Installing Puppet master

Installation on CentOS/RHEL/Fedora

Install the Puppet yum repository and then the “puppet-server” package. See https://yum.puppetlabs.com to find the correct rpm file needed to install the puppet repo for your Linux distribution. For example, for CentOS 7 or RHEL 7, do the following:

```
$ sudo rpm -ivh https://yum.puppetlabs.com/puppetlabs-release-pc1-el-7.noarch.rpm
$ sudo yum install puppetserver
```

Installation on Debian/Ubuntu

Install `curl`, `apt-transport-https` and `lsb-release`:

```
$ sudo apt-get update
$ sudo apt-get install curl apt-transport-https lsb-release
```

Get the appropriate Puppet apt repository, and then the “puppetserver” package. See https://apt.puppetlabs.com to find the correct deb file to install the puppet repo for your Linux distribution, you can use next script to make installation more silently:

```
$ wget "https://apt.puppetlabs.com/puppetlabs-release-pc1-$(lsb_release -cs).deb"
$ sudo dpkg -i "puppetlabs-release-pc1-$(lsb_release -cs).deb"
$ sudo apt-get update && sudo apt-get install puppetserver
```

Memory Allocation

By default, Puppet Server will be configured to use 2GB of RAM. However, if you want to experiment with Puppet Server on a VM, you can safely allocate as little as 512MB of memory. To change Puppet Server memory allocation, you can edit the following init config file.

- `/etc/sysconfig/puppetserver` – CentOS/RHEL/Fedora
- `/etc/default/puppetserver` – Debian/Ubuntu

Replace 2g with the amount of memory you want to allocate to Puppet Server. For example, to allocate 1GB of memory, use `JAVA_ARGS="-Xms1g -Xmx1g"`; for 512MB, use `JAVA_ARGS="-Xms512m -Xmx512m"`.

Configuration

Edit the `/etc/puppetlabs/puppet/puppet.conf` file, adding this line to the `[main]` section, and replacing `puppet.example.com` with your own FQDN:

```
dns_alt_names = puppet,puppet.example.com
```

Note

If you find `templatedir=$confdir/templates` in the config file, delete that line. It has been deprecated.

Then, restart your Puppet Server to apply changes:

a. For Systemd:

```
$ sudo systemctl start puppetserver
```

b. For SysV Init:

```
$ sudo service puppetserver start
```

PuppetDB installation

After configuring Puppet Server to run on Apache with Passenger, the next step is to add PuppetDB so that you can take advantage of exported resources, as well as have a central storage location for Puppet facts and catalogs.

Installation on CentOS/RHEL 7 (Adjust if your version is different.)

```
$ sudo rpm -Uvh https://yum.postgresql.org/9.4/redhat/rhel-latest-x86_64/pgdg-centos94-9.4-2.noarch.rpm
$ sudo yum install puppetdb-terminus.noarch puppetdb postgresql94-server postgresql94 postgresql94-
contrib.x86_64
$ sudo /usr/pgsql-9.4/bin/postgresql94-setup initdb
$ sudo systemctl start postgresql-9.4
$ sudo systemctl enable postgresql-9.4
```

Installation on Debian/Ubuntu

```
$ sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt/ $(lsb_release -cs)-pgdg main" >
/etc/apt/sources.list.d/pgdg.list'
$ wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add -
$ sudo apt-get update
$ sudo apt-get install puppetdb-terminus puppetdb postgresql-9.4 postgresql-contrib-9.4
```

Configuration

For CentOS/RHEL/Fedora only, the next step is to edit `/var/lib/pgsql/9.4/data/pg_hba.conf` and modify the METHOD to be `md5` in these two lines:

```
# IPv4 local connections:
host    all             all             127.0.0.1/32            md5
# IPv6 local connections:
host    all             all             ::1/128                 md5
```

Restart service after change configuration:

```
$ sudo systemctl restart postgresql-9.4
```

Create a PostgreSQL user and database:

```
# su - postgres
$ createuser -DRSP puppetdb
$ createdb -O puppetdb puppetdb
$ exit
```

```
```

The user is created with no permission to create databases (-D), or roles (-R) and does not have superuser privileges (-S). It will prompt for a password (-P). Let's assume a password of "yourpassword"" has been used. The database is created and owned (-O) by the puppetdb user.

Create the extension pg_trgm is the RegExp-optimized index extension:

```
$ su - postgres
$ psql puppetdb -c 'create extension pg_trgm'
$ exit
```

Test database access:

```
$ psql -h 127.0.0.1 -p 5432 -U puppetdb -W puppetdb
Password for user puppetdb:
psql (9.4.11)
Type "help" for help.
puppetdb=> \q
```

Configure `/etc/puppetlabs/puppetdb/conf.d/database.ini`:

```
[database]
classname = org.postgresql.Driver
subprotocol = postgresql
subname = //127.0.0.1:5432/puppetdb
username = puppetdb
password = yourpassword
log-slow-statements = 10
```

Create `/etc/puppetlabs/puppet/puppetdb.conf`:

```
[main]
server_urls = https://puppetdb.example.com:8081
```

Create `/etc/puppetlabs/puppet/routes.yaml`:

```
---
master:
  facts:
    terminus: puppetdb
    cache: yaml
```

Finally, update `/etc/puppetlabs/puppet/puppet.conf`:

```
[master]
 storeconfigs = true
 storeconfigs_backend = puppetdb
```

Start puppetdb service:

```
$ sudo systemctl start puppetdb
```

Once these steps are completed, restart your Puppet Server and run `puppet agent --test`:

```
$ puppet agent --test
```

Now PuppetDB is working.