

# DaticalDB Puppet Module

# Quick Overview of PE (Puppet Enterprise)

## Puppet Enterprise Installation

Puppet Enterprise is the commercial offering built up around open source puppet, one of the benefits of the enterprise offering is a “wizard” style installer that asks you a few questions and handles the install from there. Full Documentation on the installation process can be found on the Puppet Labs website ([http://docs.puppetlabs.com/pe/latest/install\\_basic.html](http://docs.puppetlabs.com/pe/latest/install_basic.html)).

You need to have DNS setup for the server in question before proceeding, it's highly recommended that you have a puppet.yourdomain.com alias that is pointing at the server you plan to install the puppet server software on.

You do not need to purchase a license to work with Puppet Enterprise in a small environment, they provide 10 nodes for free, the puppet master is one of those nodes so you can manage 9 secondary boxes without purchasing a license for additional hosts.

### Downloading the software

Pick the version that is appropriate for your operating system from here: <http://info.puppetlabs.com/download-pe.html>

You have the choice of RHEL and it's clones, Modern Windows versions, Ubuntu, Debian, Solaris, SLES and AIX. I would not suggest attempting installation outside of those, fedora for example, it does check the OS installed before attempting an installation.

### Performing the server install

After downloading the software to the system you intend to install Puppet Enterprise on you need to unpack the source, this is done with the tar command, an example given RHEL as your system type:

```
tar -zxvf puppet-enterprise-3x-el-6-x86_64.tar.gz
```

After unpacking the source you should have a directory with the installers contents, change into that:

```
cd puppet-enterprise-3x-el-6-x86_64.tar
```

You should now be able to begin the install:

```
./puppet-enterprise-installer
```

You will be prompted for yes / no answer questions about the roles you want to install, if you want a standalone box to be your puppet master (the server for puppet) then you need to answer yes to all the questions except the cloud provisioner, you only need that if you intend to have puppet manage the creation of VMware or Amazon EC2 hosts.

You will be asked several questions after you select the roles:

DaticalDB Puppet Module

Cert. name - This should default to the hostname, if you didn't create a puppet.yourdomain.com alias then the hostname that other systems can reach the server from should be chosen.

Database Support - This document assumes you are doing a standalone server, choose the option that has the installer install a database server and the appropriate schema for you.

Console Port - 443 is the default, this should be fine. It will allow you to reach the server via <https://server.yourdomain.com>

User Email and Password - This is the initial admin account, pick an email and password you will definitely remember.

SMTP Server - localhost should be fine.

Puppet Agent Certname - the hostname of the box is fine

Puppet Agent Master Hostname - This will be what you chose for the cert. name of the puppet master above.

Vendor Packages - You need to allow the installer to install any missing pre-requisites

Convenience Links - I suggest installing the convenience links to allow easier use of the puppet commands

### Performing the client install

The server doesn't do much good without a box to manage, I do not recommend doing your testing on the puppet master itself as you may have certain environmental settings provided by puppet for the master role that you wouldn't be accounting for on a generic server.

By this point you have already installed a client role, the server is also it's own client. You will want to copy the puppet enterprise installer onto your client machine and follow the install steps above to start the installer. This time you will only want to answer yes to the client role. Providing it with information on the puppet master you installed above.

### Approving the client

After installing a client the server must sign the certificate of the client and allow that client into the puppet environment.

Login to your puppet enterprise console by going to the hostname and port you setup for the console service.

By default that should be:

<https://yourserver.yourdomain.com>

The login and password will be what you chose during the puppet master installation.

After successfully logging in you will want to go to the node requests page and approve the node, the link for the approval page is circled below:



After reaching this page you should see 1 node request to join your site. Hit the blue Accept button for that host or the Accept All button to approve the request. You should now have a working puppet enterprise client / server environment.

# Installation and Usage of the DaticalDB Puppet Module

## Module Installation

Modules can be added to puppet enterprise via two methods, either by the puppet forge which works for modules that have been submitted to the puppet forge. Quite often useful modules are found on github or other sources and you can install them by hand without much effort.

### Manual installation

For a module to be seen by the puppet master in puppet enterprise it must be installed under `/etc/puppetlabs/puppet/modules`. This documentation assumes you have access to the github repository the daticaldb module is stored in.

First you need to get into the directory that houses the modules

```
cd /etc/puppetlabs/puppet/modules
```

Now clone the github repository to install the module

```
git clone git@github.com:ptinsley/daticaldb-puppet.git daticaldb
```

You should now have the module directory and it's contents available for puppet to use.

## Module Usage

There are two main ways that module components are built as a class or as a define. Classes can only exist on a box one time, this is a good choice for something like the daticaldb software install. For projects that we need to deploy that isn't a great option. Unfortunately the puppet enterprise gui doesn't support defines currently, you will have to manage the resource via the command line.

### Installing DaticalDB on a node

In the general case for DaticalDB you likely won't be able to source the rpm from an upstream repository, for that reason we package the rpm with the module. The usage below shows how you setup your installation to use the packaged rpm vs a repository source.

In the simplest layout puppet has only one manifest file, you can make this layout quite complex, this documentation assumes you have a base install with no customizations. You will need to edit the `/etc/puppetlabs/puppet/manifests/site.pp` file in your editor of choice. Inside that file you will need a node construct that represents the host you wish to install the daticaldb software on. Example using `testbox.yourdomain.com`:

```
node 'testbox.yourdomain.com' {  
  
    include daticaldb::engine  
  
}
```

Were you to have your own repository you would want that to look like this instead:

```
node 'testbox.yourdomain.com' {
  class {'daticaldb::engine':
    package_in_repo => true
  }
}
```

### Having DaticalDB manage a project

A project is where the real work is done, you point daticaldb at a project directory and it will manage the state of the migrations that are available in that project directory. If you were to update your project directory with newer migrations, the next time puppet runs it will execute those new migrations. Below is an example project running against a local mysql server. You would open the site.pp file mentioned above again in your text editor of choice and expand the node section to include the project definition.

```
node 'testbox.yourdomain.com' {
  class {'daticaldb::engine':
    package_in_repo => true
  }

  daticaldb::project {'Test':
    db_hostname => 'localhost',
    db_name      => 'puppet_test',
    db_password  => 'daticalpass',
    db_port      => '3306',
    db_type      => 'mysql',
    db_username  => 'datical',
    path         => '/datical/projects/Test',
    ensure       => 'latest',
    driver_path  => '/lib64',
    db_instance => 'Source'
  }
}
```