Using Puppet for DPM

Andrea Manzi
Martin Hellmich
Ricardo Rocha
Grigorii Latyshev
CERN

10/10/2014

Outline

- DPM configuration evolution
- Yaim2Puppet
- Example Manifests
- What's new
- Known Issues
- Future work
- Hands-on

DPM Configuration evolution (as presented during the previous Workshop)

YAIM

puppet labs

Sep ‘13 Jan ‘14 Sep ‘14

Cont. puppet module development

Obvious differences

YAIM

- Configuration only
- Configuration all local
- Only EMI components
- Bash
- You can do anything you want

Puppet

- Install and configuration
- Puppet Masters or **local**
- Everything
- Puppet DSL or ruby
- Adhere to puppet conventions

Puppet modules

- “Our” modules:
  dmlite, gridftp, lcgdm, xrootd
- Other CERN modules used:
  bdii, fetch-crl, voms
- External modules:
  firewall, mysql, stdlib

External Contributions to the modules

- From January we started publishing our modules to GitHub for external contributions:
- https://svnweb.cern.ch/trac/lcgdm/wiki/Dpm/Admin/ContributeToPuppetModules

- Thanks for your help!

Twiki configuration page

Up | Start Page | Index | History | Last Change

Dpm / Admin / InstallationConfigurationPuppetSimple

DPM (Disk Pool Manager)

The Disk Pool Manager (DPM) is a lightweight solution for disk storage management. It is part of LCGDM (https://svnweb.cern.ch/trac/lcgdm)

- DPM home (https://svnweb.cern.ch/trac/lcgdm/wiki/Dpm)
- Support: DPM Users Forum, GGUS Helpdesk, Dev List
- User Docs: (Tutorial, Cookbook / FAQ?, API Overview, Command Line Tools)
- Admin Docs: (Installation, Configuration, Maintenance, Performance, Monitoring, Reference Card, EMI Reference Card)
- Development: (Savannah, Certification, Release Procedure, Internal Testbed, Components, Coordination, Recipes)

Installation and Configuration Guide

This guide explains the installation and configuration procedure of DPM with puppet. It will introduce the puppet components to be installed, but not give an introduction into writing puppet manifests. For that, you can look at the fantastic documentation at puppetlabs. Please also have a look at this, if you want to find out the difference between modules and manifests. This guide also applies to other components, e.g. the LFC, for which puppet modules exist.

If you want to install DPM without using puppet, the following pages describe the configuration manually or with YAIM:

- Installation Guide
- Configuration with YAIM
- Manual Configuration

If you want to know more details about the puppet modules, follow this tutorial. It shows an alternative way to use the puppet modules showing exposing more configuration parameters and modules:

- Detailed Puppet Installation Guide

Open Issues

- **GridFTP** The puppet installation instructions are valid for the gridftp frontend based on dmlite, i.e. dpm-dsi >= 1.9.0. To configure the DPM-based gridftp frontend, please use the yaim configuration.

Overview

Puppet is a configuration management system that can not only preserve and update the configuration of your components, but also manage the installation. That is why this guide is an integrated installation/configuration guide. As puppet modules are developed by a larger community, you can also manage other parts of your system, e.g. the yum configuration or the firewall settings, with puppet. In the following, we describe the settings and how they can be applied with puppet.

The Sections

https://svnweb.cern.ch/trac/lcgdm/wiki/Dpm/Admin/InstallationConfigurationPuppetSimple

Puppet modules from Puppetforge

- yum install puppet

- puppet module install lcgdm-lcgdm
- puppet module install lcgdm-dmlite
- puppet module install lcgdm-gridftp
- puppet module install lcgdm-xrootd

https://forge.puppetlabs.com

Puppet modules versioning

- Each module has its own version number
- So far we have been always managed to keep backward compatibility when releasing a new version (and we would like to continue this way)
- We can safely use the latest versions of our modules to configure a DPM from version 1.8.7 onwards
  - In 1.8.7, the gridftp frontend (dpm-dsi) is not configurable via puppet

Examples

- https://github.com/cern-it-sdc-id/lcgdm-puppet-examples
- Manifests:
  - DPM-Head
  - DPM-Disk
  - DPM-Head+ DPM-Disk
  - LFC
  - repos
- new installation VS upgrade

Migration: Yaim2puppet

- We put effort to develop some Yaim functions to generate puppet manifests from site-info.def files
- https://github.com/cern-it-sdc-id/yaim2puppet
- The repo contains also a file showing mapping between YAIM variable to puppet variable
- Testing from the Collaboration sites is welcome☺

Yaim2puppet: how to use it

- /opt/glite/yaim/bin/yaim -r –s <path to site-info.def file> -f <function>

- Where <function> = config_DPM_puppet_head | config_DPM_puppet_disk | config_LFC_puppet

How to run puppet

- puppet apply headnode.pp

Or

- puppet agent -t

A look at the examples

- Variables

```
$token_password = "TOKEN_PASSWORD"
$mysql_root_pass = "PASS"
$db_user = "dpmmgr"
$db_pass = "MYSQLPASS"
$domain = "cern.ch"
```

A look at the examples

- Dependencies between modules
  - Puppet runs are multithreads, need a way to specify the order of executions.

Class[Mysql::Server] -> Class[Lcgdm::Ns::Service]

Class[Bdii::Install] -> Class[Lcgdm::Bdii::Dpm]

A look at the examples

- Firewall rules

```
firewall{"050 allow http and https":
  proto  => "tcp",
  dport  => [80, 443],
  action => "accept"
}
```

A look at the examples

- DPM

```
class{"lcgdm":
  dbflavor => "mysql",
  dbuser   => "${db_user}",
  dbpass   => "${db_pass}",
  dbhost   => "localhost",
  domain   => "${domain}",
  volist   => $volist,
}
```

A look at the examples

- dmlite

```
class{"dmlite::head":
  token_password => "${token_password}",
  mysql_username => "${db_user}",
  mysql_password => "${db_pass}",
}
```

A look at the examples

- Pools and filesystems

```
lcgdm::dpm::pool{"mypool":
  def_filesize => "100M"
}

lcgdm::dpm::filesystem {"${::fqdn}-myfs":
  pool   => "mypool",
  server => "${::fqdn}",
  fs     => "/fslocation"
}
```

A look at the examples

- Frontends

```
class{"dmlite::dav":}
class{"dmlite::srm":}
class{"dmlite::gridftp":
  dpmhost => "${::fqdn}"
}
class{"lcgdm::rfio":
  dpmhost => "${::fqdn}",
}
```

A look at the examples

- XROOTD

```
class{"xrootd::config":
  xrootd_user  => 'dpmmgr',
  xrootd_group => 'dpmmgr'
}
class{"dmlite::xrootd":
  nodetype             => [ 'head' ],
  domain               => "${domain}",
  dpm_xrootd_debug     => $debug,
  dpm_xrootd_sharedkey  => "${xrootd_sharedkey}"
}
```

A look at the examples

- Security

```
lcgdm::shift::trust_value{
  "DPM TRUST":
    component => "DPM",
    host      => "${disk_nodes}";
  "DPNS TRUST":
    component => "DPNS",
    host      => "${disk_nodes}";
  "RFIO TRUST":
    component => "RFIOD",
    host      => "${disk_nodes}",
    all       => true
}
```

```
lcgdm::shift::protocol{"PROTOCOLS":
  component => "DPM",
  proto     => "rfio gsiftp http
https xroot"
}

class{"voms::atlas":}
class{"voms::dteam":}
```

Note on Installation

- The Modules install the latest packages version available on the repos, but do not perform automatic upgrade

```
class dmlite::install (
  $debuginfo = false
) inherits dmlite::params {
..
  package {"dmlite-libs":
    ensure => present;// N.B. not latest..
  }
..
}
```

What's new

- Logging
- Real-Time Monitoring
- Memcache plugin
- Ipv6
- Xrootd4

Logging

It comes with latest dmlite (0.7.0)

- Most of the dmlite plugins are now properly logging to /var/log/dmlite/dmlite.log

```
class{"dmlite::head":
        ...
        log_level   =>  <level>,
        logcomponents  => <compoenent_name>,
        ..
        }
```

- https://svnweb.cern.ch/trac/lcgdm/wiki/Dpm/Admin/Installation_Configuration_Puppet#ConfigureLogging

Real-time Monitoring

- It comes with latest dmlite (0.7.0)
  - xrootd style for other backends (http/dav, gridftp)

```
class{"dmlite::plugins::profiler":
      collectors => ["<collector-host>:<port>"],
      auth       => <false|true|"userdn">,
}
```

- https://svnweb.cern.ch/trac/lcgdm/wiki/Dpm/Admin/Installation_Configuration_Puppet#MonitoringviaProfiler

Memcache plugin

- Available since 1.8.8 but more stable in the new version based on dmlite 0.7.0 (It has also a memory caching)

```
class{"memcached":
    max_memory => 512,
}
 ->
   class{"dmlite::plugins::memcache":
      expiration_limit => 600,
      posix            => 'on',
      func_counter     => 'on',
     }
```

- Local or external Memcached instance(s)
- https://svnweb.cern.ch/trac/lcgdm/wiki/Dpm/Admin/Installation_Configuration_Puppet#Installationandconfigurationofthedmlite-memcacheplugin

IPv6

- DPM services are IPv6 compliant (DPM-xrootd 3.5.0 with xrootd4 only missing)
- Given a node with an IPv6 address only the ip6tables rules need to be configured:

```
firewall{"050 allow 050 allow http and https for ipv6":
            provider  => 'ip6tables',
            state  => "NEW",
            proto  => "tcp",
            dport  =>[80, 443],
            action => "accept"
           }
```

- https://svnweb.cern.ch/trac/lcgdm/wiki/Dpm/Admin/Installation_Configuration_Puppet#IPV6conf

Xrootd4 support

- The puppet-dmlite 0.3.0 module already contains modification for the new Xrootd4.
  - Vomsxrd4 installation
  - Different type of log rotating

- To be enabled once released

Known issues: service restarts

- In some cases rpms upgrades/conf changes do not trigger DPM services and frontends restarts even after a puppet run
  - Yaim was configured to always restart the service at each execution
- This differs on each deployment cause not all frontends services are installed by default.
- Solution1: add deps depending on the cases:
  - Class[Dmlite::Plugins::Profiler::Install] ~> Class[Dmlite::Dav::Service]
  - Class[Dmlite::Plugins::Memcache::Install] ~> Class[Dmlite::Dav::Service]
- Solution2 : add services restart at packaging level, but it covers only the rpm upgrade case

Known issues: dpmmgr uid

- YAIM used to have a default uid value for the dpmmgr user (151).
- Puppet-lcgdm module misses this default value -> automatically assigned by the OS.
  - This means that new disks nodes added via puppet to existing installation could have issues.
- https://svnweb.cern.ch/trac/lcgdm/blog/PUPPETLEGACY

Known issues: fetch-crl and httpd

- Yaim was configuring the fetch-crl cron to reload the httpd service only in case of WebDav installation
- This is actually missing and not configurable from the CERNOps-fetchcrl module we are using.

- Recently discovered, some manual changes to apply will be documented

Known issues: xrootd federation

- This is not a real issue but the current puppet configuration for xrootd federations is quite complicated
  - Experiments federations are not easy to configure in general.
- Dedicated wiki page:

https://svnweb.cern.ch/trac/lcgdm/wiki/Dpm/Xroot/PuppetSetup#Joiningafederation

Future work: Full conf via Hiera

- https://docs.puppetlabs.com/hiera/1/
- key/value lookup tool for configuration data
- “makes Puppet better by keeping site-specific data out of your manifests.”
- We have started to implement variables lookup via Hiera, but the goal is to have every parameters configurable this way
- $servers = hiera('dmlite::plugins::memcache::params::servers', ["localhost:11211"])

Future work: Puppet-dpm

- Puppet-dpm is a “meta-module” + wrapper to existing developed module
- developed by A. Sartirana with small modifications on our side
  - Includes all other module dependencies
  - Hides the modules inter-dependencies
  - Parameters configurable via hiera

Future work: Puppet-dpm

```
class{"dpm::disknode":
  configure_vos         => false,
  configure_gridmap     => true,

  #cluster options
  headnode_fqdn         => "${::fqdn} ",
  disk_nodes            => "dpm-puppet02.cern.ch",
  localdomain           => "cern.ch",
  webdav_enabled        => true,
}
```

- https://github.com/cern-it-sdc-id/puppet-dpm

Future work: Puppet-dpm

- Pros
  - Resolution of the modules deps
  - Parameters can be specified like in yaim
- Cons
  - wrapping of all dmlite/lcgdm parameters needed if hiera is not used..
- Status
  - Still under development
  - Missing documentation

Future work: Enhancements

- Configuration of GridFTP redirection
- Effort from Collaborations?
  - Update S3/HDFS/VFS manifests plugins and testing
  - Yaim2puppet testing?
    - We could imagine a new version which generates a manifest for the puppet-dpm

Do an install from scratch

HANDS ON

QUESTIONS?

steps

- Puppet & git installation
- Puppet modules installation
- Repo conf
- Node conf
- Tests from lxplus