

---

# **puppet-diamond Documentation**

***Release 0.3.0***

**Ian Dennis Miller**

**Mar 21, 2017**



---

## Contents

---

<b>1</b>	<b>Overview</b>	<b>3</b>
<b>2</b>	<b>Introduction</b>	<b>5</b>
<b>3</b>	<b>User Guide</b>	<b>9</b>
<b>4</b>	<b>About</b>	<b>15</b>



# Puppet-Diamond

**Puppet-Diamond** is framework for creating and managing an IT Enterprise consisting of many Linux servers.



# CHAPTER 1

---

## Overview

---

Automate the administration of your servers using Puppet and Vagrant. Puppet-Diamond provides a standardized installation of Ubuntu Linux, which can be configured and deployed to create almost any enterprise IT topology. The Puppet-Diamond framework is used to create a host capable of running Flask-Diamond applications.

## Installation

```
pip install Puppet-Diamond
```

## Documentation

<http://puppet-diamond.readthedocs.org>





## Quick Start

### Installation

There are three steps to installing Puppet-Diamond:

1. Install the Puppet-Diamond software
2. Create a global Puppet-Diamond configuration
3. Initialize a directory that stores the Puppet Master configuration

```
mkvirtualenv puppet-diamond
pip install Puppet-Diamond
domo-new.sh puppet-diamond
domo-new.sh puppetmaster
```

### Basic Usage

1. sync the local config to the puppet master
2. test the config
3. apply the config

```
domo-sync.sh
domo-test.sh host1.example.com
domo-apply.sh host1.example.com
```

### Creating new assets

Follow the setup procedure described in the relevant projects:

1. *Creating a Profile*
2. *Provisioning a Host*
3. *Creating a Puppet Module*
4. *Creating a Flask-Diamond Module*

## System Requirements

Puppet-Diamond requires some software to be installed in order to function. Once you have installed these requirements, you can follow the [Quick Start](#) to start your first project. The following packages should be installed globally, as the superuser, for all users on the system to access.

- Python 2.7.x.
- Python development libraries (i.e. header files for compiling C code)
- pip
- virtualenv
- virtualenvwrapper

The following sections describe the process for installing these requirements on various systems. In each of the following examples, it is assumed you will be using a root account (or some other privileged account).

If you do not have root access, then refer to the section [Unprivileged Installation](#) for information about creating a virtualenv in your user account.

### Debian/Ubuntu

Puppet-Diamond installs cleanly on Debian and Ubuntu systems released after 2011.

```
apt-get install python python-dev python-pip build-essential
apt-get install sqlite-dev
pip install --upgrade pip
pip install --upgrade virtualenv
pip install virtualenvwrapper
```

### Redhat

Puppet-Diamond can be installed on RedHat, but ensure your package manager is installing Python 2.7; as of August 2015, RHEL provides an older version.

```
yum install python python-devel python-pip
yum install sqlite-devel
pip install --upgrade pip
pip install --upgrade virtualenv
pip install virtualenvwrapper
```

### OSX with Homebrew

Puppet-Diamond installs pretty easily on OSX with Homebrew. Make sure you are using the *admin* user for this process, just like a normal Homebrew operation.

```
brew install python --universal --framework
brew install pyenv-virtualenv
brew install pyenv-virtualenvwrapper
brew install sqlite
pip install --upgrade pip
```

## Windows with Cygwin

Here are a few resources to get you started:

- <http://www.pdxdpixel.com/blog/setting-up-python-and-virtualenv-windows-cygwin/>
- <http://atbrox.com/2009/09/21/how-to-get-pipvirtualenvfabric-working-on-cygwin/>
- <http://anythingsimple.blogspot.ca/2010/04/using-pip-virtualenv-and.html>
- <http://stackoverflow.com/questions/2173963/how-do-i-get-virtualenvwrapper-and-cygwin-to-co-operate>

**Note:** Have you done this install successfully? Let us know!

## Unprivileged Installation

Sometimes, you do not have root access to the system. It is still possible to use Puppet-Diamond, but the installation process is slightly different because it does not use virtualenvwrapper. Instead, you will create your virtualenv directly and use the *activate* macro to work on it.

```
curl -O https://raw.githubusercontent.com/pypa/virtualenv/master/virtualenv.py
python virtualenv.py my-diamond-app
. my-diamond-app/bin/activate
pip install Puppet-Diamond
```



## Creating a Profile

A “profile” is a type of host with the configuration you specify.

There might be many hosts that implement the same profile. For example, you may have multiple application servers behind a load balancer. To implement such a setup, you would need two profiles: an application server profile and a load balancer profile.

A profile may be used to provision new hosts on any cloud:

- a commercial host like Digital Ocean
- locally using a KVM setup

## Scaffold a Profile

The general form for creating a new profile looks like this

```
demo-new.sh profile $PROFILE_TYPE
```

So to create a new profile called *application-server* that will be tailored for running applications:

```
demo-new.sh profile application-server
```

## Next Steps

Once you have a profile, you need to create at least one host that uses the profile. See the following for more information about that.

*Provisioning a Host*

## Provisioning a Host

When you need more computing resources, you must allocate them in a process called *provisioning*. This is the process of creating new hosts.

### Overview

A new host is provisioned in two stages:

1. The virtual host is created using Vagrant
2. The newly created host is configured using Puppet

### Cloud Providers

The virtual host may be created using any provider that Vagrant is capable of interfacing with. Digital Ocean is provided as part of Puppet-Diamond. However, Linux KVM, Xen, and Amazon EC2 have been shown to work with Puppet-Diamond too.

#### Digital Ocean

##### Scaffold a new configuration

```
export HOSTNAME=host1
domo-new.sh digitalocean-host $HOSTNAME
domo-sync.sh
```

##### Create the host

```
cd $PD_PATH/$PD_MASTER/hosts/$HOSTNAME
get_puppet_certs.py $HOSTNAME
generate_sshd_keys.sh
vagrant up --provider=digital_ocean
```

### DNS

Create an A record with this IP in the DNS control on Digital Ocean. Go to [domains list](#) and click to view all PTR records. Click **add record** and then [add A record](#). Finally, paste the IP address and hostname.

### Linux KVM

It is also possible to run your own virtual cloud with any moderately equipped Linux machine. Puppet-Diamond supports this, but the documentation is forthcoming.

### First-run tasks

There are a few tasks that should be run each time a new host is provisioned.

### set password for domo user

```
vagrant ssh  
passwd domo
```

### update packages

```
vagrant ssh  
apt-get update  
apt-get upgrade -y  
reboot
```

### run puppet one last time

```
domo-apply.sh $HOSTNAME
```

## Debugging

It may be necessary to debug the puppet master. In that case, try some of the following:

Try re-provisioning with puppet.

```
vagrant provision
```

See that the cert is really listed on the puppet master.

```
ssh $PD_PUPPETMASTER_SSH_HOST  
puppet cert list --all
```

Restart the puppet master server.

```
ssh $PD_PUPPETMASTER_SSH_HOST  
service puppetmaster restart
```

View the logs

```
ssh $PD_PUPPETMASTER_SSH_HOST  
tail -f /var/log/syslog
```

Log on to the new node and poke around

```
vagrant ssh
```

## Next Steps

Now that you have provisioned a host, you can create new modules to configure it for specific purposes. There are two main types of modules supported by Puppet-Diamond:

- *Creating a Puppet Module*
- *Creating a Flask-Diamond Module*

## Creating a Puppet Module

A Puppet Module is used to control a server resource, like a daemon, an OS package, or hardware.

### Puppet-Diamond environment

Enter the Puppet-Diamond virtual environment and import your configuration settings.

```
workon puppet-diamond
source ~/.puppet-diamond
```

### Scaffold initial files

The first step is to scaffold a new module. The general form of the command is:

```
domo-new.sh diamond-module ${SERVICE_NAME}
```

The following example demonstrates creating a puppet module for Postfix.

```
domo-new.sh diamond-module postfix
```

### Add to a host

Add the new module class to a profile, located in `${PD_PATH}/${PD_MASTER}/profiles`.

```
class {"postfix": ;}
```

### Sync and Apply

Sync to the puppetmaster and apply the changes to the client.

```
domo-sync.sh
domo-test.sh ${PUPPET_HOST}.example.com
domo-apply.sh ${PUPPET_HOST}.example.com
```

## Creating a Flask-Diamond Module

A Flask-Diamond Module will enable a Flask-Diamond application to be installed using Puppet-Diamond onto a host in your enterprise.

### Puppet-Diamond environment

Enter the Puppet-Diamond virtual environment and import your configuration settings.

```
workon puppet-diamond
source ~/.puppet-diamond
```



## Scaffold initial files

The first step is to scaffold a new module. The general form of the command is:

```
domo-new.sh diamond-module ${APPLICATION_NAME}
```

The following example demonstrates creating a diamond module for an application called *my-diamond-app*.

```
domo-new.sh diamond-module my-diamond-app
```

## SSH deploy keys

Create application SSH deploy keys.

```
cd ${PD_PATH}/${PD_MASTER}/diamond/${APPLICATION_NAME}
ssh-keygen -t rsa -f files/ssh/id_rsa
open ${PD_GIT_WEB_URL}/${PD_GIT_GROUP}/${APPLICATION_NAME}/deploy_keys/new
pbcopy < files/ssh/id_rsa.pub
```

## Add to a host

Add the new module class to a profile, located in `${PD_PATH}/${PD_MASTER}/profiles`.

```
class {"my-diamond-app": ;}
```

## Sync and Apply

Sync to the puppetmaster and apply the changes to the client.

```
domo-sync.sh
domo-test.sh ${PUPPET_HOST}.example.com
domo-apply.sh ${PUPPET_HOST}.example.com
```



## Puppet-Diamond License

### The MIT License (MIT)

Puppet-Diamond, Copyright (c) 2015 Ian Dennis Miller.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.