Lab 04.2: A Puppet Module to Manage MariaDB
IN719 Systems Administration

Introduction

In this lab we will build a more complex module to manage our database server software. This module will use a collection of related *classes*. We've used classes already, but you may not have paid much attention to them. Note that this lab is based on an example from *Pro Puppet* by James Turnbull. In this module we will handle not just installation and configuration of a service, but also preinstallation tasks, operation, and ongoing maintenance of the service.

1 Module setup

Create a standard module structure with the following files and directories in the `/etc/puppet/code/modules` directory of your puppetmaster.

```
mariadb
mariadb/files/50-server.cnf
mariadb/manifests/init.pp
mariadb/manifests/install.pp
mariadb/manifests/config.pp
mariadb/manifests/service.pp
mariadb/templates
```

Notice that we're using some more manifest files than we've used in the past. We will be writing a bit more code and we need to organise it more deliberately. Also, note that you can get a copy of the `50-server.cnf` file from the `week04` subdirectory of the class GitHub repository.

2 mariadb::install

The **mariadb::install** class includes the resources needed to install MariaDB. Put the following in your `install.pp` file

```
class mariadb::install {
  package { "mariadb-server" :
            ensure => present,
            require => User["mysql"],
  }
  user { "mysql":
         ensure => present,
         comment => "MariaDB user",
         gid => "mysql",
         shell => "/bin/false",
         require => Group["mysql"],
  }
  group { "mysql" :
          ensure => present,
  }
 }
```

It's not a typo that the user and group are "`mysql`".

Note how we use `require` directives to make sure that things are set up in the correct order, and we don't bother attempting steps that will fail because prerequisites are not met.

3 mariadb::config

Place the following resources in your `config.pp` file.

```
class mariadb::config {
  file { "/etc/mysql/mariadb.conf.d/50-server.cnf":
    ensure => present,
    source => "puppet:///modules/mariadb/50-server.cnf",
    mode => "0444",
    owner => "root",
    group => "root",
    require => Class["mariadb::install"],
    notify => Class["mariadb::service"],
  }
}
```

Notice how these resources require mariadb::install, and they also *notify* mariadb::service. The `require` directive means that Puppet won't apply the `config` class if the `install` class hasn't been applied successfully. The `notify`directive means that the server daemon will be restarted whenever Puppet changes its configuration.

4 mariadb::service

The maria::service class is brief. Place it in your `service.pp` file.

```
class mariadb::service {
  service { "mysql" :
    ensure => running,
    hasstatus => true,
    hasrestart => true,
    enable => true,
    require => Class["mariadb::config"],
  }
}
```

This class will make sure that the server daemon is running and will restart it if necessary when its configuration is changed by Puppet.

5 mariadb class

Finally we just combine our classes in the `init.pp` file.

```
class mariadb {
  include mariadb::install, mariadb::config, mariadb::service
}
```

Now you can apply the module to your db server by placing `include mariadb` in the node definition for your db server. Don't include this module in other nodes because we don't want to install or run the MariaDB server on them.