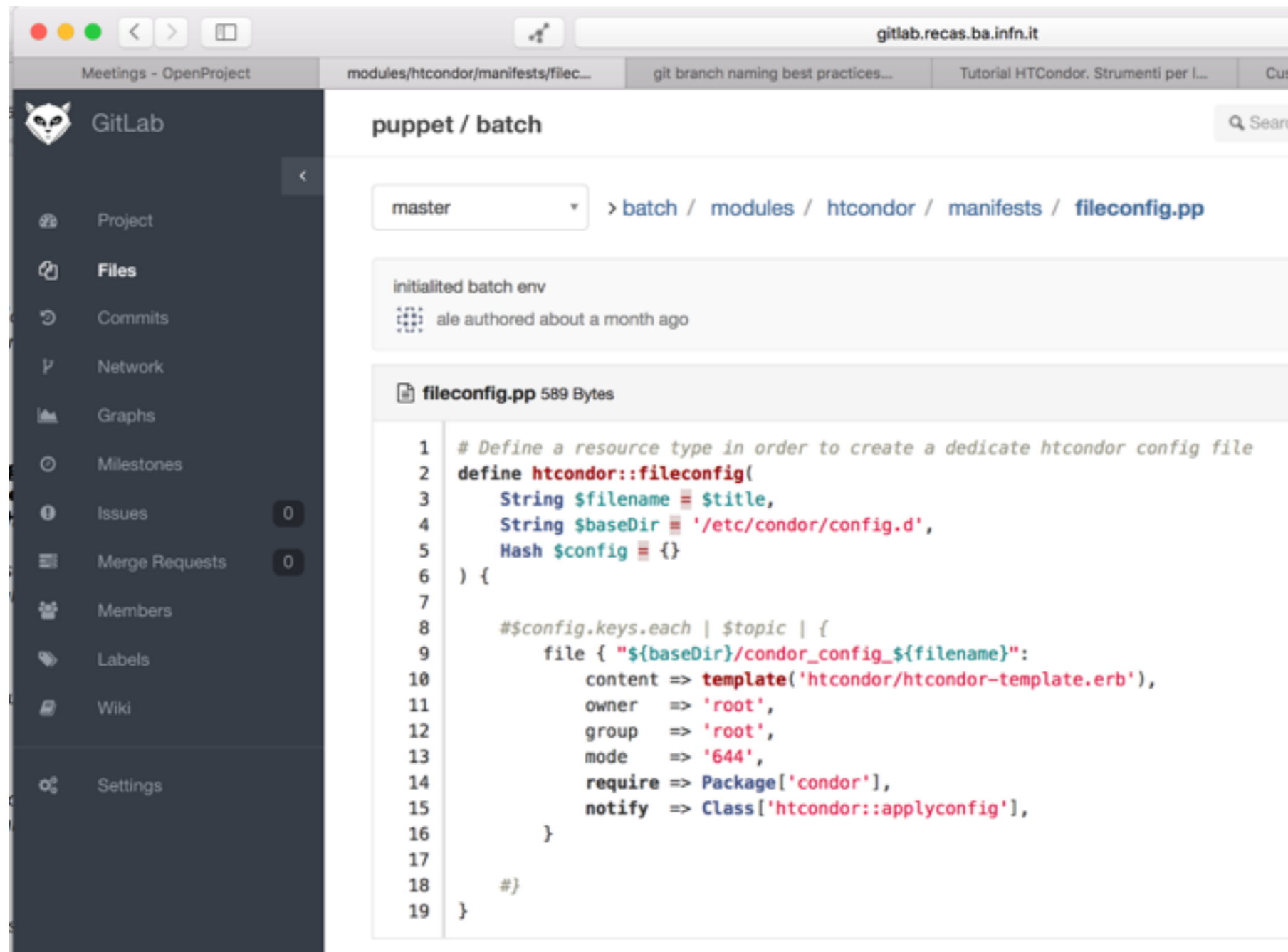
HTCondor puppet module, desiderata

- *taking into account my short experience deploying HTCondor by hand the module should provide the following features*
 - *support all the current and the future knobs*
 - *exploiting the LOCAL_CONFIG_DIR in order to group knobs into configuration files following the personal administrator criteria*
 - *configuration files name follow the administrator criteria*
 - *use defined resource types instead of class to support it*

HTCondor puppet module features

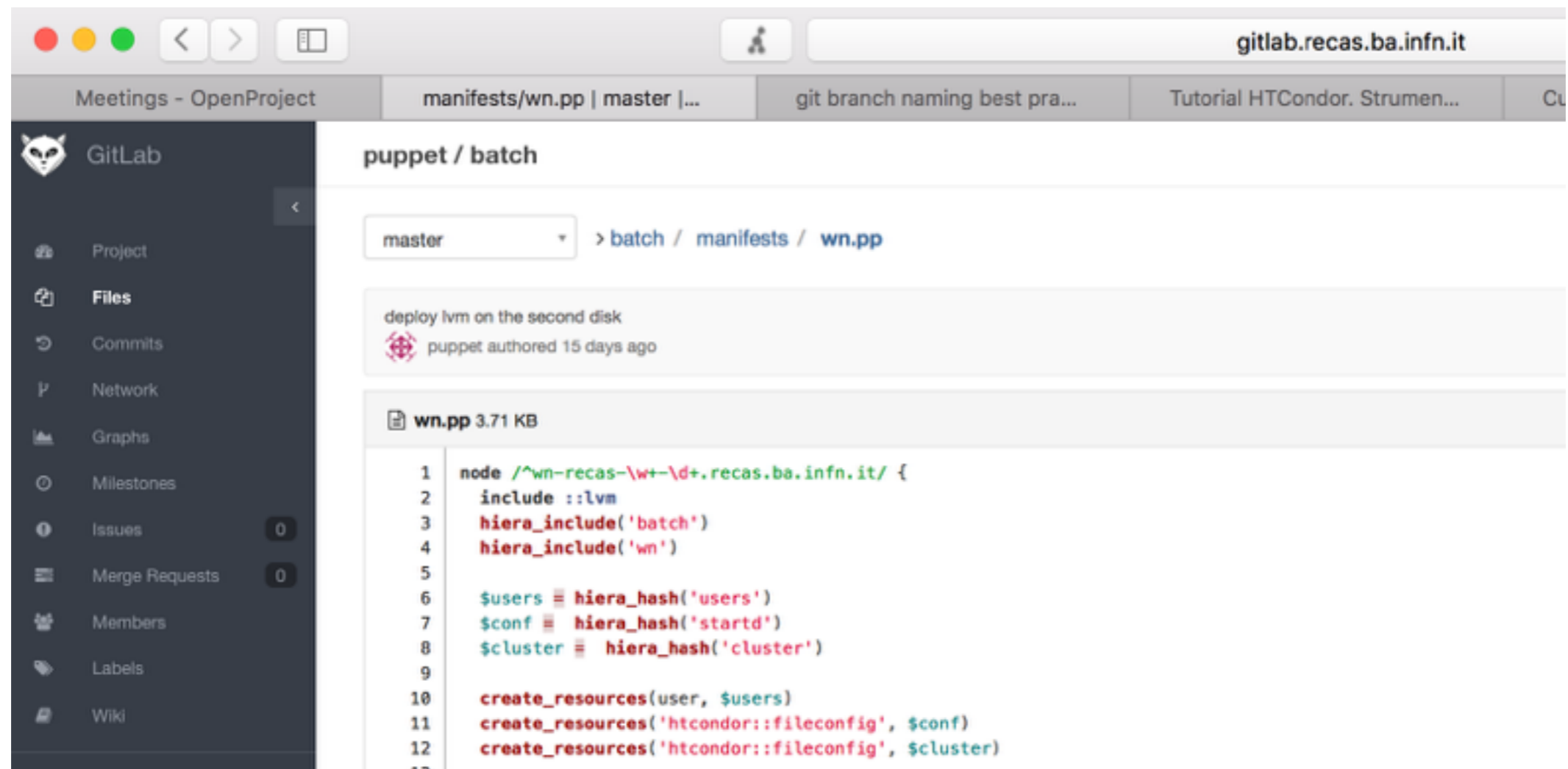
- *select the repo and install it*
- *install condor and pin it at*
- *handle the service*
- *Issue the condor_reconfig*
- *configure security using password*
- *configure cgroup in order to apply memory limit*
- *dynamically define HTCondor configuration using hiera*

core feature supply by a
“defined resources type”



```
1 # Define a resource type in order to create a dedicate htcondor config file
2 define htcondor::fileconfig(
3     String $filename = $title,
4     String $baseDir = '/etc/condor/config.d',
5     Hash $config = {}
6 ) {
7
8     # $config.keys.each | $topic | {
9     file { "${baseDir}/condor_config_${filename}":
10         content => template('htcondor/htcondor-template.erb'),
11         owner   => 'root',
12         group  => 'root',
13         mode   => '644',
14         require => Package['condor'],
15         notify => Class['htcondor::applyconfig'],
16     }
17
18     #}
19 }
```

example



The screenshot shows a web browser window with the URL `gitlab.recas.ba.infn.it`. The browser tabs include "Meetings - OpenProject", "manifests/wn.pp | master |...", "git branch naming best pra...", and "Tutorial HTCondor. Strumen...". The GitLab sidebar on the left contains navigation options: Project, Files, Commits, Network, Graphs, Milestones, Issues (0), Merge Requests (0), Members, Labels, and Wiki.

The main content area displays the file `puppet / batch` on the `master` branch. The file path is `batch / manifests / wn.pp`. A commit message "deploy lvm on the second disk" is shown, authored by "puppet" 15 days ago. The file `wn.pp` is 3.71 KB in size.

```
1 node /^wn-recas-\w+-\d+.recas.ba.infn.it/ {
2   include ::lvm
3   hiera_include('batch')
4   hiera_include('wn')
5
6   $users = hiera_hash('users')
7   $conf = hiera_hash('startd')
8   $cluster = hiera_hash('cluster')
9
10  create_resources(user, $users)
11  create_resources('htcondor::fileconfig', $conf)
12  create_resources('htcondor::fileconfig', $cluster)
13 }
```

GitLab

- Project
- Files
- Commits
- Network
- Graphs
- Milestones
- Issues 0
- Merge Requests 0
- Members
- Labels
- Wiki
- Settings

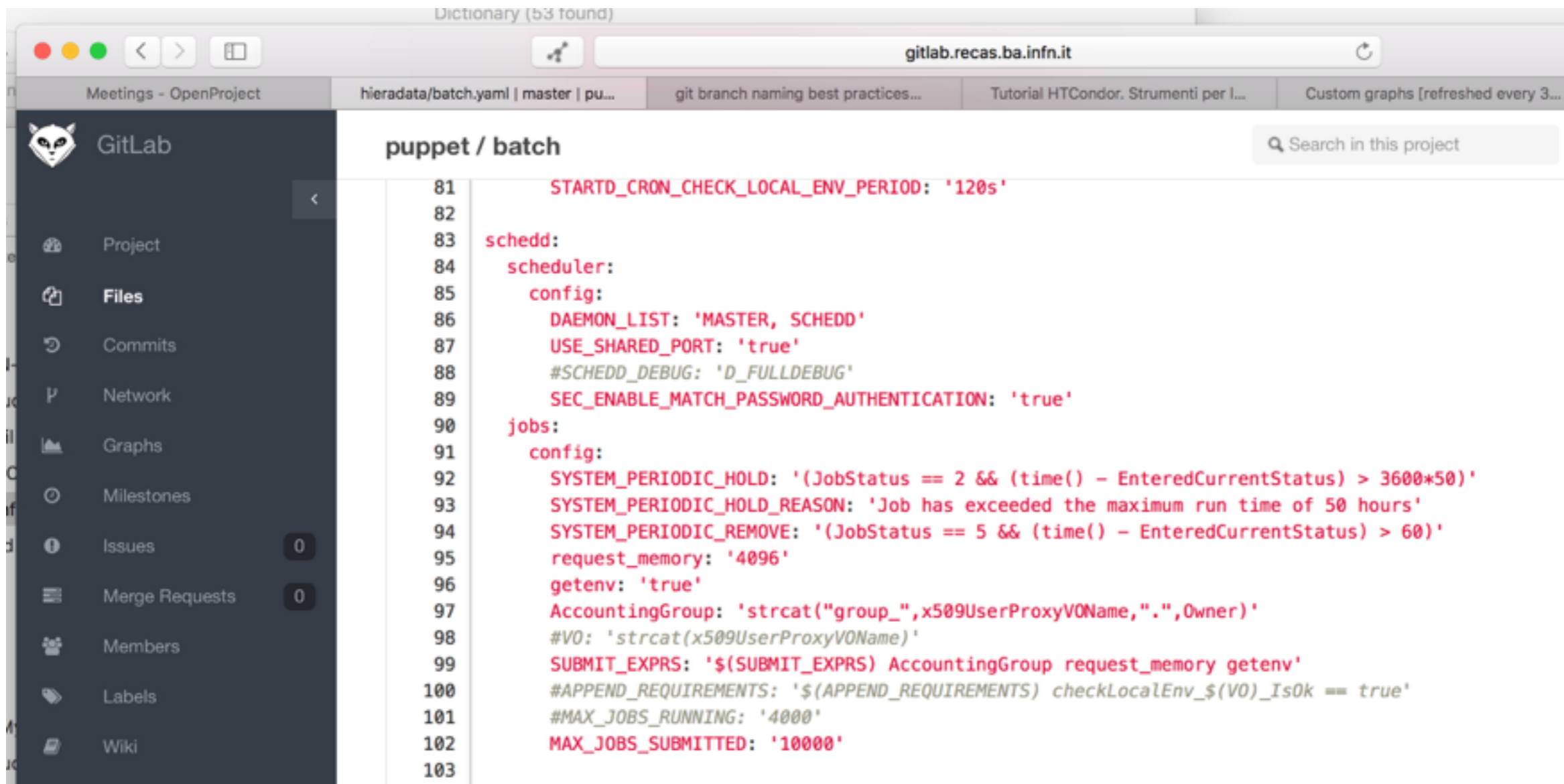
puppet / batch

```
38 cluster:
39   base:
40     config:
41       CONDOR_HOST: 'recas-htc-ctl-01, recas-htc-ctl-02'
42       UID_DOMAIN: 'ReCaSCluster'
43       TRUST_UID_DOMAIN: 'true'
44       SOFT_UID_DOMAIN: 'true'
45     security:
46       config:
47         SEC_PASSWORD_FILE: '/etc/condor/condorSharedSecret'
48         SEC_DAEMON_INTEGRITY: 'REQUIRED'
49         SEC_DAEMON_AUTHENTICATION: 'REQUIRED'
50         SEC_DAEMON_AUTHENTICATION_METHODS: 'PASSWORD'
51         SEC_CLIENT_AUTHENTICATION_METHODS: 'FS, PASSWORD'
52         #ALLOW_DAEMON: 'condor_pool@$(UID_DOMAIN)'
53         ALLOW_DAEMON: '*'
54
55   startd:
56     node:
57       config:
58         DAEMON_LIST: 'MASTER, STARTD'
59         #START: 'TotalSlots < 54'
60         START: 'TotalLoadAvg < 70'
61         RANK: 'TotalCpus - TotalSlots'
62         SUSPEND: 'false'
63         PREEMPT: 'false'
64         KILL: 'false'
65         #NUM_CPUS: '43'
66         NUM_SLOTS: '1'
67         NUM_SLOTS_TYPE_1: '1'
68         SLOT_TYPE_1: 'cpus=100%,mem=100%,auto'
69         SLOT_TYPE_1_PARTITIONABLE: 'true'
70         CONSUMPTION_POLICY: 'true'
71         CONSUMPTION_CPUS: 'TARGET.RequestCpus'
72         SLOT_WEIGHT: 'Cpus'
73         CGROUP_MEMORY_LIMIT_POLICY: 'soft'
74         BASE_CGROUP: 'htcondor'
75         #STARTD_DEBUG: 'D_FULLDEBUG'
76         EXECUTE: '/home/condor'
77         SEC_ENABLE_MATCH_PASSWORD_AUTHENTICATION: 'true'
78         ALLOW_DAEMON: 'submit-side@matchsession/90.147.168.7'
79         STARTD_CRON_JOBLIST: 'CHECK_LOCAL_ENV'
80         STARTD_CRON_CHECK_LOCAL_ENV_EXECUTABLE: '/usr/libexec/condor/check_vo_local_env.py'
81         STARTD_CRON_CHECK_LOCAL_ENV_PERIOD: '120s'
82
```

exploiting merge_behavior in order to provide fine host customization

- *Available with the parser future*
- *merge multiple dictionary in one main data structure*
- *the single dictionary can be defined in the hierarchy hiera files*
- *you can define the hierarchy direction using “deep” or “deeper” in the hiera configuration file*

Example



The screenshot shows a web browser window displaying the GitLab interface. The browser's address bar shows the URL `gitlab.recas.ba.infn.it`. The page title is `puppet / batch`. The left sidebar contains the GitLab navigation menu with items: Project, Files, Commits, Network, Graphs, Milestones, Issues (0), Merge Requests (0), Members, Labels, and Wiki. The main content area displays a Puppet manifest file named `batch` with the following configuration:

```
81     STARTD_CRON_CHECK_LOCAL_ENV_PERIOD: '120s'
82
83     schedd:
84       scheduler:
85         config:
86           DAEMON_LIST: 'MASTER, SCHEDD'
87           USE_SHARED_PORT: 'true'
88           #SCHEDD_DEBUG: 'D_FULLDEBUG'
89           SEC_ENABLE_MATCH_PASSWORD_AUTHENTICATION: 'true'
90       jobs:
91         config:
92           SYSTEM_PERIODIC_HOLD: '(JobStatus == 2 && (time() - EnteredCurrentStatus) > 3600*50)'
93           SYSTEM_PERIODIC_HOLD_REASON: 'Job has exceeded the maximum run time of 50 hours'
94           SYSTEM_PERIODIC_REMOVE: '(JobStatus == 5 && (time() - EnteredCurrentStatus) > 60)'
95           request_memory: '4096'
96           getenv: 'true'
97           AccountingGroup: 'strcat("group_",x509UserProxyVOName,".",Owner)'
98           #VO: 'strcat(x509UserProxyVOName)'
99           SUBMIT_EXPRS: '$(SUBMIT_EXPRS) AccountingGroup request_memory getenv'
100          #APPEND_REQUIREMENTS: '$(APPEND_REQUIREMENTS) checkLocalEnv_$(VO)_IsOk == true'
101          #MAX_JOBS_RUNNING: '4000'
102          MAX_JOBS_SUBMITTED: '10000'
103
```

example

The screenshot shows a web browser window displaying the GitLab interface. The browser's address bar shows the URL `gitlab.recas.ba.infn.it`. The page title is `puppet / batch`. A blue notification banner at the top states: "Your changes have been successfully committed". Below this, the breadcrumb navigation shows the current file path: `master > batch / hieradata / nodes / ce-01.recas.ba.infn.it.yaml`. A commit message is visible: `update puppet authored less than a minute ago`. The main content area displays the code for the file `ce-01.recas.ba.infn.it.yaml`, which is 494 Bytes. The code is as follows:

```
1 ---
2 grid:
3   - emi3repo
4   - emi3igisiteinfo
5
6 middleware: emi-cream-ce
7
8 nic:
9   eth1:
10     ensure: 'up'
11     ipaddress: '90.147.168.7'
12     netmask: '255.255.254.0'
13     dns1: '90.147.169.200'
14     domain: 'recas.ba.infn.it'
15     gateway: '90.147.168.1'
16
17 zabbix:
18   condor:
19     content: 'UserParameter=CONDOR.job_count[*,python /etc/zabbix/agent/CONDOR/job_count.py $1 $2'
20
21
22 schedd:
23   history:
24     config:
25       HISTORY: '/var/lib/condor/history/ce-01'
26   jobs:
27     config:
28       MAX_JOBS_RUNNING: '1567'
```